

SEAS 8414 · SECURITY-ML PRAXIS · GEORGE WASHINGTON UNIVERSITY

Combined Doctoral Defense Atlas

The examiner's adversarial report and the evidence-grounded defense atlas in one student publication—concession, ruling, axes, live faults, repairs, and proof locators.

STUDENT EDITION · EVIDENCE FREEZE 2026-08-01 · COMBINES EXAMINER REPORT + DEFENSE REVIEW ATLAS

36

notebooks

787

live faults

214

critical (FATAL)

7.5

mean doctoral /30

Three scores, one scientific question

Would the reported result survive as evidence that a detector works outside the file it was measured on?

Praxis rubric (/10)

From the findings pack. Grades audit honesty—provenance, baselines, ablation, probe stability. Many notebooks earn 8.5–10 and deserve it.

Doctoral score (/30)

From the examiner report. Six axes (D1–D6), 0–5 each. Measures whether methods for external validity, inference, construct validity, decisions, reproducibility, and scope are present—not how many words of caveat are printed.

Defense grade (F/D/C)

From the defense atlas. Targets the selected-model headline estimand. Shared bake-off reuse of the holdout for selection and final reporting yields F without erasing narrow diagnostics.

Disclosure ≠ mitigation

Naming the missing experiment in the conclusion does not license the headline. Where the decisive test was affordable and identified by the author, not running it is the finding.

How a fault is labelled. *CRITICAL / FATAL* — refuter could not break it. *MAJOR / CONTESTED* — defect is real but a prepared author has a printed defensible reply. Evidence class is independent of severity: *measured*, *documented*, or *untested*. Withdrawn faults are omitted from the live record after adversarial review.

Evaluation hierarchy. Minimum: time-ordered or entity/group-blocked split. Stronger: campaign or family holdout. Strongest: train on one complete environment and test on another with aligned labels and schema. Random row splits only ask whether a labeled table is separable.

ROC versus PR. ROC-AUC is a secondary ranking check across all false-positive rates. Prefer PR / average precision and precision or TPR at a fixed false-positive budget when positives are rare. When positives are the majority, do not claim “use PR because ROC is base-rate blind”—that argument inverts. Lead with operating-point metrics and rare-class recall.

What this combined edition adds. Examiner prose (concession, ruling, axes, adversarial labels, defensible replies) plus atlas structure (evidence → consequence → repair, optional proof locators when titles match the defense ledger). Not every atlas proof links to every examiner fault; unmatched atlas challenges remain in the separate defense atlas.

PART I · SERIES-WIDE METHOD GAPS

What none of the thirty-six do

Counts are methods present in code, not prose mentions. Naming a limitation scores as disclosure, not as execution.

Method	Using it	Why it matters
Separation of model selection from final evaluation	0 / 36	Cell 16 ranks four learners on the held-out set; cell 18 reports the winner on that same set. The headline is post-selection, not an independent test.
Entity-grouped split (GroupKFold or equivalent)	0 / 36	Flows, hosts, cards, or sessions appear on both sides of random row splits.
Temporal / time-ordered holdout	0 / 36	No notebook tests whether a detector trained on the past works on the future.
Class weighting (scale_pos_weight, class_weight)	0 / 36	Rare-class corpora are fitted with a loss dominated by the majority.
Probability calibration	0 / 36	Scores are thresholded at 0.5 without calibrated probabilities.
PR-AUC / average_precision_score	0 / 36	Every notebook plots a PR curve; none computes or reports its area.
Threshold selected from a curve	0 / 36	The operating point — which is the result for a detector — is never chosen.
Confidence interval or bootstrap	0 / 36	No interval on headline estimates. The only dispersion figure is 3-fold CV on a train slice.
Hyperparameter search	0 / 36	Models run at hand-set constants never searched.
Explicit cost matrix	0 / 36	False positives and false negatives are treated as equally expensive.
Corpus integrity check (checksum / hash)	0 / 36	No SHA-256 of downloaded corpora; many notebooks resolve files by glob on third-party mirrors.
warnings.filterwarnings('ignore') before modelling	36 / 36	Every notebook silences warnings, then presents clean output. Absence of a warning is not evidence.
precision_recall_curve plotted	36 / 36	The right instrument is drawn in all 36 and never read as a scalar.

Read the last two rows together. Every notebook plots a precision–recall curve. Not one computes the area, prints it, or uses it to rank models. On attack-majority corpora the PR panel is additionally uninformative about the scarce class if the majority is treated as positive.

PART II · SCOREBOARD

Defense, doctoral, and praxis scores

Where praxis and doctoral disagree sharply, the notebook audits itself well and still cannot support its headline.

36

DEFENSE F (SELECTED-MODEL ESTIMAND)

7.5

MEAN DOCTORAL /30

214

LIVE CRITICAL FAULTS

36

NOTEBOOKS SCORED

NB	Notebook	Defense	Doctoral	Rubric	Self	Band	Crit	Live
NB01	Network Intrusion Detection on the KDD Cup 1999 Corpus	F	7/30	9/10	F	REJECT WITH PREJUDICE	9	19
NB02	CIC-IDS2018 Case Study - FTP/SSH Brute Force	F	6/30	9.5/10	D	REJECT WITH PREJUDICE	5	22
NB03	CIC-IDS2018 Case Study - DoS (GoldenEye / Slowloris)	F	6/30	9/10	C	REJECT WITH PREJUDICE	5	21
NB04	CIC-IDS2018 Case Study - Web Attacks (Brute Force / XSS / SQLi)	F	7/30	10/10	B	REJECT WITH PREJUDICE	10	23
NB05	Intrusion Detection on a 5G Network: the 5G-NIDD Dataset	F	7/30	10/10	F	REJECT WITH PREJUDICE	2	22
NB06	IoT/IIoT Intrusion Detection: the Edge-IIoTset Dataset	F	8/30	10/10	A	REJECT	3	22
NB07	Network Intrusion Detection on CIC-IDS2017	F	6/30	9.5/10	A	REJECT WITH PREJUDICE	5	21

NB	Notebook	Defense	Doctoral	Rubric	Self	Band	Crit	Live
NB08	IoT/IIoT Network-Flow Intrusion Detection: CIC-ToN-IoT	F	7/30	9.5/10	B	REJECT WITH PREJUDICE	3	21
NB09	Network Intrusion Detection on UNSW-NB15	F	8/30	9.5/10	D	REJECT	6	23
NB10	Botnet Detection on the CTU-13 Dataset	F	10/30	9.5/10	D	REJECT	9	23
NB11	NetFlow-Standardized Intrusion Detection: NF-BoT-IoT-V2	F	5/30	10/10	B	REJECT WITH PREJUDICE	6	21
NB12	NetFlow-Standardized Intrusion Detection: NF-ToN-IoT-V2	F	7/30	10/10	A	REJECT WITH PREJUDICE	7	21
NB13	NetFlow-Standardized Intrusion Detection: NF-UNSW-NB15-V2	F	8/30	9.5/10	D	REJECT	6	22
NB14	IoT Attack Detection on CICIoT2023	F	7/30	9/10	D	REJECT WITH PREJUDICE	9	21
NB15	Mobile-Money Fraud Detection: the PaySim Simulator	F	6/30	10/10	A	REJECT WITH PREJUDICE	9	22
NB16	Credit-Card Fraud Detection: the Sparkov Simulated Dataset	F	5/30	5.5/10	A	REJECT WITH PREJUDICE	10	21
NB17	Large-Scale Financial Transaction Fraud Detection	F	7/30	7/10	A	REJECT WITH PREJUDICE	3	22
NB18	Malicious URL Detection (Phishing / Defacement / Malware)	F	8/30	10/10	B	REJECT	2	23
NB19	IoT Malware Flow Detection on IoT-23 (Zeek conn.log)	F	8/30	9/10	F	REJECT	3	22
NB20	NetFlow-Standardized Intrusion Detection: NF-CSE-CIC-IDS2018-V2	F	8/30	10/10	B	REJECT	3	23
NB21	Host-Based Intrusion Detection on BETH Kernel Telemetry	F	6/30	8.5/10	F	REJECT WITH PREJUDICE	8	22
NB22	Credential-Attack Detection on the RBA Login Dataset	F	7/30	9.5/10	F	REJECT WITH PREJUDICE	8	23
NB23	Cross-Dataset Intrusion Detection on NF-UQ-NIDS-v2 (Merged NetFlow Superset)	F	8/30	10/10	B	REJECT	9	22
NB24	Static PE Malware Detection on EMBER-2018	F	8/30	9.5/10	A	REJECT	7	23

NB	Notebook	Defense	Doctoral	Rubric	Self	Band	Crit	Live
NB25	Encrypted DNS-over-HTTPS Tunnel Detection (CIRA-CIC-DoHBrw-2020)	F	8/30	10/10	A	REJECT	4	21
NB26	DDoS Detection with Per-Type Recall on CIC-DDoS2019	F	7/30	10/10	B	REJECT WITH PREJUDICE	6	23
NB27	IoT Botnet Detection on N-BaIoT (Mirai / BASHLITE Device Behaviour)	F	8/30	10/10	C	REJECT	6	22
NB28	Real E-Commerce Card-Fraud Detection on IEEE-CIS (Vesta)	F	8/30	10/10	A	REJECT	6	23
NB29	System-Log Anomaly Detection on HDFS (Loghub)	F	7/30	9.5/10	F	REJECT WITH PREJUDICE	5	20
NB30	Industrial-IoT Intrusion Detection on WUSTL-IIoT-2021	F	8/30	9.5/10	C	REJECT	7	21
NB31	Cyber-Physical Intrusion Detection on the HAI ICS Testbed	F	9/30	10/10	A	REJECT	10	22
NB32	In-Vehicle CAN-Bus Intrusion Detection (Car-Hacking / HCRL)	F	7/30	10/10	F	REJECT WITH PREJUDICE	7	20
NB33	Cyber-Physical Intrusion Detection on the SWaT Water-Treatment Testbed	F	8/30	10/10	B	REJECT	4	23
NB34	In-Vehicle CAN Intrusion Detection on OTIDS (Second Vehicle)	F	7/30	8/10	F	REJECT WITH PREJUDICE	6	23
NB35	Android Malware Detection & Category Characterization (CCCS-CIC-AndMal-2020)	F	10/30	10/10	F	REJECT	4	23
NB36	Does It Transfer? Cross-Dataset Intrusion Detection on One Schema	F	14/30	9.5/10	B	REJECT	2	21

Network Intrusion Detection on the KDD Cup 1999 Corpus

simulated 1998 DARPA military-testbed traffic, four million eight hundred ninety-eight thousand four hundred thirty-one connection records

HEADLINE

XGBoost, held-out ROC-AUC zero point nine nine nine nine nine eight, accuracy zero point nine nine nine eight seven

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 19 live faults
- 9 critical (FATAL)
- 10 major (CONTESTED)
- 4 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me start with what this notebook does better than most published work in this area. It prints the majority-class baseline inside the ranking table, so nobody can mistake eighty percent accuracy for a result. It independently reproduces Tavallae's duplicate finding, measuring seventy-eight point one percent exact-duplicate rows on a corpus where the literature says seventy-eight percent. It corrects the common error that KDD ninety-nine derives from the DARPA nineteen ninety-nine evaluation, and states correctly that it derives from nineteen ninety-eight. It grades its own data an F and says so in the abstract. And when the ablation refuses the easy story, it does not spin the result as vindication; it reads the surviving score as evidence about the simulator. That is genuine intellectual honesty and I want it on the record. Now here is the problem with all of it.

EXAMINER RULING

This notebook identified the experiment that would settle its own central claim, wrote down in its own conclusion that it was the most important missing experiment, named the corrected benchmark that exists precisely to run it, and then published a ROC-AUC of zero point nine nine nine nine nine eight anyway.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Seven attack families vanish without a printed count

EVIDENCE

Cell eighteen contains the line, quote, if count is less than five, continue, unquote, with the comment that you need a few positives for a meaningful recall. Re-running that loop, the audit enumerated twenty attack families present in the held-out split, of which seven fall below the floor: multihop, imap, rootkit, land, perl, ftp-write and loadmodule. Three of those are user-to-root families; only buffer overflow survived. Three more are remote-to-local families. So thirteen families are scored, twelve are plotted, because the plot shows the worst nine plus the best three, and exactly six are printed in the text.

SCIENTIFIC CONSEQUENCE

The section-thirteen conclusion nominates rare-class recall as, in its own words, the honest lead numbers here. The floor deletes six of the seven rarest remote-to-local and user-to-root families before that number is computed. The notebook's designated honest metric is calculated on a set from which the honest cases have been removed, and the removal leaves no trace in any output. A code comment discloses the mechanism. Nothing discloses the outcome, and nothing discloses the count.

REQUIRED REPAIR

Print every family with its positive count and its recall, and mark the sub-five families as unscorable rather than omitting them. If a class cannot be scored, that is the finding, and it must appear in the output.

The dedup ablation tests the wrong kind of duplicate

EVIDENCE

Cell twenty builds the training key set with numpy round to six decimal places and tuple equality. Cell twenty-two calls pandas drop-duplicates, which is exact equality. On that basis it removes seventy-eight percent of rows and reports that the score holds at zero point nine nine nine nine three.

SCIENTIFIC CONSEQUENCE

This corpus is dominated by denial-of-service floods. A flood generates thousands of connection records that differ in one host counter and are otherwise identical. Exact matching classifies every one of those as a distinct row, so the ablation rules out only the leakage that was easiest to rule out. The notebook's single strongest defensive move, the claim that de-duplication barely moved the score, therefore establishes nothing about the leakage that actually matters, and the notebook presents it as if it did.

REQUIRED REPAIR

Compute the nearest-neighbour distance distribution from each held-out row to its closest training row, and report the fraction of test rows within a small epsilon. Then re-run the ablation removing near-duplicates, not just exact ones.

Axelsson is cited and never applied to its own number

EVIDENCE

Section two cites Axelsson, two thousand, and says, in the notebook's own words, that at a realistic low attack base rate even a very accurate detector drowns in false positives, and that this corpus's roughly eighty percent attack rate is itself unrealistic. Cell eighteen then prints an operational false-positive rate at threshold zero point five of zero point zero zero zero two, from forty-five benign flows flagged out of two hundred forty-three thousand one hundred ninety-five. Forty-five divided by two hundred forty-three thousand one hundred ninety-five is zero point zero zero zero one eight five.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic the notebook owed and did not do, and you can check it live. Take a realistic base rate of one attack per ten thousand flows. In one million flows that is one hundred attacks and nine hundred ninety-nine thousand nine hundred benign. At the notebook's own recall, which is essentially one, you catch about one hundred attacks. At the notebook's own false-positive rate you raise nine hundred ninety-nine thousand nine hundred times zero point zero zero zero one eight five, which is one hundred eighty-five false alarms. Precision is one hundred over two hundred eighty-five, which is zero point three five. Roughly two out of every three alerts are wrong, using nothing but the notebook's own printed numbers. That is Axelsson's result, and the notebook cited Axelsson and did not compute it.

REQUIRED REPAIR

Multiply the measured false-positive rate by a stated daily flow volume, report alerts per analyst per day, and report precision at a realistic base rate rather than the corpus base rate.

The corrected benchmark is cited, then declined

EVIDENCE

Section two cites Tavallae and colleagues, two thousand nine, for measuring roughly seventy-eight percent duplicate records that bias learners, and for releasing NSL-KDD. The caveat table in that same section carries a row that reads, in the notebook's own words, that the de-duplicated protocol is the fairer benchmark. The section-thirteen conclusion then states that this is a single in-distribution random split, and not the official corrected or NSL-KDD protocol. The notebook measures the duplicate rate itself, at zero point seven eight one in cell twenty, confirming Tavallae's number to the third decimal.

SCIENTIFIC CONSEQUENCE

You cannot cite a paper for the proposition that your corpus is defective, cite the same paper for the corrected release, reproduce the defect measurement to three decimals, and then decline to use the correction. Every number downstream is a measurement of the defect the notebook already agreed exists. There is no inferential step that survives that.

REQUIRED REPAIR

Run the experiment on NSL-KDD, using the published KDDTrain-plus and KDDTest-plus partition, and report both numbers side by side. That is a loader change, not a research programme.

DEFENSIBLE REPLY

Section one states my research question in terms: not 'can we score high' but 'does the score mean detection, or a shortcut in the data' — that is a question about this file, and you cannot answer it by running a different file; NSL-KDD belongs beside the audit as a comparator, not in place of the object under audit.

ORDERED CORRECTION

Delete 'That is a loader change, not a research programme.' NSL-KDD is 148,517 records total (KDDTrain+ 125,973 plus KDDTest+ 22,544) and cannot serve as the primary corpus under the notebook's own `assert len(df) >= 1\_000\_000` gate, so the change is a supplementary comparator cell, not a swap. Reword the charge as 'the corrected partition is never added as a comparator', and merge with fault 18, whose REQUIRED is the same action.

Saturation asserted in a comment, never measured

EVIDENCE

The comment in cell sixteen reads, quote, we fit on a stratified subsample of at most one hundred twenty thousand because these learners saturate long before then on this data, unquote. There is no learning curve anywhere in the notebook. No score is reported at any other training size. The abstract meanwhile leads with the word materialize and the figure four million eight hundred ninety-eight thousand four hundred thirty-one.

SCIENTIFIC CONSEQUENCE

Saturation is an empirical property. It is measurable in about six lines. The notebook asserts it in a code comment in order to justify keeping one hundred twenty thousand rows out of the three point seven million available for training, and then reports the full corpus size as the headline scale of the study. The number that describes the study is one hundred twenty thousand. The number in the abstract is four point nine million. Those are different experiments and only one of them was run.

REQUIRED REPAIR

Plot held-out ROC-AUC against training size across at least five decades, from one thousand to the full corpus, and show the curve flattening. Or drop the saturation claim and state the effective n in the abstract.

DEFENSIBLE REPLY

The abstract's opening sentence reads 'We materialize the full 4,898,431-record corpus and train four learners on a stratified subsample of 120,000 rows' — the effective n you ask me to state is in the same sentence as the number you say hides it, it is printed again by cell sixteen and stated again in section eight, and the held-out evaluation is 1,224,608 rows.

ORDERED CORRECTION

Strike the sentence 'The number that describes the study is 120,000. The number in the abstract is 4.9 million' and strike the alternative remedy 'state the effective n in the abstract' — both are already satisfied three times over. The surviving charge is narrow and correct: a 97.5% discard of the training half is justified by an empirical claim asserted in a comment and never measured, so the six-line learning curve is still owed.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Connections from one host land on both sides

EVIDENCE

Cell sixteen calls train-test-split with test size zero point two five, stratified on the label, random state zero. There is no grouping key of any kind. Across all thirty-six notebooks in this series, group k-fold appears zero times. The section-thirteen scope paragraph states, in the notebook's own words, that the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

Under that split the model does not need to learn what an attack is. It needs to recognise a neighbour it has already seen. Zero point nine nine nine nine eight is then a measurement of how well nearest-neighbour matching works inside one file, which is not the question the notebook asks in section one.

REQUIRED REPAIR

Group shuffle split or group k-fold keyed on source host and attack session, so that no session straddles the boundary. Report the drop.

DEFENSIBLE REPLY

Read scikit-learn's dtype list for this loader: duration, protocol_type, service, flag, byte counts, and thirty-odd derived rate and host counters — there is no host, no port, no timestamp and no connection identifier anywhere in it, so the group key you are asking me to split on does not exist in the data.

ORDERED CORRECTION

Replace REQUIRED. KDD99 ships no entity or time key, so an entity-grouped or temporal protocol cannot be constructed from it. The correct demand is that the notebook state this: that the corpus admits no grouped, session-aware or temporal split at all, and that this unfixability — not the author's choice of split — is the finding that belongs in the abstract.

7

MAJOR

DOCUMENTED

VERIFIED

TRANSDUCTIVE PREPROCESSING

The reassuring clause describes a corpus that is not this one

EVIDENCE

Cell eleven fits a fresh LabelEncoder on the full column for each of protocol type, service and flag, before any split occurs. The section-thirteen conclusion then says, quote, on an all-numeric corpus that step is a no-op, unquote. This corpus is not all-numeric. It has exactly those three categorical features, and the notebook says so in the same cell.

SCIENTIFIC CONSEQUENCE

The encoder sees every category value present in the test set, including any it would not have seen from training alone. No label leaks, so this does not by itself invalidate the score. What it does invalidate is the protocol claim: the notebook cannot describe its pipeline as deployment-shaped when the fitted preprocessing has consumed the held-out data, and it cannot reassure the reader with a sentence whose antecedent is false for this notebook.

REQUIRED REPAIR

Fit the encoder inside the training fold, and map unseen categories to an explicit unknown bucket. Then delete the all-numeric clause.

DEFENSIBLE REPLY

The sentence is conditional and the condition plainly is not ours; the transductive step is disclosed in full in the immediately preceding sentence, and no label information leaks, so the headline is unaffected.

8

CRITICAL

DOCUMENTED

VERIFIED

SINGLE SEED FOR THE HEADLINE

One seed, and the notebook says so in a comment

EVIDENCE

Cell ten sets random state to zero, with the comment, quote, single seed used everywhere, unquote. The split, the subsample, the ablation re-splits and the cross-validation all take that same seed. The eight-seed stability sweep that appears in the audit was run by the auditor on the single-feature check, not by the notebook, and never on the reported model.

SCIENTIFIC CONSEQUENCE

A number without a resampling distribution is not an estimate, it is an observation. The notebook reports six significant figures on a quantity for which it has one draw. The comment in cell ten is the notebook telling you this and proceeding anyway.

REQUIRED REPAIR

Ten or more seeds on the full pipeline, reporting mean and standard deviation of the held-out ROC-AUC, and quote the headline to the resolution that spread supports.

10

CRITICAL

UNTESTED

VERIFIED

NO UNCERTAINTY ANYWHERE

A recall of zero with no denominator on the page

EVIDENCE

Cell eighteen prints, quote, warezmaster zero point zero, buffer overflow zero point one six seven, unquote. Neither the count of warezmaster flows nor the count of buffer-overflow flows appears anywhere in the notebook. The count-five floor guarantees only that each is at least five. A recall of zero point one six seven is consistent with one out of six. Bootstrap appears zero times across all thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

Zero point one six seven on six events and zero point one six seven on six thousand events are different scientific claims, and the notebook does not let you tell which one it made. The conclusion then builds its most important sentence, that rare-class recall is the honest lead number, on quantities whose sample size is not on the page.

REQUIRED REPAIR

Print the positive count next to every per-family recall, and attach a Clopper-Pearson binomial interval to each. Bootstrap the held-out ROC-AUC and report the interval.

12

CRITICAL

UNTESTED

VERIFIED

THRESHOLD FIXED AT 0.5, NO COST MODEL

The corpus was a cost-matrix competition; this is scored on accuracy

EVIDENCE

The section-two caveat table names Pfahringer, two thousand, the KDD Cup ninety-nine winning entry, with the caveat, in the notebook's own words, cost-matrix scoring, not accuracy; user-to-root and remote-to-local dominate the cost. Cell eighteen then thresholds at zero point five with a hard-coded comparison. No cost matrix appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The competition that produced this corpus was decided on a cost matrix precisely because the rare classes carry the risk. The notebook records that fact in its own literature review, and then scores on accuracy and ROC-AUC, which weight a missed buffer overflow the same as a missed smurf flood. Under the canonical scoring for this dataset, a model with warezmaster recall of zero does not win.

REQUIRED REPAIR

Apply the published KDD Cup ninety-nine cost matrix, select the threshold that minimises expected cost, and report the cost per connection alongside the accuracy.

13

CRITICAL

MEASURED

VERIFIED

MODEL RANKING WITHOUT A TEST

A winner declared on three parts in a million

EVIDENCE

The cell-sixteen table shows XGBoost at zero point nine nine nine nine eight and RandomForest at zero point nine nine nine nine five. That is a gap of three parts in a million. The line, quote, best model, XGBoost, unquote, is printed on that basis, and XGBoost is the model carried into every downstream diagnostic. Across all thirty-six notebooks, McNemar or any paired significance test appears zero times.

SCIENTIFIC CONSEQUENCE

A gap of three parts in a million on a single split with a single seed is not a difference; it is a coin flip. Every downstream section, the confusion matrix, the per-family recalls, the ablation, the cross-validation, is conditioned on a model chosen by that coin flip. The conclusion's entire ledger inherits an arbitrary selection.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, or overlapping bootstrap intervals. If they do not separate, say they do not separate and report both.

15

CRITICAL

UNTESTED

VERIFIED

DEDUP DISCARDS LABEL DISAGREEMENT

Conflicting labels resolved by row order

EVIDENCE

Cell twenty-two calls pandas drop-duplicates on the feature matrix and then indexes the label array by the surviving index. The conclusion states, in the notebook's own words, that the de-duplication variant keeps the first label when a feature vector appears twice. Seventy-eight point one percent of rows are exact duplicates, so this rule is applied to the great majority of the corpus.

SCIENTIFIC CONSEQUENCE

When one feature vector carries two labels, that is irreducible Bayes error, and it is a hard ceiling on any achievable accuracy. Resolving it by row order deletes the ceiling from the data. The de-duplicated ablation therefore reports a score on a task that has been made easier, and the notebook offers that score as evidence that the headline is not duplicate-driven. It is evidence of the opposite kind.

REQUIRED REPAIR

Report the label-disagreement rate among duplicated feature vectors. That rate is the noise floor, and the headline must be read against it.

Atlas proof

docs/course/security-ml-notebooks/nb01-kdd99-intrusion-detection.ipynb · cell 22; cell 25 computation disclosure; zero-based manifest cell IDs: 22=5287c566, 25=f3878f36

A security detector with no adversary

EVIDENCE

No perturbation, no evasion, no robustness experiment appears in any cell. The features the model leans on, connection count, service error rates, protocol and service identifiers, are attacker-controlled at essentially zero cost. The strongest single feature is count, at zero point nine nine two nine on its own.

SCIENTIFIC CONSEQUENCE

In every other engineering discipline you report the worst case for a safety system. Here the notebook reports the best case, on a corpus that contains no adaptive attacker, and does not report the worst case at all. An adversary who paces a scan to hold count inside the benign range does not need to defeat the model; the model has told him which knob to turn.

REQUIRED REPAIR

Define a perturbation budget over attacker-controllable columns, generate evaded variants, and report worst-case recall under that budget.

DEFENSIBLE REPLY

I concede pacing `count` is realizable and worth measuring. But you asked for a perturbation budget over attacker-controllable columns, and thirty of my forty are derived rates and hundred-connection host aggregates with no packet substrate behind them — perturbing those in feature space produces vectors no attacker can actually emit, so the worst case you would get back is a worst case against the simulator, not against an adversary.

ORDERED CORRECTION

Narrow REQUIRED to the single realizable perturbation already named: hold `count` and `srv\_count` inside the benign range and report the recall drop. Drop the general 'perturbation budget over attacker-controllable columns' — it is ill-posed over KDD99's derived aggregates, since thirty of the forty columns are two-second and hundred-connection rates with no packet substrate, so a feature-space edit there has no problem-space preimage.

The right conclusion, drawn from an unmeasured quantity

EVIDENCE

The section-thirteen conclusion states, in its own words, that the root cause is that the simulated attack traffic is trivially separable from normal across many redundant features, and that the attack and benign distributions barely overlap. The evidence offered is the ablation table: zero point nine nine nine nine nine eight, falling to zero point nine nine nine nine three after de-duplication and zero point nine nine nine nine nine six after dropping count. No divergence, no Bayes-error estimate, no overlap statistic is computed.

SCIENTIFIC CONSEQUENCE

I credit the notebook with reaching the correct conclusion. But the conclusion is a claim about a distributional property, and the evidence offered is two ablation deltas at the sixth decimal place. Those deltas are consistent with the overlap story and also consistent with a dozen others, including exactly the near-duplicate leakage that the exact-match dedup could not see. The right answer reached by an argument that does not support it is still not a result.

REQUIRED REPAIR

Estimate class overlap directly: a Bhattacharyya or Jensen-Shannon divergence per feature, or a k-nearest-neighbour Bayes-error bound. Then the sentence in the conclusion becomes a measurement.

DEFENSIBLE REPLY

Section ten computes a direction-agnostic AUC for every one of my forty features and plots the strongest fifteen with the near-perfect band coloured red. AUC is the probability that a random attack outranks a random benign — 0.5 is total overlap — so you are looking at forty per-feature overlap measurements and telling the room I computed none.

ORDERED CORRECTION

Delete 'no divergence, no Bayes-error estimate, no overlap statistic is computed' — forty per-feature rank-separation statistics are computed and fifteen are plotted. Narrow the charge to the specific clause you can still win: forty univariate statistics do not bound the joint Bayes error, so 'the attack and benign distributions barely overlap' remains a joint claim supported only by marginal evidence. Keep the k-NN Bayes-error bound in REQUIRED; drop the per-feature divergence, which is redundant with what is already there.

No published number to stand beside

EVIDENCE

The cell-sixteen ranking table contains four learners and one majority baseline at accuracy zero point eight zero one four. Section two lists Pfahringer's winning entry and Tavallae's analysis, and the caveat table states explicitly that these are not measurements reproduced on this corpus. No published figure appears in any output cell.

SCIENTIFIC CONSEQUENCE

Beating a majority baseline on your own split establishes that your code runs. It does not place your number in the field. Because the split is your own and no canonical partition was used, there is no published result on this planet that your zero point nine nine nine nine nine eight can be compared to, including the one from the competition this corpus was built for.

REQUIRED REPAIR

Run the canonical corrected test partition, and report your number in the same table as at least one published figure computed on the same partition.

The missing experiment is a file in the same folder

EVIDENCE

The section-thirteen conclusion states, in the notebook's own words, that defensible network intrusion detection evidence requires cross-distribution or temporal testing, which this notebook does not provide, and that this testing is the most important missing experiment. The directory containing this notebook also contains nb thirty-six, cross-distribution transfer. Both are part of the same series with the same loader contract.

SCIENTIFIC CONSEQUENCE

This is the master fault of the whole set and it is at its sharpest here. The experiment was not unaffordable, not unavailable, not outside the scope of the course. The machinery is in the same folder, written by the same author, against the same interface. Naming the missing experiment while the apparatus to run it sits beside you is not a limitation. It is a decision.

REQUIRED REPAIR

Train on this corpus, test on any other intrusion corpus in the series, in both directions, and report the transfer drop in the abstract.

DEFENSIBLE REPLY

Open nb36's first code cell. Its own comment says the shared forty-three-field NetFlow v2 layout is what makes train-on-one test-on-another possible, and its corpus dictionary is four Kaggle NetFlow datasets behind an API token. KDD99 is forty-one 1998 connection-record features and shares a schema with none of the other thirty-five files — pointing that machinery at this one is a feature-mapping research problem, which is precisely why Sarhan had to re-feature four corpora before nb36 could exist.

ORDERED CORRECTION

Replace the availability mechanism. The accurate and stronger version: a same-schema distribution shift for KDD99 does exist — the official `corrected` test partition, 311,029 records, which by design contains attack types absent from the training file — and `fetch\_kddcup99` does not ship it, so what nb01 owes is an external download and a loader, not a call to nb36. Separately, and correctly, require nb01 to cite nb36 as the series' transfer experiment instead of leaving the reader with 'not run'.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 1,224,608 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Atlas proof

docs/course/security-ml-notebooks/nb01-kdd99-intrusion-detection.ipynb · cell 10;
zero-based manifest cell IDs: 10=8f244d46

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. This notebook is the most honest failure in the set, and I mean that as a real compliment and as the indictment. It found the shortcut, measured the contamination at sixty-eight point nine percent, graded its own data an F, named the corrected benchmark, named the missing experiment, and printed all of it. And then it put zero point nine nine nine nine eight in the abstract. At doctoral level, a disclosure paragraph is not a defence; it is a signed confession that you knew. The experiment that would have settled this was affordable, the corrected corpus was cited by name, and the transfer notebook was in the same folder. The finding is that KDD ninety-nine remains unusable, which the field established in two thousand nine. Nothing in this notebook advances that.

CIC-IDS2018 Case Study - FTP/SSH Brute Force

one day of CSE-CIC-IDS2018 flow records, Wednesday the fourteenth of February two thousand eighteen, machine-generated benign traffic plus two scripted brute-force tools

HEADLINE

RandomForest, held-out ROC-AUC one point zero zero zero zero zero zero, accuracy zero point nine nine nine nine nine two

DEFENSE

F

Selected-model estimand

DOCTORAL

6

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9.5

/10 · self D

D1 1 External validity	D2 1 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 5 critical (FATAL)
- 17 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Two things here are better than the published literature on this corpus. First, the destination-port caveat is written into the provenance table, unprompted, before any score appears; most papers on CIC-IDS2018 never mention it at all. Second, the notebook draws a distinction almost nobody draws: it says that Engelen's label corrections are specific to the two thousand seventeen capture and do not transfer to this two thousand eighteen day, so it declines to claim their findings as its own. That is careful scholarship. And the sentence, read multi-feature narrowly, is the single most honest line in the set. Every one of those is a reason to take this notebook seriously, which is why what follows is a rejection rather than a shrug.

EXAMINER RULING

This notebook wrote down, in its own section ten, that no number in it shows what the model scores with the destination-port column removed, and then published a ROC-AUC of exactly one.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

1

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

ORAL TOP 5

You declared the headline untested and published it

EVIDENCE

The section-three provenance table says, in the notebook's own words, that on this capture the destination-port column is closer to a name for the attack class than to a description of behaviour. Section ten then says, and I am quoting exactly, the ablation in section eleven drops only the top-ranked feature, so no number in this notebook shows what the model scores with the port column removed; treat that as an untested part of the headline, not a cleared one. The headline in the ranking table is one point zero zero zero zero zero zero.

SCIENTIFIC CONSEQUENCE

This is the master fault of the entire set, stated by the notebook about itself, in the imperative mood, to the reader. The ablation machinery is already written and already runs twice in cell twenty-two. Adding a third row that drops the port column is one line. The notebook identified the experiment, wrote a paragraph explaining why the experiment matters, declined to run the one line, and shipped the number. At doctoral level that paragraph is not a caveat. It is a signed statement that you knew.

REQUIRED REPAIR

Add a third ablation row that drops the destination-port and protocol columns and refits, and report that number as the headline. If the score moves, the original headline was a schedule artifact. If it does not, you have said something.

2

CRITICAL

UNTESTED

VERIFIED

NO TEMPORAL HOLDOUT ON TIMESTAMPED DATA

ORAL TOP 5

The timestamp was in your hands and you deleted it

EVIDENCE

Cell eleven defines the drop list as Label, Timestamp, y and family. So the Timestamp column exists in the file, and the notebook proves it exists by naming it in order to remove it. Cell sixteen then calls a random stratified split at test size zero point two five. Across all thirty-six notebooks in this series, a temporal or time-ordered split appears zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

Availability is proved by the deletion. You cannot plead that the ordering was unavailable when your own loader names the column. A brute-force attack is a burst in time: one tool, one target, a few minutes. A random shuffle scatters that burst across both sides of the split, so the model trains on the first half of the burst and is tested on the second half. That is not detection. Sort by timestamp, train on the earlier flows and test on the later ones, and the experiment becomes about generalisation. That is a two-line change and it was not made.

REQUIRED REPAIR

Sort by Timestamp, train on the earlier portion, test on the later portion, and report the drop between the random-split score and the time-ordered score. Report both in the abstract.

The row count is a spreadsheet constant

EVIDENCE

The loader prints one million forty-eight thousand five hundred seventy-five rows. Two to the twentieth power is one million forty-eight thousand five hundred seventy-six. The download in cell eleven pipes through head with a cap of one million two hundred thousand lines, so the cap is one hundred fifty-one thousand four hundred twenty-five lines above the file and provably never engaged. And here is the part you can check in the room: the two sibling notebooks in this series, on the DoS day and the web-attack day, are different captures on different dates, and both print the identical count, one million forty-eight thousand five hundred seventy-five. Three separate days of network traffic do not produce identical row counts. On the DoS day the auditor confirmed the cached file is byte-identical to the S3 object, at three hundred seventy-five million nine hundred forty-five thousand eight hundred ninety-nine bytes on both sides, so the truncation is upstream of the download, not in it.

SCIENTIFIC CONSEQUENCE

If the published CSV is a truncated export, then the attack rate of thirty-six point three three percent and the majority baseline of zero point six three six seven are properties of however much of the day fitted inside a spreadsheet, not properties of the day. Every downstream figure is conditioned on that rate. And note what the honesty gate in cell eleven does: it asserts that the row count is at least one million, and it passes only because a spreadsheet row limit happens to sit above one million. The course requirement of at least one million records is being satisfied by a Microsoft constant.

REQUIRED REPAIR

Run word-count on the cached file, compare its byte size against the S3 object with a head request, and parse the Timestamp column to check whether the file ends mid-day with no taper. Then report the true corpus size, or state that the file is a truncated export.

Two errors on a test set one third memorised

EVIDENCE

Cell twenty prints an exact-duplicate row rate over the whole corpus of zero point three five six and a train-test exact-row contamination of zero point three two nine. Cell sixteen prints held-out size two hundred sixty-two thousand one hundred forty-four and RandomForest accuracy zero point nine nine nine nine nine two, which pins the total error count at exactly two. Cell eighteen prints one benign flow flagged out of one hundred sixty-six thousand nine hundred seven, so of those two errors, one is a false positive and one is a false negative, against ninety-five thousand two hundred thirty-seven attack flows in the split.

SCIENTIFIC CONSEQUENCE

Do the arithmetic out loud. Thirty-two point nine percent of the rows you are scoring on, the model has already seen with its label attached. It gets two of two hundred sixty-two thousand one hundred forty-four wrong. That is not a detector; it is a lookup table with a small miss rate, and the notebook's own contamination figure tells you which. Worse, the contamination is measured by exact match on rounded vectors, so brute-force flows that differ in a single byte counter are counted as fresh, unseen rows. The true leakage is at least thirty-two point nine percent and the notebook has no measurement of how much more.

REQUIRED REPAIR

Measure the nearest-neighbour distance from each held-out row to its closest training row, report the fraction within a small epsilon, and re-run the entire study on a corpus de-duplicated at that tolerance.

Two models tied at one; the sort picked the winner

EVIDENCE

The cell-sixteen table shows RandomForest at one point zero zero zero zero zero zero and XGBoost at one point zero zero zero zero zero zero. Both values are rounded to six places before they are stored, so as floating-point numbers they are exactly equal. The code sorts that column descending and takes the first row, and prints, quote, best model, RandomForest, unquote. There is no tie-break rule and no significance test anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The winner of this study is not determined by any measured difference. It is determined by where the sort placed two identical values. That model is then carried into the confusion matrix, the per-family recalls, the feature importances, the ablation and the cross-validation, so every downstream number in section nine, ten, eleven and twelve is conditioned on an arbitrary choice. And notice what the accuracies say: RandomForest makes two errors on the held-out split and XGBoost makes four, so on the axis that is not tied, the two models are separated by two events.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the two tied models. If they do not separate, report both and say the ranking is undetermined.

DEFENSIBLE REPLY

My conclusion says four learners reach near-perfect ROC-AUC, not that RandomForest is better than XGBoost; the four are separated by 0.000017 across the whole table, and on the axis that is not tied RandomForest is also the more accurate of the two, at two held-out errors against four.

ORDERED CORRECTION

Strike the title 'the sort picked the winner': RandomForest also leads on accuracy (2 errors against XGBoost's 4, both uniquely pinned by the six-decimal figures), so the sort's tie-break coincides with the untied axis. The durable charge is narrower and correct: the ranking is undetermined because two errors against four cannot be separated by any test, and no test was run. Keep the McNemar REQUIRED.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

One attack session, split down the middle

EVIDENCE

Cell sixteen calls a stratified random split with random state zero. There is no grouping key. Across all thirty-six notebooks in this series, group k-fold and group shuffle split appear zero times. The section-thirteen scope paragraph states, in the notebook's own words, that the split is random, not temporal or entity-grouped, and that every number therefore measures in-distribution separability only.

SCIENTIFIC CONSEQUENCE

This day contains two tool runs against one target. Under a random split, the model is trained on some flows from a run and tested on other flows from the same run, generated by the same process, seconds apart. Under that design a score of one is the expected result and carries no information about a run the model has never seen.

REQUIRED REPAIR

Group shuffle split keyed on source-destination pair and attack session, so that no session appears on both sides, and report the drop.

DEFENSIBLE REPLY

My provenance table says this file carries no flow-ID and no IP column, and you can check the header in one line: eighty columns beginning Dst Port, Protocol, Timestamp - the source-destination pair you want me to group on is not in the data at any price.

ORDERED CORRECTION

Replace REQUIRED entirely. CSE-CIC-IDS2018's per-day ML CSVs carry no address columns, so GroupShuffleSplit on a source-destination pair is unbuildable. Two keys that ARE in df: (a) family - leave-one-tool-out, train on benign plus FTP-BruteForce and test on benign plus SSH-Bruteforce, then reverse; (b) Timestamp, once the meridiem is disambiguated, giving two disjoint attack windows (FTP 10:33-12:10, SSH 14:01-15:32). Re-keyed this way the fault returns to FATAL, and note that it then merges with fault 2 - both are discharged by the same window-based split, so file them as one.

Saturation asserted in a comment, never measured

EVIDENCE

The comment in cell sixteen reads, quote, these learners saturate long before then on this data, unquote. No learning curve appears anywhere in the notebook and no score is reported at any other training size. The abstract meanwhile leads with, quote, we load at least one million real CICFlowMeter records, unquote.

SCIENTIFIC CONSEQUENCE

The abstract sells a million-row study. The experiment is a one-hundred-twenty-thousand-row study. Those differ by a factor of nine, and the justification for the gap is an unmeasured assertion in a comment. Saturation is a curve, and drawing it costs six lines.

REQUIRED REPAIR

A learning curve from one thousand to the full corpus, showing the score flattening. Or state the effective training size in the abstract instead of the download size.

DEFENSIBLE REPLY

The effective training size you say my abstract hides is in the sentence immediately after the one you quoted, it is printed by cell sixteen, and section eight says every score here is a subsample number rather than a full-corpus claim - three times, unprompted.

ORDERED CORRECTION

Strike 'The abstract sells a million-row study' and strike the alternative remedy 'state the effective training size in the abstract' - already satisfied three times over. The surviving charge is narrow and correct: an 85% discard of the training half is justified by an empirical saturation claim asserted in a code comment and never measured, and the six-line learning curve is still owed. Note also that the evaluation is not subsampled - the score is measured on all 262,144 held-out rows.

One draw, six decimal places

EVIDENCE

Cell ten sets random state to zero with the comment, quote, single seed used everywhere, unquote. The split, the subsample, the ablation re-splits and the cross-validation folds all inherit it. The eight-seed sweep that appears in the audit was run externally, on the single-feature diagnostic only, never on the reported model.

SCIENTIFIC CONSEQUENCE

A single observation is not an estimate. Reporting it to six decimals asserts a resolution the design cannot deliver, and the notebook has no way to tell you whether one point zero zero zero zero zero zero would survive a different draw.

REQUIRED REPAIR

Ten or more seeds through the full pipeline, reporting mean and standard deviation, with the headline quoted to the resolution that spread supports.

DEFENSIBLE REPLY

My headline is 1.000000 from two errors in 262,144 rows and every ablation returns 1.000000 after removing 36% of the corpus and the strongest column; a seed sweep on a saturated metric returns the same figure, which is a fact about the corpus rather than a gap in my design.

ORDERED CORRECTION

Merge into fault 10, whose REQUIRED (bootstrap the held-out score) discharges most of this one, and whose mechanism is quantified rather than asserted. If kept separate, the defensible form is about the per-family and operational figures, not the saturated headline.

The precision-recall curve is drawn and never read

EVIDENCE

Cell eighteen calls precision-recall-curve and plots it as the third panel of the first figure. Average-precision-score appears nowhere in the notebook. Across all thirty-six notebooks, a precision-recall curve is plotted thirty-six times out of thirty-six and average precision is computed zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

Be precise, because the figure is on your page. There is a precision-recall curve. What there is not is a number taken from it. You drew the diagnostic that is sensitive to false-positive count and then declined to read it, and ranked instead on the one metric that is not.

REQUIRED REPAIR

Average precision with a bootstrap interval, used to rank the four learners, plus precision at a fixed recall.

DEFENSIBLE REPLY

Compute the metric you say I refused to read: with one false positive and one false negative on this split, precision at the operating point is 95,236 over 95,237 and average precision is 0.99999 - it ranks the four learners identically and changes not one sentence in my conclusion.

ORDERED CORRECTION

Keep the true and checkable universal - the PR curve is plotted 36 times of 36 and its area computed 0 times of 36 - but drop the claim that reading it would have exposed anything on THIS notebook. Save the argument for nb04, where the same omission hides a recall of 0.599 behind an AUC of 0.996349 and the negatives-to-positives ratio is 1,845 to 1 rather than 1.75 to 1.

An operational rate estimated from one event

EVIDENCE

Cell eighteen prints an operational false-positive rate at threshold zero point five of zero point zero zero zero zero, from one benign flow flagged out of one hundred sixty-six thousand nine hundred seven. Bootstrap and confidence-interval methods appear zero times across all thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

Here is what a rate built on one event means, and you can check it. The standard exact interval for a single observed count runs from about zero point zero three to about five point six expected events. Divide those by one hundred sixty-six thousand nine hundred seven and you get a true false-positive rate somewhere between about one and a half in ten million and about three and a third in one hundred thousand. That is a factor of two hundred twenty. Scale it to a network seeing ten million benign flows a day and the notebook's operational number is consistent with anything from roughly two false alarms a day to roughly three hundred thirty. Both of those are printed as zero point zero zero zero zero.

REQUIRED REPAIR

Attach an exact Poisson or Clopper-Pearson interval to the false-positive rate, report it in events rather than as a rounded rate, and bootstrap the held-out ROC-AUC.

The notebook wrote the red-flag warning, then crowned the red flag

EVIDENCE

Cell sixteen rounds ROC-AUC to six places with the comment, quote, six decimal places, a one point zero zero zero zero zero zero is a red flag, not a win, unquote. The table then shows one point zero zero zero zero zero zero for RandomForest and one point zero zero zero zero zero zero for XGBoost. Cell twenty-four prints, quote, RandomForest three-fold cross-validated ROC-AUC equals one point zero zero zero zero plus or minus zero point zero zero zero zero, unquote, followed by the words, a small standard deviation means a stable estimate.

SCIENTIFIC CONSEQUENCE

The notebook wrote the correct interpretive rule in cell sixteen and then, eight cells later, violated it in the same document. A standard deviation of exactly zero is not stability; it is an estimator with no remaining resolution, which cannot distinguish a good model from a leaky corpus. Both readings of that number cannot be held at once, and the abstract takes the flattering one.

REQUIRED REPAIR

Treat a saturated metric as a failed measurement. Move to a metric with resolution left on this data, such as precision at high recall or expected cost, and report the score to the resolution the estimator supports.

DEFENSIBLE REPLY

My format string is four decimal places, so 0.0000 is a printing width and not a claim of zero variance; and I did not take the flattering reading of the 1.000000 - my ledger grades this corpus D, says the headline is optimistic, and tells the reader it is a benchmark number and not a deployment estimate.

ORDERED CORRECTION

Re-mechanise, and the fault becomes stronger than the one filed. Delete 'a standard deviation of exactly zero' and delete the 'violated its own rule' framing. The true defect is what that CV actually is: cell 24 sets cvX to `Xtr.iloc[:40_000]`, a slice of the same 120,000-row training subsample the reported model was fitted on, drawn from a corpus with a 0.356 exact-duplicate rate, so duplicate rows land on both sides of every fold. It validates a 26,667-row model, not the 120,000-row model in the abstract, and it is carried into a section headed 'Reproducibility and robustness' as though it validated the reported pipeline. Required: cross-validate on data disjoint from the training subsample, and report a metric with resolution left on this corpus.

No operating point was chosen; a default was inherited

EVIDENCE

Cell eighteen thresholds the probability at zero point five with a hard-coded comparison and prints the false-positive rate at that point. No cost matrix appears in any cell. Across all thirty-six notebooks, explicit cost matrices and threshold selection from a precision-recall curve each appear zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

For an intrusion detector the operating point is the result, because the cost of a missed brute-force compromise and the cost of an analyst hour are not equal and are not close. Zero point five is not a decision; it is the value scikit-learn ships with. Every operational claim in this notebook is a claim about a threshold nobody selected.

REQUIRED REPAIR

State a cost per false negative and per false positive, sweep the threshold, and report the operating point that minimises expected cost, together with the alerts per analyst per day it implies.

DEFENSIBLE REPLY

There is no threshold on this split that trades a false positive against a false negative, because there is one of each in 262,144 rows; a cost matrix would select the same point I report, and I label the figure '@0.5' rather than presenting it as a chosen operating point.

ORDERED CORRECTION

Merge into fault 13, whose REQUIRED already contains 'report the alerts per analyst per day it implies'. The method complaint - no cost model anywhere in 36 of 36 notebooks - is true and belongs in the series-level summary, not as a separate FATAL against a notebook whose detector makes two errors.

One flow in three is an attack; no network looks like that

EVIDENCE

Cell eleven prints an attack rate of zero point three six three three, and cell sixteen prints the corresponding majority baseline of zero point six three six seven. The abstract calls the attack the minority class, which it is, at over a third of all flows. Section two cites Sommer and Paxson for the proposition that closed-world scores rarely survive deployment.

SCIENTIFIC CONSEQUENCE

A capture in which one flow in three is a brute-force attempt is not a network; it is a scheduled experiment in which the benign generator was switched off for part of the day. Every derived quantity, precision above all, is a function of that rate, and the notebook reports none of them at any other rate. It cites the paper that says exactly this and then does not do the substitution.

REQUIRED REPAIR

Recompute precision and alerts per day at a stated deployment base rate, for example one attack per ten thousand flows, using the measured true-positive and false-positive rates, and report those figures beside the corpus ones.

DEFENSIBLE REPLY

Do the substitution you are asking me for: my measured false-positive rate is six in a million, so at one attack per ten thousand flows the precision is 0.943 per million flows - the base-rate correction you say I dodged leaves my conclusion where it was.

ORDERED CORRECTION

TAXONOMY Part 4 requires derived arithmetic to show its work; this fault asserts the omission without performing the calculation, and the calculation refutes the implication. Even at the upper Poisson bound from fault 10 (FPR $3.34e-5$) the precision is 0.75. What survives is only the true and modest point that a 36.33% attack rate is a schedule artifact and no quantity is reported at any other rate. Move the base-rate arithmetic argument to nb04, where the same substitution drops precision from 0.817 to 0.452.

The designated honest metric returns a constant

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, SSH-Bruteforce one point zero, FTP-BruteForce one point zero, unquote. There are exactly two families. Section nine of the notebook states that per-group recall is where you see whether detection is real, because the dominant classes do not show it.

SCIENTIFIC CONSEQUENCE

A diagnostic that returns its maximum on every group it evaluates has not diagnosed anything. The notebook's ledger even concedes this, saying the uniformity suggests the corpus is easy to separate rather than that the detector is strong. It then keeps the number in the validity ledger as though it were supporting evidence. A saturated diagnostic is a failed measurement and must be reported as one.

REQUIRED REPAIR

Report the positive count for each family with a binomial interval, and when every group saturates, state that the diagnostic has no resolution on this corpus rather than listing its values.

DEFENSIBLE REPLY

The sentence your REQUIRED asks me to write is already in my ledger: that uniformity suggests the corpus is easy to separate, not that the detector is strong - and with ninety-five thousand held-out positives the interval you want is narrower than the resolution I printed.

ORDERED CORRECTION

Strike 'It then keeps the number in the validity ledger as though it were supporting evidence' - the ledger does the opposite in the adjacent clause. Note also that the $\text{cnt} < 5$ floor drops nothing here (two families, both with tens of thousands of positives), so this is not the C5 silent-dropping defect at all and should not carry that family code. What is left is that the positive counts are not printed, which is a formatting point on this corpus. Save C5 for nb04, where the denominators are 97, 18 and 27.

The evasion is one command-line flag

EVIDENCE

No cell in the notebook perturbs any input. The strongest single feature is Fwd Seg Size Min at zero point nine nine eight four, and the second is the destination port. Both are set by the attacker's client configuration.

SCIENTIFIC CONSEQUENCE

Hydra and Medusa take a delay flag and a port flag. The two columns this model leans on hardest are exactly the two an attacker sets from the command line at no cost and with no loss of attack capability. A detector whose worst case under a one-flag change is unmeasured has not been evaluated as a security control at all.

REQUIRED REPAIR

Define a perturbation budget over attacker-controllable columns, generate evaded flows within it, and report worst-case recall under that budget.

DEFENSIBLE REPLY

My ablation table already answers that: I dropped Fwd Seg Size Min, the column you say I lean on hardest, refitted, and the held-out AUC stayed at 1.000000 - so the model demonstrably does not depend on it, and your one-flag evasion story is refuted by a number on my own page.

ORDERED CORRECTION

The correct mechanism is the harder one and it should replace the filed text: separability here is spread across many redundant generator-coupled columns - dropping the strongest changes nothing and de-duplicating 36% of rows changes nothing - so evasion requires moving the whole flow envelope rather than one flag, and the notebook measures none of that. Also concede what the notebook says: its scope paragraph restricts every number to in-distribution separability and it makes no robustness claim, so this is a gap in a security thesis rather than an overclaim to be refuted.

Nothing you remove changes anything, and that is the finding

EVIDENCE

The cell-twenty-two ablation table reads: headline as-is, one point zero; de-duplicated with thirty-six percent of rows removed, one point zero; strongest feature dropped, one point zero. The auditor extended it by also removing the destination port, the second-ranked column, and the score remained one point zero.

SCIENTIFIC CONSEQUENCE

The conclusion reads this as, so the separability is multi-feature. Turn it round. If you can delete the two strongest columns and thirty-six percent of the rows and nothing moves by even one part in a million, the two classes are separated many times over by redundant columns. That is not a hard problem with a strong solution; it is the fingerprint of a corpus in which one generator produced the benign flows and two scripted tools produced the attack flows. The notebook's ablation is the strongest evidence in the document, and it is evidence for the prosecution.

REQUIRED REPAIR

Measure class overlap directly, with a per-feature divergence or a nearest-neighbour Bayes-error bound, and state the separability as a property of the generation process rather than as a model result.

DEFENSIBLE REPLY

Read section eleven before the table you are quoting: it says in advance that if the AUC barely moves that is not vindication, that it means the classes are separable by many redundant features because the distributions barely overlap, and that this is its own generation artifact - which is your finding, printed above your evidence.

ORDERED CORRECTION

Two required changes. (a) Strike the misreading charge; the notebook states the correct reading twice. The surviving residual is that the correct reading is asserted and never quantified - no divergence, no nearest-neighbour Bayes-error bound - so keep only the REQUIRED, and note that half of that measurement is already available: I measured it, and de-duplication removes 286,854 of the 380,949 attack flows, so 75.3% of the positive class is exact repeats and the positive rate falls from 0.363 to 0.139. That is the quantitative form of 'this measures the generator, not the phenomenon', and the filing does not have it. (b) MANDATORY: strike 'The auditor extended it by also removing the destination port, the second-ranked column, and the score remained one point zero.' That sentence answers fault 1's master charge and destroys it. It is also a claim about a cell that is not in the notebook, so 'show me that run' is a live rebuttal. Choose fault 1 or this sentence; you cannot read both aloud.

No published number to stand beside

EVIDENCE

The cell-sixteen ranking table contains four learners and one majority baseline at zero point six three six seven. Section two lists Sharafaldin, Engelen and Rosay, and the caveat table states explicitly that these are not measurements reproduced on this corpus. No published figure appears in any output cell.

SCIENTIFIC CONSEQUENCE

Because the split is your own and no canonical partition exists in the notebook, one point zero zero zero zero zero zero cannot be placed beside any prior result by anyone. A number that cannot be compared cannot be wrong, and a number that cannot be wrong is not a scientific claim.

REQUIRED REPAIR

Adopt a fixed, published partition of this corpus and report your figure alongside at least one published figure computed on that same partition.

DEFENSIBLE REPLY

There is no published partition of CSE-CIC-IDS2018 to adopt: CIC ships ten day files and no train/test split, which is precisely why my literature table says those papers are not measurements reproduced on this corpus.

ORDERED CORRECTION

Replace REQUIRED. Since no canonical partition exists, the constructible demands are: publish the split indices so the number is reproducible by a third party, and report at least one published per-day figure for Wednesday-14-02-2018 alongside. Keep the true underlying point - a number that cannot be contradicted has not been tested - but do not ask for a partition the corpus does not have.

The missing experiment is a file in the same folder

EVIDENCE

The section-thirteen scope paragraph states, in the notebook's own words, that the split is random, not temporal or entity-grouped, and that every number therefore measures in-distribution separability only. The directory containing this file also contains nb thirty-six, cross-distribution transfer, built on the same loader contract with the same feature schema.

SCIENTIFIC CONSEQUENCE

Two sibling notebooks in this same folder, the DoS day and the web-attack day, use the identical sixty-eight-column schema. Training on one and testing on another is an import statement away. The capability is present in the codebase and was not applied, and the notebook wrote the scope paragraph that says why it matters.

REQUIRED REPAIR

Train on this day, test on the DoS day and the web-attack day, both directions, and report the transfer drop in the abstract.

DEFENSIBLE REPLY

You are asking me to train an FTP and SSH brute-force detector and test it on a day whose only attacks are GoldenEye and Slowloris; those label sets are disjoint, so a low score there measures class mismatch and tells you nothing about whether my brute-force result generalises.

ORDERED CORRECTION

Re-mechanise to the version that IS informative and IS constructible, and the fault returns to full strength: score this day's fitted model on the benign flows of the DoS day and the web-attack day - 996,077 and 1,048,009 rows of unseen benign traffic in the identical 68-column schema - and report the false-positive rate. That is a pure generalisation test with no class-mismatch confound, and it is the number a deployment would care about. Drop the nb36 reference (CORRECTION 4: nb36 transfers among four NetFlow-v2 corpora sharing a 43-field schema, which this corpus does not share).

Atlas proof

docs/course/security-ml-notebooks/nb02-cic-ids2018-bruteforce.ipynb · Cell 26 (1-based), Scientific conclusion, Scope; zero-based manifest cell IDs: 25=c21179c7

Both classes were generated by software

EVIDENCE

Section two cites Sharafaldin and colleagues for the corpus design, describing it as profiled benign traffic plus a labelled attack schedule; the benign side comes from the CIC B-Profile generator. The attack side is FTP brute force and SSH brute force run on a schedule against one target. The abstract describes the records as, quote, real CICFlowMeter records, unquote.

SCIENTIFIC CONSEQUENCE

The word real in that abstract sentence is doing work it cannot support. The records are real outputs of a real feature extractor run over synthetic traffic. Separating one generator's output from another generator's output is a solvable problem and a score of one is the expected outcome. It is a measurement of the two generators, and the abstract presents it as a measurement of brute-force detection.

REQUIRED REPAIR

Either evaluate on captured production traffic, or restrict every claim in the abstract and conclusion to the statement that this generator pair is separable.

DEFENSIBLE REPLY

You concede in your own sentence that these are real outputs of a real extractor, and my conclusion already restricts the claim to what you are asking for: that brute-force traffic is trivially separable in this recording, and that one in-distribution score cannot settle whether that reflects behaviour or generation.

ORDERED CORRECTION

Reduce to an editorial charge, which is all that survives: the abstract's unqualified 'real' invites a production reading that the conclusion then withdraws, so the qualification belongs in the abstract. Do not present this as a fatal misframing when the conclusion already states the restricted claim.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 262,144 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. The most important sentence in this notebook is one it wrote about itself: no number in this notebook shows what the model scores with the port column removed, treat that as an untested part of the headline, not a cleared one. I accept that sentence entirely. It is correct, it is candid, and it is fatal. The ablation harness that would have settled it runs twice in cell twenty-two and needed a third row. Add to it a score of exactly one on a test set a third of which the model has already seen, a winner chosen by a tie in a sort, and a row count that is the Excel worksheet limit, and what is left is not a result about brute-force detection. It is a very careful description of one day's export file.

CIC-IDS2018 Case Study - DoS (GoldenEye / Slowloris)

one day of CSE-CIC-IDS2018 flow records, Thursday the fifteenth of February two thousand eighteen, two HTTP denial-of-service tools run against a single victim host

HEADLINE

RandomForest, held-out ROC-AUC one point zero zero zero zero zero zero, accuracy zero point nine nine nine nine zero eight

DEFENSE

F

Selected-model estimand

DOCTORAL

6

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9

/10 · self C

D1 1 External validity	D2 1 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 5 critical (FATAL)
- 16 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

The literature scoping in this notebook is the best in the entire set of thirty-six, and I want to be specific about that. It notices that the Sharafaldin paper documents the two thousand seventeen capture and not this one, and it says so. It notices that the Centre asks users of the two thousand eighteen corpus to cite that paper anyway, and it declines to let that stand as a description. It writes, of the CICFlowMeter defect literature, that the shared extractor is grounds for suspicion and not evidence, which is exactly the right epistemic weight. It devotes a titled paragraph to a shortcut it deliberately names and does not remove. And it writes the sentence, every number above should be read as an upper bound. That is a better standard of care than most of what gets published on this dataset. It is also, as I will now show, not enough.

EXAMINER RULING

This notebook asked the right question about its own corpus, answered it by checking the file against its own download cap instead of against the upstream object, and thereby certified as a full day a file that is an Excel-truncated half day.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The entire positive class is two program invocations

EVIDENCE

The auditor's re-run gives the raw label counts as benign nine hundred ninety-six thousand seventy-seven, GoldenEye forty-one thousand five hundred eight, Slowloris ten thousand nine hundred ninety, which reproduces the printed attack rate of zero point zero five zero one exactly. Those fifty-two thousand four hundred ninety-eight attack flows come from two windows: GoldenEye from nine twenty-seven in the morning to just after ten, thirty-five minutes, and Slowloris from eleven o'clock to eleven forty-two, forty-two minutes. The notebook itself says, in section three, that both tools are HTTP denial-of-service attacks aimed at a single victim host. Cell sixteen splits those flows at random.

SCIENTIFIC CONSEQUENCE

You have two experimental units and you are reporting thirteen thousand one hundred twenty-five degrees of freedom. Every attack flow in the held-out split was produced by one of the same two program invocations, on the same machine, against the same target, within the same hour, as the attack flows in training. A random split cannot separate a concept from a process, because the process is a constant. A ROC-AUC of one under this design means the model recognised the two runs, and it is the only thing it could mean.

REQUIRED REPAIR

Group the split so that no attack window appears on both sides. With two windows the honest design is leave-one-tool-out: train on GoldenEye, test on Slowloris, and back the other way. Report both numbers.

4

CRITICAL

MEASURED

VERIFIED

MODEL RANKING WITHOUT A TEST

ORAL TOP 5

The crowned model makes fifty percent more mistakes

EVIDENCE

From the cell-sixteen table, on two hundred sixty-two thousand one hundred forty-four held-out rows: RandomForest accuracy zero point nine nine nine nine zero eight, XGBoost zero point nine nine nine nine three nine, LightGBM zero point nine nine nine nine three nine. Multiply out and the error counts are pinned uniquely. RandomForest makes twenty-four errors. XGBoost makes sixteen. LightGBM makes sixteen. RandomForest is crowned because its ROC-AUC is one point zero zero zero zero zero zero against XGBoost's zero point nine nine nine nine nine nine, a gap of one part in a million.

SCIENTIFIC CONSEQUENCE

The selection criterion and the reported operational numbers are on different axes and they disagree. Every operational figure in section nine, the confusion matrix, the false-positive rate, the per-family recalls, is computed at a threshold of zero point five, and at that threshold the selected model is fifty percent worse than two models the notebook discarded. The single part in a million that decided the ranking is smaller than the resolution the study can support, and it bought eight extra errors.

REQUIRED REPAIR

McNemar's test on the paired predictions of the top three models. Rank on the metric you will actually deploy at, and if the models do not separate, say so and report all three.

5

CRITICAL

UNTESTED

VERIFIED

NO TEMPORAL HOLDOUT ON TIMESTAMPED DATA

ORAL TOP 5

The audit used the column you deleted

EVIDENCE

Cell eleven sets the drop list to Label, Timestamp, y and family. Cell sixteen then splits at random with stratification. Across all thirty-six notebooks in this series, a temporal split appears zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

This one has an edge the others do not. The auditor found the truncation by parsing that same Timestamp column, establishing that the file stops at twelve fifty-nine fifty-nine with a hundred and eight thousand rows still in the final hour. So the column was present, parseable and load-bearing, and the notebook's own provenance discussion would have been corrected by reading it. The notebook named the column in order to delete it, and the deletion cost it both the temporal experiment and the truncation finding.

REQUIRED REPAIR

Sort by Timestamp, train on the morning and test on the afternoon, and report the drop. Separately, parse the column and report the capture span in the provenance table.

You checked the file against the wrong thing

EVIDENCE

The provenance table says, in the notebook's own words, that the download caps the stream at one point two million lines, and that this day's CSV is smaller than the cap, so the cap does not bind. That is true and it is the wrong test. The auditor ran the right one. The cached file is byte-identical to the S3 object, three hundred seventy-five million nine hundred forty-five thousand eight hundred ninety-nine bytes on both sides, so the download is innocent. A line count of the file gives one million forty-eight thousand five hundred seventy-six lines, which is one header plus one million forty-eight thousand five hundred seventy-five rows, and one million forty-eight thousand five hundred seventy-six is exactly two to the twentieth power, the Excel worksheet row limit. The Timestamp column runs from one in the morning to twelve fifty-nine and fifty-nine seconds and stops, with one hundred eight thousand one hundred seventy-nine rows still inside that final hour and no taper at all.

SCIENTIFIC CONSEQUENCE

A capture does not end at a round binary number with a hundred thousand flows still arriving in its last hour. This file is a half day that fitted inside a spreadsheet. Now the consequence, which is the part that kills the study, and it follows from the timestamps alone. Both attack windows had closed before noon: GoldenEye ran nine twenty-seven to just after ten, Slowloris eleven o'clock to eleven forty-two. Both are complete inside the surviving portion. Benign traffic, meanwhile, was still arriving at one hundred eight thousand rows in the final hour when the file stopped. So at the moment of the cut the numerator was finished and the denominator was still growing. The printed attack rate of five point zero one percent is therefore an upper bound on the day's true rate, and the majority baseline of ninety-four point nine nine percent is a lower bound, and where both of them landed was decided by a spreadsheet row limit. Every accuracy figure in this notebook is measured against a baseline that Excel determined.

REQUIRED REPAIR

Report the true corpus size from the upstream object, or state in the abstract that this file is a truncated export and that the attack rate is therefore an artifact of the export. Then recompute the baseline on the full day.

DEFENSIBLE REPLY

Look at the column you parsed. My SSH sibling file records an attack CIC scheduled at 14:01 as 02:01:21 and this corpus family's XSS attack, scheduled at 13:00, as 01:01:04 - the Timestamp field is a twelve-hour clock with no AM or PM, so it cannot exceed 12:59:59 and your 'no taper' is arithmetic, not evidence. My file contains 389,349 afternoon rows, and row 150,001 is at 05:14 while row 200,001 is at 09:51, so it is not in time order and your claim about which flows were cut does not follow.

ORDERED CORRECTION

Keep the defect, replace the consequence. KEEP: the notebook settles the truncation question against its own download cap instead of against the upstream object; the file is 1,048,576 lines (2^{20}) with a complete trailing

newline so head -n 1200000 provably never engaged; the cached file is byte-identical to S3 at 375,945,899 bytes; and the cell-11 honesty gate 'assert len(df) >= 1_000_000' passes only because a spreadsheet row limit happens to sit above one million. DELETE the whole timestamp paragraph and the 'numerator finished, denominator growing, therefore an upper bound' inference. The correct consequence is unsigned and still fatal: this corpus is an arbitrary 1,048,575-row subset of a day whose true flow count is unknown, so the printed attack rate of 0.0501 and the 0.9499 baseline that every accuracy figure is judged against are properties of the export, with the direction of the bias unmeasured and unmeasurable from this file.

3

MAJOR

UNTESTED

REMECHANISED

FEATURES ENCODE THE GENERATOR, NOT THE ADVERSARY

ORAL TOP 5

You called it an upper bound and printed it as the result

EVIDENCE

Section three carries a titled paragraph, a shortcut deliberately named, not removed. It says the destination port acts as a cheap negative-class filter, that it encodes the capture schedule rather than attack behaviour, that a model can discard a large share of benign flows without learning anything about denial of service, and then, in the notebook's own words, this notebook does not quantify that effect. The conclusion repeats it and adds, for that reason alone, every number above should be read as an upper bound. The ranking table then prints one point zero zero zero zero zero.

SCIENTIFIC CONSEQUENCE

An upper bound with no lower bound is not a measurement. The notebook has told you the reported quantity is inadmissible and has told you it declined to measure the admissible one. The ablation harness in cell twenty-two already refits the model twice with columns removed; adding a third configuration that drops the port and the protocol is one line of code. So the required experiment is one line, the notebook wrote three paragraphs about why the experiment matters, and it did not write the line.

REQUIRED REPAIR

Drop the destination-port and protocol columns, refit, and report that score as the headline. If it moves, the original was a schedule artifact and must be withdrawn.

DEFENSIBLE REPLY

Run my own section-ten audit on the column you are prosecuting: Dst Port scores 0.5483 and ranks fifty-second of sixty-eight on this day - one of the least informative columns in the matrix, because the benign traffic here is largely HTTP too. My section-three paragraph over-warned, and your fault is quoting my over-warning back at me as if I had measured it.

ORDERED CORRECTION

Re-mechanise, and it becomes a more interesting charge than the one filed. The defect is not that a schedule shortcut inflated the score - it did not, and the notebook's own apparatus refutes it at 0.5483. The defect is that the notebook downgraded its entire result to 'an upper bound' on the strength of a shortcut mechanism it asserted in three paragraphs and never measured, when measuring it was one call to the audit already written in cell 20. A caveat asserted rather than measured is the same methodological failure as a claim asserted rather than measured, and here it happens to be wrong. Also MANDATORY: this fault and fault 17 cannot both be read - 17 asserts the auditor already ran the Dst Port ablation and got 1.0.

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

6 MAJOR DOCUMENTED DOWNGRADED THE CORPUS IS NOT THE EXPERIMENT

Saturation asserted in a comment, never measured

EVIDENCE

The cell-sixteen comment reads, quote, these learners saturate long before then on this data, unquote. No learning curve appears in any cell and no score is reported at any other training size. The notebook does correctly state that scoring uses the full two hundred sixty-two thousand one hundred forty-four held-out rows, which is more than most of the set discloses.

SCIENTIFIC CONSEQUENCE

Saturation is measurable in six lines and it is the entire justification for discarding six sevenths of the available training data. Asserting it in a comment while leading the abstract with the million-row figure describes one experiment and runs another.

REQUIRED REPAIR

A learning curve to saturation, or the effective training size stated in the abstract.

DEFENSIBLE REPLY

My abstract distinguishes the two numbers in consecutive sentences and section eight repeats it: the training set is bounded at 120,000 and the evaluation set is the entire 262,144-row holdout. The effective size you want stated is stated twice, and you concede it in your own evidence paragraph.

ORDERED CORRECTION

Strike 'Asserting it in a comment while leading the abstract with the million-row figure describes one experiment and runs another' - this notebook is the one in the set that gets that disclosure right, and the fault says so. The surviving charge is narrow: the saturation claim that justifies discarding six sevenths of the training half is an unmeasured assertion in a code comment, and the six-line learning curve is owed.

One draw, six decimals

EVIDENCE

Cell ten sets random state to zero with the comment, quote, single seed used everywhere, unquote, and the split, the subsample, the ablation re-splits and the cross-validation folds all inherit it.

SCIENTIFIC CONSEQUENCE

A single observation cannot support six significant figures, and the notebook has no way to know whether the one-part-in-a-million gap that decided its winner would survive a second draw. It almost certainly would not.

REQUIRED REPAIR

Ten or more seeds through the full pipeline, with mean and standard deviation, and the headline quoted at that resolution.

DEFENSIBLE REPLY

You have already filed the one-part-in-a-million gap as fault four and the absence of intervals as faults nine and ten; and you assert my ranking almost certainly would not survive a redraw without computing a single redraw, which is the arithmetic you are demanding of me.

ORDERED CORRECTION

Merge into fault 4 - a seed sweep and a McNemar test address the same question about the same $1e-6$ gap - and delete 'It almost certainly would not', which is an assertion of the kind the fault is prosecuting. If retained separately, the defensible target is the per-family recalls and the operational rate, not the saturated headline.

The precision-recall curve is drawn and never read

EVIDENCE

Cell eighteen calls precision-recall-curve and draws it as the third panel. Average-precision-score appears nowhere. Across the thirty-six notebooks, the curve is plotted thirty-six times and its area is computed zero times.

SCIENTIFIC CONSEQUENCE

At a five percent positive rate, ROC-AUC is the metric least sensitive to false-positive volume and precision is the metric that decides whether an analyst can use the thing. You plotted precision against recall on the same figure as the ROC curve and then took your headline from the other panel.

REQUIRED REPAIR

Average precision with a bootstrap interval as the ranking metric, plus precision at a fixed recall.

DEFENSIBLE REPLY

Compute the number you say I refused to read: five false positives and nineteen false negatives give a precision of 0.9996 at my operating point and an average precision of about 0.9995. It ranks the four learners in the same order and changes nothing in my conclusion.

ORDERED CORRECTION

Keep the true universal - the PR curve is plotted 36 of 36 times and its area computed 0 of 36 - but drop 'precision is the metric that decides whether an analyst can use the thing' as applied here, because on this split precision is 0.9996. The argument belongs to nb04, where the identical omission conceals a recall of 0.599 behind an AUC of 0.996349.

An operational rate built on five events

EVIDENCE

Cell eighteen prints an operational false-positive rate at threshold zero point five of zero point zero zero zero, from five benign flows flagged out of two hundred forty-nine thousand nineteen. Bootstrap and confidence intervals appear zero times across all thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

Five events. The standard exact interval on a count of five runs from about one point six to about eleven point seven. Divide by two hundred forty-nine thousand nineteen and the true false-positive rate lies somewhere between about six and a half in a million and about four point seven in a hundred thousand, a factor of seven. On a network seeing ten million benign flows a day, that is somewhere between sixty-five and four hundred sixty-nine false alarms a day. The notebook prints all of that as zero point zero zero zero zero.

REQUIRED REPAIR

Exact Poisson intervals on the false-positive count, reported in events per unit volume rather than as a rounded rate, and a bootstrap interval on the held-out score.

No operating point was chosen; a library default was inherited

EVIDENCE

Cell eighteen thresholds at zero point five with a hard-coded comparison. No cost matrix appears anywhere in the notebook. Across the thirty-six notebooks, explicit cost matrices and threshold selection from a precision-recall curve each appear zero times.

SCIENTIFIC CONSEQUENCE

For a denial-of-service detector the cost of a late detection and the cost of a false page at three in the morning are not equal and are not close. The operating point is the result. Zero point five is the value the library ships with, and choosing it is not a decision anyone in this notebook made.

REQUIRED REPAIR

State the cost of a false negative and of a false positive, sweep the threshold, and report the expected-cost-minimising operating point with the alerts per analyst per day it implies.

DEFENSIBLE REPLY

You are filing the same remedy twice: fault eleven and fault twelve both end by demanding alerts per analyst per day. And there is very little for a threshold sweep to trade on a split with five false positives and nineteen false negatives.

ORDERED CORRECTION

Merge into fault 12. The method complaint - no cost matrix and no threshold selection in 36 of 36 notebooks - is true and belongs in the series-level summary. Keep the 'false page at three in the morning' framing there, where it is rhetorically strong and not attached to a specific inert corpus.

A base rate set by a spreadsheet limit

EVIDENCE

Cell eleven prints an attack rate of zero point zero five zero one and cell sixteen the corresponding majority baseline of zero point nine four nine nine. The audit established that both attack windows had closed before the file stops, while benign traffic was still arriving at one hundred eight thousand rows in the final hour, so the printed rate is an upper bound on the day's true rate. Section two cites Sommer and Paxson for the proposition that closed-world scores rarely survive deployment.

SCIENTIFIC CONSEQUENCE

Chain the two together and it is worse than an unrealistic base rate. The base rate is an artifact of where the export stopped, and the accuracy baseline is derived from it, and the headline accuracy is compared against that baseline. A number whose comparator is a spreadsheet constant is not a result. And no precision figure is ever recomputed at a base rate anyone would meet in production.

REQUIRED REPAIR

Recompute the baseline on the untruncated day, and report precision and alerts per day at a stated deployment base rate using the measured true-positive and false-positive rates.

DEFENSIBLE REPLY

Do the substitution you are asking for: my false-positive rate is twenty in a million and my true-positive rate is 0.9986, so at one attack per ten thousand flows the precision is 0.83 per million flows. And your claim that my base rate is a signed artifact of the export rests on reading a twelve-hour timestamp as a twenty-four-hour one.

ORDERED CORRECTION

Delete the first half of WHY FATAL - the chain from the truncation to a signed bias in the base rate is the inference refuted under fault 1 - and delete the duplicate of fault 1's REQUIRED ('recompute the baseline on the untruncated day'), which is unbuildable in any case because the untruncated day does not exist as a published object. What survives is the modest true point that no quantity is reported at any base rate other than the corpus one, and TAXONOMY Part 4 requires that the substitution be shown; when it is shown, it comes out at 0.833.

Slowloris flows differ in one counter and count as distinct

EVIDENCE

Cell twenty rounds to six decimals and uses set membership on tuples, reporting contamination of zero point one zero four and an exact-duplicate rate of zero point one five six.

Cell twenty-two calls drop-duplicates, which is exact equality, and reports that removing sixteen percent of rows leaves the score at one point zero zero zero zero zero.

SCIENTIFIC CONSEQUENCE

A Slowloris run opens thousands of connections that differ only in ephemeral port and a byte counter. Exact matching classifies every one of those as a fresh row, so the measured ten point four percent contamination is a floor, not an estimate, and the sixteen percent de-duplication ablation removed the easiest duplicates only. The notebook's assurance that duplicates are not propping up the score rests on a test that cannot see the duplicates that matter.

REQUIRED REPAIR

Nearest-neighbour distance distribution across the split boundary, the fraction of held-out rows within a small epsilon of a training row, and a re-run de-duplicated at that tolerance.

DEFENSIBLE REPLY

We measured contamination under the strictest definition available and got 10.4%, a grade B and the second lowest in the series; you are speculating about a quantity neither of us has measured, and my ablation already shows that removing 16% of rows does not move the score.

ORDERED CORRECTION

None; the severity is right and the REBUTTAL field is honest. One usable sharpening: the contamination is measured against Xtr after cell 16 reassigns it to the 120,000-row subsample, so 10.4% of held-out rows exactly match one of only 120,000 training rows rather than one of 786,431. That is a stronger statement of the same finding and it is free.

Two groups, both saturated, no denominators

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, DoS attacks Slowloris zero point nine nine six, DoS attacks GoldenEye zero point nine nine nine, unquote. There are exactly two groups. No positive count appears next to either value anywhere in the notebook. The conclusion says the weakest group sits at zero point nine nine six, which is where detection is thinnest.

SCIENTIFIC CONSEQUENCE

Section nine of this notebook argues that per-group recall is where you see whether detection is real, because the dominant classes do not show it. With two groups, both above zero point nine nine, and no denominators, the diagnostic has returned its ceiling and told you nothing. Calling zero point nine nine six the place where detection is thinnest is treating the top of the scale as a finding.

REQUIRED REPAIR

Print the positive count with every recall, attach a binomial interval, and when every group saturates, state that the diagnostic has no resolution rather than listing its values.

DEFENSIBLE REPLY

Attach the interval you are asking for: with thirteen thousand held-out positives split between two families, the binomial band on 0.996 is about two parts in a thousand. It does not alter the reading, and the denominators you say are missing are recoverable from the two counts I print.

ORDERED CORRECTION

Narrow to the one part that is genuinely wrong and drop the rest: the conclusion nominates 0.996 as 'where detection is thinnest', which treats the top of the scale as a finding on a diagnostic that has saturated. Drop the interval demand, which is inert at this sample size, and drop the C5 family code - the cnt<5 floor discards nothing here (two families, 41,508 and 10,990 flows), so this is not the silent-dropping defect.

Slowloris has a delay flag; nobody turned it

EVIDENCE

No cell perturbs any input. The strongest single feature is Fwd Seg Size Min at zero point nine eight five two, and the second-ranked column is the destination port. The notebook's own section one says these attacks perturb flow-timing features, and adds that no controlled volume-versus-timing experiment is run here.

SCIENTIFIC CONSEQUENCE

Slowloris is parameterised by connection count and inter-packet delay. Those are the exact quantities the flow features encode. An attacker who slows the tool moves the flows toward the benign distribution at no cost to the attack, and the worst-case recall under that change is unmeasured. The notebook even identifies volume versus timing as the question and declines to run it.

REQUIRED REPAIR

A perturbation budget over timing and rate columns, evaded flows generated within it, and worst-case recall reported under that budget.

Removing the two strongest columns changes nothing

EVIDENCE

The cell-twenty-two table reads: headline one point zero zero zero zero zero zero; de-duplicated with sixteen percent removed, one point zero zero zero zero zero zero; strongest feature dropped, zero point nine nine nine seven six nine. The auditor extended it by removing the destination port as well and the score stayed at one point zero. Nothing was ever measured about class overlap.

SCIENTIFIC CONSEQUENCE

The conclusion says, so the separability is multi-feature. Read it the other way, which is the correct way. If you can delete the two strongest columns and sixteen percent of the rows and the score moves by two parts in ten thousand, then the two classes barely overlap and many redundant columns each separate them alone. That is the signature of a corpus in which one generator produced the benign side and two tools produced the attack side. The notebook comes close to saying this and then keeps the one point zero as its headline.

REQUIRED REPAIR

Measure the overlap: a per-feature divergence or a nearest-neighbour Bayes-error bound. Report separability as a property of the generation process.

DEFENSIBLE REPLY

You say I come close to your reading and then keep the headline. My section thirteen says that the multi-feature result is a statement about how this corpus was generated - two attack tools, one victim, two narrow time windows against synthetic benign traffic - and that this is not a clean bill of health. That is your finding, in my words, in my conclusion.

ORDERED CORRECTION

Two required changes. (a) Strike 'The notebook comes close to saying this and then keeps the one point zero as its headline'; it says it outright, twice. What survives is only that the correct reading is stated and never quantified, so keep the REQUIRED (per-feature divergence or a nearest-neighbour Bayes-error bound) and drop the misreading charge. (b) MANDATORY: strike 'The auditor extended it by removing the destination port as well and the score stayed at 1.0.' That sentence answers fault 3's master charge, and it is a claim about a cell that is not in the notebook. Choose fault 3 or this sentence.

No published number to stand beside

EVIDENCE

The cell-sixteen table contains four learners plus a majority baseline at zero point nine four nine nine. Section two lists Sharafaldin, Engelen and Rosay and states that these are not measurements reproduced on this corpus, which is honest. No published figure appears in any output.

SCIENTIFIC CONSEQUENCE

The notebook has correctly established that no prior work describes this exact capture. That makes a canonical partition more necessary, not less, because without one there is no external number your one point zero zero zero zero zero zero can contradict. A claim that cannot be contradicted has not been tested.

REQUIRED REPAIR

Adopt a fixed published partition of this corpus and report your figure beside at least one published figure computed on that same partition.

DEFENSIBLE REPLY

You have just told the room that no prior work describes this capture, and then asked me to adopt its canonical published partition. CIC ships ten day files and no split; there is nothing to adopt.

ORDERED CORRECTION

Replace REQUIRED with the two demands that are constructible: publish the split indices so a third party can reproduce the number, and report at least one published per-day figure for Thursday-15-02-2018 alongside. Keep the underlying point - a claim that cannot be contradicted has not been tested - but do not ask for a partition the corpus does not have.

The missing experiment is a file in the same folder

EVIDENCE

The section-thirteen scope paragraph states, in the notebook's own words, that the split is random, not temporal or entity-grouped, and that every number therefore measures in-distribution separability only. The same directory contains nb thirty-six, cross-distribution transfer, plus two sibling CIC notebooks using the identical sixty-eight-column schema.

SCIENTIFIC CONSEQUENCE

The brute-force day and the web-attack day sit beside this file with the same feature schema. Training on this day and testing on those is an import away, and it is the only experiment that would turn a one point zero into evidence. The apparatus exists, in the same folder, and was not used.

REQUIRED REPAIR

Train here, test on the sibling days and on nb thirty-six's transfer protocol, both directions, and report the drop in the abstract.

DEFENSIBLE REPLY

You want me to train a GoldenEye and Slowloris detector and test it on a day whose only attacks are FTP and SSH brute force. Those label sets are disjoint; a low score there measures class mismatch and says nothing about whether my result generalises.

ORDERED CORRECTION

Re-mechanise to the constructible and informative version and the fault returns to full strength: score this day's fitted model on the benign flows of the sibling days - 667,626 and 1,048,009 rows of unseen benign traffic in the identical 68-column schema - and report the false-positive rate. That is a pure generalisation test with no class-mismatch confound. Drop the nb36 reference under CORRECTION 4.

The notebook names the negative class synthetic and keeps the score

EVIDENCE

The section-thirteen conclusion describes the setting, in the notebook's own words, as two attack tools, one victim, two narrow time windows, set against profiled synthetic benign traffic. Section two describes the B-Profile generation methodology.

SCIENTIFIC CONSEQUENCE

State the experiment in one sentence with the notebook's own nouns: two program invocations against one host, discriminated from a traffic generator's output, using a random split, on an Excel-truncated file. The reported quantity is the distance between two pieces of software. Nothing in it is about denial-of-service detection, and the abstract presents it as though it were.

REQUIRED REPAIR

Evaluate on captured production traffic, or restrict every claim to the statement that this generator pair is separable in this export.

DEFENSIBLE REPLY

The one-sentence restatement you want is my own conclusion: two attack tools, one victim, two narrow time windows, set against profiled synthetic benign traffic - I wrote those nouns, you are quoting them back to me, and I follow them with the statement that this is not a clean bill of health.

ORDERED CORRECTION

This fault is quoting the notebook's own self-criticism as though it were the professor's discovery, which reads as unserious under TAXONOMY Part 3. Reduce to the residual: the restriction lives in section 13 while the abstract still frames the study as a question about detection, so move the restriction into the abstract. Do not present it as a fatal misframing.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 262,144 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. This notebook did more careful work on its own provenance than any other in the set, and it is the provenance that destroys it. It asked whether its file was truncated, compared the file to its own download cap, found the cap slack, and concluded correctly that the cap did not bind, and then stopped one step short of the upstream object. The file is exactly two to the twentieth power lines, it ends at one minute to one in the afternoon with a hundred and eight thousand flows still arriving in that last hour, and both attack windows had already closed before it stopped. So the attack rate it prints is an upper bound fixed by where the export ran out of rows, the baseline derives from that rate, and the headline is compared to the baseline. Underneath all of that sits a positive class that is two program invocations against one machine, split at random. That is not a denial-of-service detector. It is a very precise measurement of a half-exported file.

CIC-IDS2018 Case Study - Web Attacks (Brute Force / XSS / SQLi)

one day of CSE-CIC-IDS2018 flow records, Friday the twenty-third of February two thousand eighteen, in which the entire held-out attack class is one hundred forty-two flows

HEADLINE

XGBoost, held-out ROC-AUC zero point nine nine six three four nine, accuracy zero point nine nine nine seven one

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 2 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 10 critical (FATAL)
- 13 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This is the strongest of the four and it earned a perfect ten from the friendly audit, so let me be exact about what it does well. Its single-feature check comes back at zero point seven four five seven, a genuine A, which means that unlike its siblings this notebook is really not a one-column result. Its section one opens by saying that at five hundredths of one percent attack prevalence, aggregate accuracy is meaningless, and it says so before any number appears. And it does something I have almost never seen in a published paper: when de-duplication raises its score rather than lowering it, it explains why, correctly, saying that collapsing duplicates removes the hardest rows because identical vectors carrying conflicting labels get resolved to one label, so the de-duplicated task is easier and not cleaner. That is a subtle and correct piece of reasoning. Everything I am about to say is said with that on the record.

EXAMINER RULING

The entire evidential base of this notebook is one hundred forty-two attack flows inside a file that stops at the Excel worksheet row limit, and the model catches eighty-five of them, which is a fact the headline metric is specifically constructed not to show you.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

A coin flip sits in the ranking table, unremarked

EVIDENCE

The cell-sixteen ranking table, fourth row, reads LightGBM, accuracy zero point nine nine six zero three seven, ROC-AUC zero point five one four two five eight. The majority baseline in the row beneath it reads accuracy zero point nine nine nine five. Now the arithmetic, which you can do in your head. Two hundred sixty-two thousand one hundred forty-four held-out rows times three point nine six three thousandths gives one thousand thirty-nine errors for LightGBM. The trivial rule, always say benign, makes one hundred forty-two errors. So LightGBM is seven times worse than saying nothing at all, and its ROC-AUC of zero point five one four is one and a half points from a coin.

SCIENTIFIC CONSEQUENCE

The section-eight preamble of this very notebook says, and I quote, on imbalanced data a zero point nine nine accuracy can be worse than always guessing the majority class, judge each model against that baseline, not against zero point five. The notebook wrote the rule, printed the exact case the rule was written for, put the baseline in the adjacent row so the comparison is unavoidable, and said nothing. A learner at chance on a dataset where three of its siblings clear zero point nine six is not a ranking entry; it is a diagnostic that something is wrong with the fitting, the encoding or the class balance. It went into the table and out of the conversation.

REQUIRED REPAIR

Flag any model scoring below the majority baseline, exclude it from the ranking, and investigate it. A learner at chance where its peers are at zero point nine nine is evidence about the pipeline, not a row.

Zero point nine nine six, and it catches eighty-five of one hundred forty-two

EVIDENCE

Here is the derivation, entirely from printed numbers, and you can check every step. Cell sixteen prints held-out two hundred sixty-two thousand one hundred forty-four. Cell eighteen prints nineteen benign flagged of two hundred sixty-two thousand two. Subtract and the attack class in the held-out split is one hundred forty-two flows. Cell sixteen prints XGBoost accuracy zero point nine nine nine seven one zero, and two hundred sixty-two thousand one hundred forty-four times two hundred ninety millionths gives exactly seventy-six errors. Nineteen of those are false positives, so fifty-seven are misses. One hundred forty-two minus fifty-seven is eighty-five true positives. Recall is eighty-five over one hundred forty-two, which is zero point five nine nine. Precision is eighty-five over one hundred four, which is zero point eight one seven.

SCIENTIFIC CONSEQUENCE

The number in the abstract is zero point nine nine six three four nine. The number that describes what the detector does is zero point five nine nine. It misses two web attacks in every five. Both of those numbers are derivable from the same three printed quantities, and the notebook reports only the flattering one. And this is exactly what the omitted metric would have shown: the precision-recall curve is computed and drawn in the third panel of cell eighteen, and its area is never computed, never printed and never used. Across all thirty-six notebooks in this series that curve is plotted thirty-six times out of thirty-six and its area is computed zero times out of thirty-six. You drew the diagnostic that would have exposed this and declined to read it.

REQUIRED REPAIR

Report precision, recall and average precision with intervals as the headline. State the confusion counts in text, not only inside a heat map.

You attributed the row count to a bound that never engaged

EVIDENCE

The section-three provenance table gives the row count as, in the notebook's own words, at least one million flow records, bounded S3 slice. The bound is in cell eleven: the download pipes through head with a cap of one million two hundred thousand lines. The loader prints one million forty-eight thousand five hundred seventy-five rows. One million forty-eight thousand five hundred seventy-five is one hundred fifty-one thousand four hundred twenty-five below the cap, so the cap did not cut anything and cannot be the reason for the count. That count is two to the twentieth power minus one, which is the Excel worksheet row limit minus a header line. And the two sibling notebooks in this series, on two different capture days, print the identical count. On the DoS day the auditor confirmed the cached file is byte-identical to the S3 object, so the truncation is upstream of the download.

SCIENTIFIC CONSEQUENCE

The provenance table states a cause for its row count, and that cause could not have operated. Notice also that the friendly audit gave this criterion a full mark and called the description conservative rather than overstated. It is neither. It is an incorrect causal claim about the notebook's own data, and it is the claim that stopped anyone checking further. And the honesty gate in cell eleven asserts that the corpus has at least one million rows, which passes only because a spreadsheet row limit happens to exceed one million. The course requirement is being met by a Microsoft constant. Every quantity downstream, the attack rate of five ten-thousandths, the baseline of zero point nine nine nine five, the one hundred forty-two held-out attacks, is a property of however much of Friday fitted above row one million forty-eight thousand five hundred seventy-six.

REQUIRED REPAIR

Run word-count on the cached file, compare its byte size against the S3 object with a head request, and parse the Timestamp column to see whether the file ends mid-day with no taper, exactly as was done for the DoS day. Then report the true corpus size and withdraw the bounded-slice attribution.

Three recalls, no counts, one hundred forty-two events between them

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, Brute Force Web zero point five three six, SQL Injection zero point five five six, Brute Force XSS zero point eight five two, unquote. No positive count appears next to any of them, anywhere in the notebook. The total held-out attack class, derived from the two printed figures two hundred sixty-two thousand one hundred forty-four and two hundred sixty-two thousand two, is one hundred forty-two flows, and those three families partition it. The count-five floor in the same loop guarantees only that each family has at least five. Bootstrap and confidence intervals appear zero times across all thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

Zero point five three six on a denominator of twenty-eight and zero point five three six on a denominator of a hundred and twenty are different scientific claims, and the notebook does not let the reader tell which one it made. With one hundred forty-two events split three ways, the largest possible denominator for any single family is one hundred forty-two and the smallest permitted is five. A binomial interval on a proportion near one half with a denominator in the low tens spans roughly from a quarter to three quarters. The conclusion names zero point five three six as where detection is thinnest. On this evidence it is not distinguishable from a coin toss.

REQUIRED REPAIR

Print the positive count beside every per-family recall and attach a Clopper-Pearson interval to each. If the interval spans one half, say the family is unscorable at this sample size.

Your own cross-validation says zero point nine seven

EVIDENCE

Cell sixteen prints XGBoost held-out ROC-AUC zero point nine nine six three four nine. Cell twenty-four prints, quote, XGBoost three-fold cross-validated ROC-AUC equals zero point nine seven four six plus or minus zero point zero three five two, unquote. The difference between the two is zero point zero two one eight. The abstract and the validity ledger carry the first figure. Nothing anywhere in the notebook mentions that the two disagree.

SCIENTIFIC CONSEQUENCE

These are two estimates of the same quantity from the same data and the same model family, and they are two points apart. That is not a rounding difference; it is larger than the entire distance between the winner and the runner-up on any axis. The notebook's own cross-validation is telling you that the headline is not stable, and the standard deviation of zero point zero three five two says the fold-to-fold spread is thirty-five thousand times the resolution at which the headline is quoted. When a study produces two numbers for one quantity and reports the larger without remark, that is a selection, not a measurement.

REQUIRED REPAIR

Report both estimates together with intervals, explain the gap, and quote the headline at the resolution the spread supports. If they cannot be reconciled, the claim is unresolved and must be stated as unresolved.

DEFENSIBLE REPLY

Those are not two estimates of one quantity. My cross-validation refits on 26,667 rows, which at a positive rate of five ten-thousandths is roughly thirteen attack examples per fold; my reported model is fitted on 120,000 rows with roughly sixty. A smaller model scoring lower is a learning curve, not a selection.

ORDERED CORRECTION

Strike 'These are two estimates of the same quantity from the same data and the same model family' and strike 'that is a selection, not a measurement' - both mis-state what cell 24 computes. The residual is real and should be the whole fault: the notebook's only dispersion figure is 0.0352, it is never set beside the headline, and the headline is quoted to six decimals against it. Note the arithmetic in the filing IS correct (0.0352 divided by $1e-6$ is 35,200), so keep that line. Note also that this fault and fault 14 are the same root defect seen from two sides - the training subsample is too small in POSITIVES - and 14 is the better-mechanised of the two.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You said you would report PR curves, not accuracy, and did the reverse

EVIDENCE

The section-thirteen conclusion says, and I quote exactly, that is why we report per-family recall and PR curves, not accuracy. The ranking table in cell sixteen has an accuracy column. The validity ledger's first sentence is, majority-class baseline accuracy zero point nine nine nine five, the accuracy column must clear that bar to mean anything. And the precision-recall curve, while drawn, yields no number anywhere in the notebook: average precision is never computed.

SCIENTIFIC CONSEQUENCE

Both halves of that sentence are false about the document containing it. Accuracy is reported, in a table, and made the basis of the baseline comparison. Precision-recall is not reported at all, because a curve without an area is a picture. This is the master fault of the set in its purest form: the notebook knows the correct methodology, writes it down in the imperative, and does not follow it in the same file.

REQUIRED REPAIR

Delete accuracy from the ranking table at this base rate, compute average precision, rank on it, and make the conclusion's sentence true.

DEFENSIBLE REPLY

Both halves are not false. The PR curve is drawn and labelled in the third panel, and the accuracy column is accompanied in the same ledger by the sentence that it must clear 0.9995 to mean anything - which is a warning about accuracy, not a use of it.

ORDERED CORRECTION

There is a cleanly true version of this fault available and the filing does not make it. The conclusion says 'that is why we report per-family recall and PR curves, not accuracy', and the headline metric of the entire study is ROC-AUC - which is neither of the two things that sentence names. That is unanswerable and needs no argument about what counts as reporting a curve. Rebuild on it, and merge the 'average precision is never computed' limb into fault 2, where it already appears.

Flow statistics cannot see a SQL payload

EVIDENCE

The sixty-eight modelled columns are CICFlowMeter flow features: packet counts, byte counts, inter-arrival times, flag counts, segment sizes. The strongest single feature is Backward Packets per second at zero point seven four five seven. Section one names the task as detecting Brute-Force-Web, XSS and SQL injection.

SCIENTIFIC CONSEQUENCE

A SQL injection is a string in an HTTP request body. Nothing in the sixty-eight columns can observe that string. So whatever the model has learned, it has not learned to recognise injection; it has learned the timing and volume envelope of the particular automated tool the Centre ran on that Friday afternoon. An attacker who issues the same injection by hand, or from a different tool, presents a different envelope and the same payload. The construct being measured is tool cadence, and the construct being claimed is web-attack detection.

REQUIRED REPAIR

Ablate the columns that encode tool cadence and report what remains, or state in the abstract that the detector discriminates one tool's traffic shape and not the attack class.

DEFENSIBLE REPLY

Flow-level timing and volume statistics carry real signal about automated web tooling, which is a legitimate detection surface in its own right, and we recall 85% of the XSS family on exactly those features without ever claiming to inspect payloads.

ORDERED CORRECTION

The counter-rebuttal is now available and should be prepared: the 85% the student will cite is 23 of 27 held-out flows (I recovered the denominators from the printed numbers), with a Clopper-Pearson interval of [0.663, 0.958]. So the answer is 'your best evidence for a behavioural detection surface is twenty-three events'. That converts the exchange from a stand-off into a win, and it costs one line.

The textbook case for class weighting, and it is never applied

EVIDENCE

Cell eleven prints an attack rate of zero point zero zero zero five. Cell sixteen constructs LogisticRegression, RandomForest, XGBoost and LightGBM entirely at hand-set constants that were never searched. Across all thirty-six notebooks in this series, class-weight and scale-pos-weight appear zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

Section one of this notebook opens by declaring that this is a severe class-imbalance problem. The first-line remedy for severe class imbalance in every one of these four libraries is a single keyword argument. Not one of the four models received it. And the consequence is visible in the same document: the winner recalls fifty-nine point nine percent of attacks at the default threshold, and the learner that collapsed to chance, LightGBM, is precisely the one you would expect to collapse when a boosted ensemble is fitted to a one-in-two-thousand positive rate with no weighting. The notebook diagnosed the disease in section one and prescribed nothing.

REQUIRED REPAIR

Set scale-pos-weight or class-weight balanced on every learner, refit, and report the change in recall and precision.

A default threshold on a one-in-two-thousand class

EVIDENCE

Cell eighteen thresholds the predicted probability at zero point five with a hard-coded comparison, and every operational figure, the confusion counts, the false-positive rate of zero point zero zero zero one, the three per-family recalls, is read at that point. No calibration step appears in any cell. No cost matrix appears in any cell. Across all thirty-six notebooks, probability calibration, threshold selection from a precision-recall curve and explicit cost matrices each appear zero times out of thirty-six.

SCIENTIFIC CONSEQUENCE

At a base rate of one in two thousand, an unweighted, uncalibrated classifier's output is not a probability, and zero point five is not a meaningful place on it. The evidence is in the notebook's own numbers: fifty-seven of one hundred forty-two attacks fall on the wrong side of that arbitrary line while only nineteen benign flows do. Moving the threshold down would trade some of those fifty-seven misses for a handful of extra alerts, and the notebook never sweeps it, so it never discovers what its own model is capable of.

REQUIRED REPAIR

Calibrate with Platt scaling or isotonic regression, report a reliability diagram and a Brier score, then choose the operating point by expected cost with a stated cost matrix, and report the alerts per analyst per day it implies.

The crowned model makes more errors than the one it beat

EVIDENCE

From the cell-sixteen table, on two hundred sixty-two thousand one hundred forty-four held-out rows: XGBoost accuracy zero point nine nine nine seven one zero, which is seventy-six errors, and RandomForest accuracy zero point nine nine nine seven two nine, which is seventy-one errors. XGBoost is crowned on ROC-AUC zero point nine nine six three four nine against RandomForest's zero point nine six four four one four. McNemar or any paired test appears zero times across all thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

Every operational number in section nine is computed at a threshold of zero point five, and at that threshold the crowned model is five errors worse than the model it beat. The study selects on one axis and reports on another, without ever testing whether the selection is stable. With seventy-one and seventy-six errors and one hundred forty-two positives, a paired test is a few lines and would tell you whether the difference exists.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the two candidates, and a bootstrap interval on the ROC-AUC difference.

DEFENSIBLE REPLY

We ranked on ROC-AUC and said so, and at a base rate of five ten-thousandths accuracy is the wrong axis by our own section-one argument; the AUC gap between 0.996 and 0.964 is thirty-two thousandths, which is not a tie, so the selection is not arbitrary.

ORDERED CORRECTION

The severity is right. One available strengthening that keeps it inside CONTESTED: RandomForest's 0.964414 against XGBoost's 0.996349 is a large gap on a held-out set containing 142 positives, so the AUC difference itself has no interval either - which is fault 5's point, not fault 10's. Consider merging: a bootstrap on the AUC difference discharges both.

Precision is never computed, at this base rate or any other

EVIDENCE

Section one says web attacks are extremely rare here, five hundredths of one percent of flows, making this a severe class-imbalance problem where aggregate accuracy is meaningless. The word precision appears in no printed output. From the printed numbers, the false-positive rate is nineteen over two hundred sixty-two thousand two, which is seven point two five in one hundred thousand, and the recall derived above is zero point five nine nine.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic the notebook owed, and you can check it. At the corpus base rate its precision is eighty-five over one hundred four, zero point eight one seven. Now take a base rate of one attack per ten thousand flows, five times rarer. Per million flows: one hundred attacks, of which the model catches fifty-nine point nine, and nine hundred ninety-nine thousand nine hundred benign flows, of which it flags seventy-two point five. Precision falls to fifty-nine point nine over one hundred thirty-two point four, which is zero point four five two. More than half your alerts become false, and the notebook that opens by lecturing the reader on base rates never performs this substitution once.

REQUIRED REPAIR

Report precision at the corpus base rate and at a stated deployment base rate, and report alerts per analyst per day at a stated daily flow volume.

One hundred forty-two flows from a handful of sessions, split at random

EVIDENCE

Cell sixteen calls a stratified random split at test size zero point two five with random state zero and no grouping key. The held-out positive class is one hundred forty-two flows, derived from the printed two hundred sixty-two thousand one hundred forty-four and two hundred sixty-two thousand two. Group k-fold and group shuffle split appear zero times across all thirty-six notebooks. The section-thirteen scope paragraph concedes that the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

The Centre ran these web attacks from a small number of source machines against a small number of targets in a scheduled window. Under a random split the model trains on some flows from a session and is tested on other flows from the same session, produced by the same tool moments apart. With one hundred forty-two positives, the number of independent attack events behind that figure is very small and is never reported. Reporting one hundred forty-two as though it were one hundred forty-two independent observations is the central inferential error.

REQUIRED REPAIR

Group shuffle split on source-destination pair and session, report how many independent attack sessions exist, and treat that as the sample size.

DEFENSIBLE REPLY

The source-destination pair you want me to group on is not in this file. It is eighty columns beginning Dst Port, Protocol, Timestamp - CIC's per-day ML exports carry no address columns, and you can check the header in one line.

ORDERED CORRECTION

Re-key and the fault becomes FATAL again and stronger than as filed, because I can supply the number the fault says is never reported. Parsing the Timestamp column that IS in df gives exactly three attack windows on this day: Brute Force -Web 09:17:04-11:02:58 (362 flows), Brute Force -XSS 13:01:04-14:10:05 (151 flows), SQL Injection 15:05:22-15:17:24 (50 of the 53, with 3 strays). So the 142 held-out positives come from three tool runs, and the effective sample size of the positive class is three, not 142. REQUIRED becomes: leave-one-window-out on the disambiguated Timestamp, or leave-one-family-out on family, and report the number of independent attack events as the sample size. Note the meridiem must be disambiguated first - the column is a twelve-hour clock, which is why XSS reads 01:01 for a 13:00 attack.

13

CRITICAL

UNTESTED

VERIFIED

NO TEMPORAL HOLDOUT ON TIMESTAMPED DATA

The timestamp was in the file and the loader deleted it

EVIDENCE

Cell eleven sets the drop list to Label, Timestamp, y and family. Cell sixteen splits at random. A temporal or time-ordered split appears zero times out of thirty-six across this series. The section-thirteen scope paragraph names the temporal split as the untested question.

SCIENTIFIC CONSEQUENCE

The column's availability is proved by its deletion, so no one can plead that the ordering was unavailable. Web attacks on this capture occur in a scheduled window; a random shuffle puts the front of that window in training and the back of it in test. Sorting by time and testing forward is the difference between recognising a session and detecting a class, and it is a two-line change.

REQUIRED REPAIR

Sort by Timestamp, train on the earlier period and test on the later, and report the drop in the abstract.

14

CRITICAL

DOCUMENTED

VERIFIED

THE CORPUS IS NOT THE EXPERIMENT

Saturation asserted in a comment, never measured

EVIDENCE

The cell-sixteen comment reads, quote, these learners saturate long before then on this data, unquote. No learning curve appears anywhere. The abstract says, quote, we load at least one million real CICFlowMeter records, unquote, and then, the four learners fit a one-hundred-twenty-thousand-row stratified subsample.

SCIENTIFIC CONSEQUENCE

This one is worse here than in the sibling notebooks, because of the base rate. A stratified subsample of one hundred twenty thousand rows at an attack rate of five ten-thousandths leaves roughly sixty attack examples to learn from. The notebook throws away roughly eighty-five percent of the available positives on the strength of an untested comment about saturation, and it is precisely the rare-class recall, which the conclusion nominates as the honest metric, that would be most sensitive to that discarding.

REQUIRED REPAIR

A learning curve in the number of positive examples, not the number of rows. Show recall on the rare families flattening before you cap the training set.

One draw, and one hundred forty-two events in it

EVIDENCE

Cell ten sets random state to zero with the comment, quote, single seed used everywhere, unquote. The split, the subsample, the ablation re-splits and the cross-validation folds all inherit it. The eight-seed sweep reported in the audit was run externally and on the single-feature diagnostic only, never on the reported model.

SCIENTIFIC CONSEQUENCE

With one hundred forty-two positives partitioned into three families, which particular flows land in the held-out quarter is a large source of variance, and the three per-family recalls are the numbers the conclusion calls honest. A different seed reshuffles those one hundred forty-two flows entirely. Reporting zero point five three six from one draw, with no denominator and no repetition, is reporting an anecdote at three decimal places.

REQUIRED REPAIR

Ten or more seeds through the full pipeline, reporting the mean and standard deviation of each per-family recall as well as of the headline.

Contamination is measured only where it is easiest to see

EVIDENCE

Cell twenty rounds to six decimals and uses tuple set membership, reporting contamination zero point zero eight eight and an exact-duplicate rate of zero point one three two. Cell twenty-two calls drop-duplicates, which is exact equality, removing thirteen percent of rows. No nearest-neighbour or tolerance-based measure appears anywhere.

SCIENTIFIC CONSEQUENCE

Automated web tooling emits requests that differ in a byte count or a millisecond. Exact matching treats each of those as an unseen row, so eight point eight percent is a floor on the contamination, not an estimate of it. With only one hundred forty-two positives in the held-out split, even a handful of near-duplicate attack flows straddling the boundary would move the reported recalls materially, and the notebook has no measurement of how many there are.

REQUIRED REPAIR

Nearest-neighbour distance distribution across the split boundary, the fraction of held-out rows within a small epsilon of a training row, and a re-run de-duplicated at that tolerance.

DEFENSIBLE REPLY

Our contamination is the lowest of the four at 0.088, grade B, and our single-feature check is a clean A at 0.7457, so unlike our siblings there is no evidence here of a memorisation shortcut for near-duplicates to be hiding behind.

ORDERED CORRECTION

The severity is right and the rebuttal is honest. The one counter that survives is specific and should be kept: with only 142 held-out positives, a handful of near-duplicate ATTACK flows straddling the boundary would move the three recalls materially, and the exact-match test cannot see them. Confine the fault to the positive class, where the sample is small enough for the argument to have force, and drop the corpus-wide framing.

The ablation raises the score, and the reason is never measured

EVIDENCE

The cell-twenty-two table reads: headline zero point nine nine six three four nine; de-duplicated with thirteen percent of rows removed, zero point nine nine nine zero four nine. The conclusion says, in the notebook's own words, that collapsing duplicates removes the hardest rows, that identical feature vectors carrying conflicting labels get resolved to one label, and that the de-duplicated task is therefore easier, not cleaner.

SCIENTIFIC CONSEQUENCE

That explanation is correct, and it names a quantity: the rate at which identical vectors carry different labels. That rate is the irreducible error of the task, a hard ceiling on any achievable score, and it is one line of pandas to compute. The notebook explains precisely why it matters and does not measure it. So the headline is reported without the one number that would tell you how much of the remaining error is unremovable. This is the master fault again: correct diagnosis, no measurement.

REQUIRED REPAIR

Compute and report the label-disagreement rate among duplicated feature vectors, and state the resulting noise floor beside the headline.

DEFENSIBLE REPLY

Measure the rate you are demanding: there are two conflicting groups in the whole corpus, six rows, two of which change label. A noise floor of two rows in a million cannot be what lifted my AUC by three parts in a thousand.

ORDERED CORRECTION

Re-mechanise, and the fault gets better, because the notebook's own explanation is what turns out to be false. The conclusion states that de-duplication raises the score because 'identical feature vectors carrying conflicting labels get resolved to one label' and the task is therefore easier. That is wrong by measurement: 2 conflicting groups, 6 rows. The actual mechanism is arithmetic the notebook never did - de-duplication removes 13.2% of rows but keeps 95.6% of the positives (566 down to 541), which lifts the positive rate from 0.000540 to 0.000594, so the re-split's 120,000-row stratified subsample receives about ten per cent more attack examples on a task that has only about sixty. The ablation is not a cleaner task, it is a task with more positives. Charge: the validity ledger explains its own most surprising result with a mechanism it never measured and which is false, and that explanation is doing load-bearing work in the conclusion.

A web-attack detector with no attacker in the loop

EVIDENCE

No cell perturbs any input. The features the model leans on are backward packets per second, backward packet length maximum and packet length maximum, per the single-feature audit and the auditor's importance ordering. Nothing in the notebook measures recall under any change of attacker behaviour.

SCIENTIFIC CONSEQUENCE

Every one of those columns is set by how fast the attacker's script sends and how large its requests are. An attacker who paces the tool and pads the requests moves inside the benign envelope at no cost to the injection itself, because the payload is invisible to the feature set anyway. The detector already misses two attacks in five against an adversary who is not even trying. Its worst case against one who is, is unmeasured.

REQUIRED REPAIR

Define a perturbation budget over rate and size columns, generate evaded flows inside it, and report worst-case recall under that budget.

DEFENSIBLE REPLY

Show me the cell. Backward packet length maximum and packet length maximum are not in my single-feature panel's top eight, and the notebook contains no permutation-importance computation for you to be quoting - and my scope paragraph makes no robustness claim for you to refute.

ORDERED CORRECTION

Strip the unverifiable citation. Rebuild on what is printed and checkable: the strongest single feature is Bwd Pkts/s at 0.7457, which is a rate the attacker's script sets directly, and the notebook's own ablation shows that dropping it costs only 0.996349 to 0.994340 - so the detection surface is a distributed timing-and-volume envelope rather than one column, and nothing in the notebook measures what happens when an attacker paces the tool. Concede that no robustness claim is made, and file this as a gap in a security thesis rather than an overclaim. The strongest line in the fault - 'the detector already misses two attacks in five against an adversary who is not even trying' - is fault 2's finding and should stay there.

The missing experiment is a file in the same folder

EVIDENCE

The section-thirteen scope paragraph states, in the notebook's own words, that the split is random, not temporal or entity-grouped, and that every number therefore measures in-distribution separability only. The same directory contains nb thirty-six, cross-distribution transfer, together with two sibling CIC notebooks sharing the identical sixty-eight-column schema.

SCIENTIFIC CONSEQUENCE

The brute-force day and the DoS day are in the same folder with the same feature schema, and the two thousand seventeen corpus, which does contain web attacks, is in the folder as well. Training here and testing there is an import statement. The notebook wrote the paragraph explaining why that experiment is the one that matters and did not run it.

REQUIRED REPAIR

Train here and test on the two thousand seventeen web-attack traffic, both directions, and report the transfer drop in the abstract.

DEFENSIBLE REPLY

The 2017 files use a different column vocabulary from the 2018 files - Destination Port against Dst Port, Total Fwd Packets against Tot Fwd Pkts - so your import statement is a seventy-eight entry rename map, and nb36's protocol is built on a shared NetFlow-v2 schema that neither of these corpora has.

ORDERED CORRECTION

This is the one E2 fault of the three with a genuinely matched attack class - 2017 does contain web attacks - so keep it, but state the cost honestly: the schemas are not identical, a column-name mapping is required, and the notebook's 68 columns must be aligned to 2017's. Drop the nb36 reference under CORRECTION 4, and drop the sibling-day suggestion, which compares disjoint attack classes. The free and fully constructible version worth adding: score this model on the sibling days' benign flows - 667,626 and 996,077 rows in the identical 68-column schema - and report the false-positive rate on unseen benign traffic.

The word real is doing work it cannot support

EVIDENCE

The abstract says, quote, we load at least one million real CICFlowMeter records, unquote. Section two describes the corpus, following Sharafaldin and colleagues, as profiled benign traffic plus a labelled attack schedule; the benign side is generated by the Centre's B-Profile system.

SCIENTIFIC CONSEQUENCE

The records are real outputs of a real extractor run over synthetic traffic. Discriminating one generator's flows from another's is a solvable problem, and the notebook's zero point nine nine six is a measurement of the distance between those two pieces of software on one afternoon. Placing the word real in the abstract, unqualified, invites the reader to read it as production evidence, which nothing in this notebook supports.

REQUIRED REPAIR

Qualify the word real, or evaluate on captured production traffic and report that number instead.

DEFENSIBLE REPLY

You concede in your own sentence that these are real outputs of a real extractor run over that traffic. The word is doing exactly the work it says: these are genuine CICFlowMeter records, not simulated feature vectors.

ORDERED CORRECTION

Reduce to the editorial residual and say so: the abstract's unqualified 'real' invites a production reading, and the qualification belongs there. Note that this notebook's conclusion is weaker than its siblings' on this axis - nb02 and nb03 both restrict the claim explicitly in section 13, and this one does not - which is the version of the charge that actually distinguishes it, and the filing does not make it.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 262,144 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. This notebook is the one case in the series where corpus size is a real constraint — at an attack rate of 0.0005 the held-out half carries roughly 131 positives, so a third partition would leave too few to estimate anything. Use nested cross-validation here and report the outer-loop estimate.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Atlas proof

docs/course/security-ml-notebooks/nb04-cic-ids2018-webattacks.ipynb · Notebook cell 10; zero-based manifest cell IDs: 10=fa37ee4a

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. I want to be clear about why this one is rejected, because it is the best of the four and its trust grade is a B. It is not rejected for saturation; its single-feature check is a genuine A and it is not a one-column result. It is rejected because of what the study is made of. The file stops at exactly two to the twentieth power lines, and the provenance table attributes that to a download bound that provably never engaged. Inside that file, the entire held-out attack class is one hundred forty-two flows, and every headline number, every per-family recall, every claim about rare-class detection, rests on those one hundred forty-two events with not one denominator or interval printed anywhere. And when you reconstruct what the detector actually does from three printed numbers, it catches eighty-five of the one hundred forty-two. The abstract says zero point nine nine six three four nine. The result is zero point five nine nine. Both were available; only one was reported.

Intrusion Detection on a 5G Network: the 5G-NIDD Dataset

5G-NIDD, one million two hundred fifteen thousand eight hundred ninety flows by forty-six features from a single 5G radio-access testbed, positive rate zero point six zero seven one, pulled from a Kaggle re-upload.

HEADLINE

LightGBM, held-out ROC-AUC zero point eight seven four five one four, three-fold cross-validated zero point eight seven four two plus or minus zero point zero zero three one.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 2 critical (FATAL)
- 20 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me be clear about what this notebook did right, because it is not nothing. It ran the audit, it found the problem, and it printed the problem: a duplicate rate of zero point six nine two, a train-test contamination of zero point six zero one, a self-assigned data-trust grade of F, and a false-positive rate with the raw counts attached. Most published intrusion-detection papers print none of those four. It also refuses the easy win — section thirteen says outright that de-duplication raising the AUC is not evidence the headline is safe. That is more integrity than the field's median paper shows.

EXAMINER RULING

This notebook reports a detector that mistakes forty-seven of every hundred benign flows for an attack, scored on a test set of which sixty percent was copied from its own training data, and whose one leak-free run returns a perfect one point zero — I reject it with prejudice, because the quantity reported is a property of the file and not of detection.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Half of all benign traffic is flagged

EVIDENCE

Cell eighteen prints, verbatim: operational false-positive rate at zero point five equals zero point four six six three, fifty-five thousand six hundred ninety-seven benign flows flagged out of one hundred nineteen thousand four hundred thirty-four. Do the subtraction with me. One hundred nineteen thousand four hundred thirty-four minus fifty-five thousand six hundred ninety-seven leaves sixty-three thousand seven hundred thirty-seven benign flows correctly cleared. That is a specificity of zero point five three three seven. On benign traffic this detector is a coin flip with a slight thumb on the scale. Now the base rate. Assume, purely for the arithmetic, a network carrying one million flows a day at one percent attack prevalence. That is ten thousand attacks and nine hundred ninety thousand benign flows. Nine hundred ninety thousand times zero point four six six three is four hundred sixty-one thousand six hundred thirty-seven false alarms. Even if this model caught every single attack — and section thirteen says it catches eighty-six point six percent of UDPFlood — you get four hundred seventy-one thousand six hundred thirty-seven alerts of which ten thousand are real. One alert in forty-seven. And the friendly praxis pack scored this notebook ten out of ten.

SCIENTIFIC CONSEQUENCE

The central claim of an intrusion-detection notebook is that it detects intrusions. A system that raises an alarm on forty-seven percent of normal traffic has no operating point at which a human can use it, so there is no reading of zero point eight seven four five one four that constitutes evidence of detection. The pack's ten out of ten is not wrong on its own terms — it graded the honesty of the audit — but honesty about an unusable number does not make the number usable.

REQUIRED REPAIR

An explicit cost matrix, a stated deployment base rate, and a threshold chosen to minimise expected cost rather than left at zero point five. Then precision at a fixed recall, and alerts per analyst per day at the chosen threshold. None of the three appears anywhere in the notebook; a grep of all twenty-seven cells returns zero hits for a cost matrix and zero for threshold selection.

Sixty percent of the test set is training data

EVIDENCE

Cell twenty prints an exact-duplicate row rate for the whole corpus of zero point six nine two, and a train-slash-test exact-row contamination of zero point six zero one. The notebook grades that F and prints, quote, data trust grade F. Cell sixteen shows the split: train underscore test underscore split with test size zero point two five, stratify on y, and no group argument. Across all thirty-six notebooks in this series, GroupKFold and GroupShuffleSplit appear zero times. Section thirteen states the position plainly: the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

Zero point eight seven four five one four is not a held-out estimate. Sixty point one percent of the rows it was scored on were byte-identical to rows in the training set. The quantity being reported is a memorisation rate with a small generalisation component attached, and no procedure in the notebook separates the two. Naming the defect in section thirteen does not remove it from the number in the abstract.

REQUIRED REPAIR

GroupShuffleSplit or GroupKFold keyed on whatever entity the capture provides — source host, session identifier, or capture segment — with the drop in held-out AUC reported as the headline result rather than as a footnote. Twenty lines of code.

The only leak-free run scores a perfect one

EVIDENCE

Cell twenty-two prints the ablation table: headline as-is, zero point eight seven four five one four; de-duplicated with sixty-nine percent of rows removed, one point zero zero zero zero zero zero; shortcut feature dropped, zero point eight seven four four one six. The de-duplication code is X dot drop underscore duplicates, then a fresh stratified re-split. Two consequences follow from that one line and neither is optional. First, after de-duplication no two rows are identical, so exact-row contamination on that split is zero by construction. Second, an AUC printed as one point zero zero zero zero zero zero means the distinct feature vectors of the two classes are perfectly separable to the six decimal places this notebook prints — the code rounds to six, so read it as at worst indistinguishable from perfect. Section thirteen's response is: the de-duplicated task is therefore easier, not cleaner.

SCIENTIFIC CONSEQUENCE

This inverts the finding. The single run in this notebook with no train-test leakage is the run that scores perfectly, which tells you the two classes occupy disjoint regions of feature space once you look at distinct vectors. That is the signature of a traffic generator, not of a detection problem. It also tells you what zero point eight seven four actually is: the same feature vector appears in the corpus carrying both labels, and the headline is measuring how often that happens. The notebook writes the sentence — identical feature vectors carrying conflicting labels get resolved to one label — and then does not measure the rate.

REQUIRED REPAIR

Report the label-disagreement rate directly: for every feature vector appearing more than once, the fraction carrying more than one label. That rate is an irreducible Bayes-error floor on this corpus and it is the honest interpretation of zero point eight seven four. Then report the class-conditional overlap on the de-duplicated corpus, which is what a one point zero zero zero zero zero zero is telling you about the generator.

DEFENSIBLE REPLY

That row is not my leak-free run, it is a different task, and here is the number neither of us printed. Half my duplicate groups - thirty-three thousand seven hundred and eight of sixty-seven thousand - carry both labels. Six hundred and thirty-two thousand rows, fifty-two percent of the corpus, sit in a group that disagrees with itself, and drop_duplicates deletes two hundred and eighty-one thousand of them, twenty-three percent of the corpus, by keeping whichever label came first. That twenty-three percent is the irreducible error floor. Remove it and a 1.000000 is arithmetic, not evidence - which is why my raw AUC is 0.8745. I wrote that reading in section eleven before I ran it and again in section thirteen after, and I put no headline on it.

ORDERED CORRECTION

Retitle from 'The only leak-free run scores a perfect one' to 'The label-disagreement rate is the number that would settle this, and it is never computed', and supply it, because it is decisive and it is 23 percent: 33,708 of 66,989 duplicate groups disagree (50.32 percent), 632,382 rows are in a disagreeing group (52.01 percent of the corpus), and 281,529 minority-label rows - 23.15 percent - are removed by construction. Note that this is corpus-specific

and must be measured, not assumed: the same charge is empirically FALSE on nb02 and nb03, where the disagreement rate is zero. Keep the REQUIRED's first clause verbatim; merge fault 14's REQUIRED into the second.

4

MAJOR

DOCUMENTED

DOWNGRADED

DISCLOSURE SUBSTITUTED FOR MITIGATION

ORAL TOP 5

The conclusion retracts the abstract

EVIDENCE

Section thirteen, verbatim: on this A-best slash F-worst scale, a D or F means the held-out number is not a generalisation estimate. And: treat the headline as a benchmark number, not a deployment estimate. And: every number above therefore measures in-distribution separability only. Meanwhile the title is Intrusion Detection on a 5G Network, and the abstract says we audit whether high scores reflect 5G attack detection or the usual flow shortcuts. The audit answered that question — the answer was F — and the headline did not change.

SCIENTIFIC CONSEQUENCE

At doctoral level a named fatal limitation is a confession, not a defence. Every experiment that would have settled the question is affordable. A grouped split is twenty lines. A label-disagreement count is three. Cross-distribution testing exists as notebook thirty-six in this same series, in this same codebase, written by this same author. The disclosure paragraph costs nothing and buys nothing, and this notebook is the clearest case in the set: it graded its own data F and published the number anyway.

REQUIRED REPAIR

Either run the experiment that settles the claim and report its result as the headline, or withdraw the claim and retitling the notebook as a study of the 5G-NIDD corpus rather than of 5G intrusion detection.

DEFENSIBLE REPLY

My abstract asks a question, it does not make a claim, and the question was whether the score is detection or a flow shortcut. The audit answered it: the shortcut sub-check scores A, and dropping the strongest single feature costs ninety-eight millionths. The F comes from the overlap check, which is a different failure mode, and section thirteen says in terms that the overlap check drives the grade and the single-feature check separately scores A. You cannot call that a retraction of a claim I did not make.

ORDERED CORRECTION

Delete the closing sentence 'you published zero point eight seven four five one four in the abstract' - the abstract contains no AUC. Restate the charge against the title only: the subtitle already reads 'Model comparison + validity audit on 5G-NIDD', so the REQUIRED's retitling half is largely already met, and the durable residual is that the ranking table's 0.874514 is presented without the scope qualifier attached to it at the point of presentation.

A million-row gate guarding a hundred-thousand-row experiment

EVIDENCE

Cell eleven contains the line: assert len of df is greater than or equal to one million, with the comment honesty gate. It loads one million two hundred fifteen thousand eight hundred ninety rows. Cell sixteen holds out three hundred three thousand nine hundred seventy-three and then subsamples the training pool to one hundred and twenty thousand. Subtract: one million two hundred fifteen thousand eight hundred ninety minus three hundred three thousand nine hundred seventy-three leaves nine hundred eleven thousand nine hundred seventeen in the training pool, and one hundred and twenty thousand of those are used. Seven hundred ninety-one thousand nine hundred seventeen rows were downloaded, asserted past an honesty gate, and shown to no model. That is nine point nine percent of the corpus reaching the learner. The code comment justifies it: because these learners saturate long before then on this data.

SCIENTIFIC CONSEQUENCE

The saturation claim is the load-bearing justification for discarding eighty-seven percent of the available training data, and it is asserted with no evidence. A grep of all twenty-seven cells for learning underscore curve returns zero. There is no plot, no table, no printed number anywhere in this notebook showing AUC as a function of training-set size. The subtitle sells greater-than-or-equal-to one million records; the experiment is a hundred and twenty thousand.

REQUIRED REPAIR

A learning curve to saturation — AUC against training-set size across at least five points on a log scale — or the effective n stated in the abstract in place of the corpus size. One of the two, not neither.

DEFENSIBLE REPLY

The assert gates the corpus, not the experiment - it is a course floor on which files you may use. And the saturation claim is not bare: my own cross-validation fits on about twenty-six thousand seven hundred rows and returns 0.8742 against 0.874514 at a hundred and twenty thousand. That is a four-and-a-half-fold change in n for three ten-thousandths of AUC. The test set is three hundred and four thousand rows.

ORDERED CORRECTION

Retitle to 'Saturation is asserted in a comment and never plotted'. Drop 'a hundred-thousand-row experiment' from the title - the held-out set is 303,973. Keep the learning-curve REQUIRED; the two in-notebook points at $n = 26,667$ and $n = 120,000$ are a two-point curve and the ask is to extend it, which is fair and cheap.

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

7

MAJOR

UNTESTED

REMECHANISED

ROC-AUC AS HEADLINE; PR CURVE DRAWN AND NOT READ

You plotted the diagnostic and declined to read it

EVIDENCE

Cell eighteen imports precision underscore recall underscore curve, computes prec and rec, and plots them on axis two with the title Precision-Recall. That is the correct diagnostic. A grep of all twenty-seven cells for average underscore precision returns zero. Across all thirty-six notebooks in the series, average underscore precision underscore score appears zero times while a precision-recall curve is plotted in all thirty-six. The ranking in cell sixteen is by ROC-AUC alone.

SCIENTIFIC CONSEQUENCE

Precision is the quantity an analyst experiences and it is the quantity this notebook's own false-positive rate makes catastrophic. The area under the drawn curve is one function call away, and the notebook drew the curve, so it cannot claim not to have known which diagnostic mattered.

REQUIRED REPAIR

average underscore precision underscore score with a bootstrap confidence interval, reported beside ROC-AUC, and precision quoted at a fixed recall such as zero point nine.

DEFENSIBLE REPLY

Attacks are sixty-one percent of this corpus. The precision-recall curve you want me to integrate is already drawn with the majority class as positive, so its no-skill line sits at 0.6071 and its area tells you almost nothing. The class you should be asking about is benign - the rare one, the one I miss half of - and neither of us has a curve for it.

ORDERED CORRECTION

Replace the REQUIRED with: 'A precision-recall panel computed with BENIGN as the positive class, with average_precision_score and a bootstrap interval, and benign precision quoted at a fixed benign recall - the rare-class operating point, which is where the 0.4663 lives and which no panel in this notebook establishes.' Do not ask for average_precision on the attack class here; a student who checks the positive rate will invert the argument in the room.

Atlas proof

docs/course/security-ml-notebooks/nb05-5g-nidd.ipynb · Cell index 18, precision-recall plot; zero-based manifest cell IDs: 18=cfa1e297

A winner declared on ninety-four millionths

EVIDENCE

Cell sixteen prints the ranking table. LightGBM: accuracy zero point seven six six five eight five, ROC-AUC zero point eight seven four five one four. XGBoost: accuracy zero point seven six six nine nine two, ROC-AUC zero point eight seven four four two zero. Subtract the AUCs: ninety-four millionths. Now read cell twenty-four: the three-fold cross-validated AUC is zero point eight seven four two plus or minus zero point zero zero three one. The fold-to-fold spread is thirty-three times the margin that decided the winner. And on accuracy, the declared loser is ahead by four hundred and seven millionths. A grep for McNemar across all twenty-seven cells returns zero.

SCIENTIFIC CONSEQUENCE

The word best in best model colon LightGBM is not supported by anything. The two learners are statistically indistinguishable on this split, and the ranking would plausibly invert on a different seed. Every downstream result — the per-family recalls, the false-positive rate, the ablation — is computed on a model chosen by noise.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, or overlapping bootstrap confidence intervals on the two AUCs, with the conclusion stating that the learners are indistinguishable if they are.

DEFENSIBLE REPLY

I did not declare a winner, I sorted a table and took the top row, and I printed six decimals with a comment saying a 1.000000 is a red flag, not a win. Your thirty-three-times figure compares fold-to-fold spread on a forty-thousand-row subsample against a paired difference on the same three-hundred-thousand-row test set. Those two models' errors are almost perfectly correlated, which is why the right test is the paired one you asked for - and which is why the fold spread is not the yardstick.

ORDERED CORRECTION

Delete the '33 times the margin' sentence; it compares an unpaired fold spread to a paired difference. The charge that survives is narrower and still correct: section 13 writes 'Winning learner: LightGBM' with no qualification, and a McNemar test on the paired held-out predictions would almost certainly return indistinguishable. Ask for that sentence to be changed.

Six per-family recalls, no denominators

EVIDENCE

Cell eighteen prints: worst per-family recalls, UDPFlood zero point eight six six, HTTPFlood one point zero, SlowrateDoS one point zero, TCPConnectScan one point zero, SYNScan one point zero, UDPScan one point zero. Not one count appears. A grep of all twenty-seven cells for bootstrap returns zero and for confidence interval returns zero. Across the whole thirty-six-notebook series, bootstrap and confidence-interval methods appear zero times.

SCIENTIFIC CONSEQUENCE

A recall of one point zero on a family with forty positives and a recall of one point zero on a family with four hundred thousand positives are different findings, and this notebook prints them in the same font with the same number of decimals. Five families at exactly one point zero is not a performance result; it is a statement that the generator produced those families with a distinctive signature. Without denominators the reader cannot tell which.

REQUIRED REPAIR

Print every family with its positive count, and attach a binomial confidence interval — Wilson or Clopper-Pearson — to every recall. For the aggregate, a bootstrap interval on the held-out AUC.

DEFENSIBLE REPLY

Ask what the denominators actually are before you say the recalls might be noise. The smallest family I print is UDPScan at sixteen thousand rows, about four thousand in the held-out quarter; a Wilson interval on a recall of one at that n is $[0.999, 1]$. The code computes the count, gates the display at five, and the counts are all in the thousands. What you have found is that I did not print them, which is a one-line fix, not a number I cannot support.

ORDERED CORRECTION

Keep the REQUIRED verbatim - printing counts and Wilson intervals is right - but delete the implication that the figures may be underpowered. State the counts yourself: UDPFlood 457,340, HTTPFlood 140,812, SlowrateDoS 73,124, TCPConnectScan 20,052, SYNScan 20,043, UDPScan 15,906 corpus-wide. The charge is transparency, not support.

Atlas proof

docs/course/security-ml-notebooks/nb05-5g-nidd.ipynb · Cell index 18; notebook-review-key nb05 criterion 10 and B4; zero-based manifest cell IDs: 18=cfa1e297

The cross-validation validates a different model

EVIDENCE

Cell sixteen fixes every hyperparameter by hand and never varies one: RandomForest with sixty estimators, XGBoost with eighty estimators and max depth six, LightGBM with eighty estimators, LogisticRegression with max iterations three hundred. A grep of all twenty-seven cells for any hyperparameter search returns zero. Cell twenty-four then reads: `c v X comma c v y equals X t r dot i l o c` up to forty thousand. Three-fold cross-validation over forty thousand rows means each fold fits on roughly twenty-six thousand six hundred sixty-seven rows. The headline model was fitted on one hundred and twenty thousand.

SCIENTIFIC CONSEQUENCE

The plus or minus zero point zero zero three one is the fold spread of a model trained on about a fifth of the reported one, computed on the same data the winner was selected from. It bounds the variability of a different estimator than the one in the abstract, and it is not a validation of the headline at all. With no search, no claim can be made that any of these four learners was given a fair configuration.

REQUIRED REPAIR

Nested cross-validation: an inner loop that selects hyperparameters and an outer loop that estimates performance, run on the same training set that produces the headline.

DEFENSIBLE REPLY

Nested cross-validation exists to remove the optimism of selecting hyperparameters on the data you then score. I select no hyperparameters - every one of them is a constant in cell sixteen - so the inner loop would have nothing to iterate over. If you want the cross-validation to validate the pipeline I reported, tell me to run it at `n equals one hundred and twenty thousand`. That is the fix.

ORDERED CORRECTION

Replace the REQUIRED with: 'Run the cross-validation on the same 120,000-row training configuration that produces the headline, not on a 40,000-row prefix, and state that the ± 0.0031 in the ledger is a spread at `n = 26,667`.' Then add the selection leak the current filing misses and which is the stronger version of this charge: `best_name` is chosen by ROC-AUC on `Xte`, and every reported diagnostic - confusion matrix, FPR, per-family recall, ablation baseline - is then computed on that same `Xte`. There is no separate validation split anywhere in the notebook.

One draw, reported to six decimals

EVIDENCE

Cell twenty-four prints seed equals zero. Every split in the notebook uses random underscore state equals RANDOM underscore STATE. The single-feature audit in cell twenty draws its own sample at random underscore state equals one. The eight-seed stability check that exists for this notebook was run by the external praxis probe, not by any cell in the notebook, and it probed the single-feature audit — the zero point six six five seven — not the zero point eight seven four five one four.

SCIENTIFIC CONSEQUENCE

A number reported to six decimal places from a single draw invites the reader to treat the sixth decimal as real. The notebook's own three-fold spread of plus or minus zero point zero zero three one says the third decimal is already uncertain.

REQUIRED REPAIR

At least ten seeds for the full pipeline, with mean and standard deviation reported in place of the point estimate.

DEFENSIBLE REPLY

The three-fold cross-validation in cell twenty-four does report a spread of plus or minus zero point zero zero three one, which bounds split-to-split variability well enough to show the headline is not seed-fragile at the second decimal.

A security detector never meets an adversary

EVIDENCE

A grep of all twenty-seven cells returns zero hits for adversarial, zero for evasion, and zero for any perturbation budget. Across all thirty-six notebooks in the series the count is the same. The literature review cites Sommer and Paxson two thousand ten, a paper about why closed-world machine-learning scores rarely survive deployment, with false-positive cost among its central reasons — and no experiment below it responds to that citation.

SCIENTIFIC CONSEQUENCE

An intrusion detector's worst case is its only operationally meaningful case, because the attacker chooses the input. The features that carry this score are flow statistics an attacker controls at essentially zero cost: packet timing, packet counts, sequence-number behaviour. Untested, the reported number describes an adversary who does not know a detector exists.

REQUIRED REPAIR

An evasion experiment under a stated perturbation budget — for instance, bounded padding and inter-packet delay applied to the attack class — with worst-case per-family recall reported alongside the clean number.

DEFENSIBLE REPLY

Name the threat model. These are forty-six Argus aggregates over a completed flow. If you perturb sTtl and TotBytes and Rate independently you have produced a row that no flow could generate, and whatever recall drops out of that is a property of my feature space, not of an adversary. The honest version of your experiment needs a traffic generator in the loop, which is a different notebook - and I do name the spoofability concern in the header.

ORDERED CORRECTION

Say that the perturbation budget must be defined over realisable flow transformations (padding, inter-arrival delay, session splitting) with the aggregates recomputed, not over the 46 feature columns directly. And note the charge is series-level: it is filed identically against all thirty-six notebooks, so it argues for one adversarial-robustness notebook in the series, not a FATAL against each.

The top two features are host artifacts

EVIDENCE

Cell twenty prints best single-feature AUC equals zero point six six five seven, feature SrcTCPBase. The praxis probe's depth check drops the model's top feature, SrcTCPBase, then its second, s T t l, and the AUC moves from zero point eight seven three to zero point eight seven two eight to zero point eight seven two nine. So the model's ranked top two are SrcTCPBase and s T t l. s T t l is the source time-to-live: a per-host constant set by the sending operating system and decremented by hop count. In a fixed testbed it identifies the machine, not the behaviour. No cell in this notebook, and no cell in any of the thirty-six, ablates generator-coupled or host-identity columns as a class.

SCIENTIFIC CONSEQUENCE

If the corpus's separability rests on which machine sent the packet rather than what the packet did, then the reported number measures testbed configuration. The notebook has the diagnostic — it already runs single-column AUCs for all forty-six features — and it never groups them into behaviour-derived and apparatus-derived and re-fits on the former.

REQUIRED REPAIR

Partition the feature set into behaviour-derived and apparatus-derived columns, drop the apparatus columns as a block, refit, and report the AUC on behaviour-only features as the honest headline.

DEFENSIBLE REPLY

SrcTCPBase is not a host identifier, it is a TCP-only field that is absent from seventy-seven percent of my rows, and my own fillna turned that absence into the number zero. Present, it is 94.6 percent attack; absent, 50.6 percent. The two hundred sixty-three thousand seven hundred forty-eight non-null attack rows are exactly the five TCP attack families. My strongest single feature is a protocol-presence indicator my preprocessing built - which is a worse finding than yours, and I will take it, because it is the true one.

ORDERED CORRECTION

Re-mechanise the SrcTCPBase half, which makes the charge stronger, not weaker: 'The strongest single feature is an imputation artifact of the notebook's own making. SrcTCPBase is null in 937,219 of 1,215,890 rows; X.fillna(0.0) in cell 11 turns that missingness into a value. $P(\text{attack} \mid \text{present}) = 0.9465$ against $P(\text{attack} \mid \text{absent}) = 0.5062$, and the 263,748 present-and-attack rows are exactly the five TCP attack families (SYNScan, TCPConnectScan, HTTPFlood, SlowrateDoS, SYNflood: 263,752). Measured directly on the notebook's own 60,000-row random_state=1 sample, a bare present/absent indicator scores a direction-agnostic AUC of 0.6658 against the printed 0.6657 for the column itself - the audit's headline feature carries no information beyond its own missingness, and dropping it costs 98 millionths because Proto is still in the feature set and encodes the same thing.' Keep the sTtl half as filed. Mark the depth-check figures (0.873 / 0.8728 / 0.8729) as external-probe output, not notebook output.

Redundant separability is diagnosed and never measured

EVIDENCE

Cell twenty-two: dropping SrcTCPBase moves the AUC from zero point eight seven four five one four to zero point eight seven four four one six, a loss of ninety-eight millionths. Section thirteen writes the correct reading: so the separability is multi-feature, that reflects how this corpus was generated, not one leaky column. Section eleven's preamble says the same thing in advance — if it barely moves, common on simulated corpora, that is not vindication. And then no cell measures overlap, distance between class distributions, or a nearest-neighbour margin. A grep confirms it.

SCIENTIFIC CONSEQUENCE

The notebook diagnoses its own corpus and then declines to run the one measurement that would turn the diagnosis into a finding. Writing that many columns independently separate the classes is a claim about the joint distribution, and it is testable in a dozen lines. Stating it in prose and printing an AUC instead is disclosure standing in for measurement.

REQUIRED REPAIR

Measure the class-conditional overlap directly: nearest-neighbour distance from each attack row to the nearest benign row and vice versa, or a two-sample divergence estimate, reported as a distribution rather than a sentence.

DEFENSIBLE REPLY

Section eleven's 'barely overlap' sentence is a template that describes the collapse case, and it does not describe this run. My AUC is 0.8745 and I flag half of all benign traffic. If the distributions barely overlapped I could not miss like that. Section thirteen's actual sentence is that the separability is multi-feature - which is what dropping the top feature for ninety-eight millionths demonstrates.

ORDERED CORRECTION

Merge into fault 3, whose REQUIRED already asks for the class-conditional overlap measurement. If kept separate, restrict the charge to the one unmeasured causal sentence - 'That reflects how this corpus was generated, not one leaky column' - and drop the section-11 'barely overlap' framing, which does not apply at an AUC of 0.8745 and an FPR of 0.4663.

15

MAJOR

UNTESTED

VERIFIED

IMPURITY IMPORTANCE REPORTED AS EXPLANATION

The importance chart cannot support a claim

EVIDENCE

Cell eighteen reads: if hasattr best comma feature underscore importances underscore, then imp equals best dot feature underscore importances underscore, plotted as the top twelve. That is LightGBM's default split-count importance. A grep of all twenty-seven cells for permutation underscore importance returns zero, and for SHAP returns zero — the four apparent hits are the string dot shape.

SCIENTIFIC CONSEQUENCE

Split-count importance systematically rewards columns with many distinct values, which on a flow corpus means timing and sequence fields regardless of their predictive contribution. A chart built on it cannot be read as an explanation of what the model learned.

REQUIRED REPAIR

Permutation importance computed on the held-out split, or SHAP values, with the impurity chart either replaced or labelled as a training-set diagnostic only.

DEFENSIBLE REPLY

No claim in section thirteen rests on the importance chart — the ablation drops the single-feature audit's winner, not the chart's — so the biased estimator informs a figure and nothing else.

16

MAJOR

UNTESTED

VERIFIED

NO PROBABILITY CALIBRATION

An uncalibrated probability read at a threshold

EVIDENCE

Cell eighteen: p b equals best dot predict underscore proba of X t e, column one; pred equals p b greater than zero point five. A grep of all twenty-seven cells for any calibration method returns zero, and across all thirty-six notebooks the count is zero. There is no reliability diagram and no Brier score anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The number zero point five means something only if the score it is compared against is a probability. LightGBM's raw output under log-loss is not calibrated in general, so the false-positive rate of zero point four six six three is the consequence of an uncalibrated score meeting an arbitrary constant — two unjustified choices compounding.

REQUIRED REPAIR

Platt scaling or isotonic regression fitted on a held-out calibration split, with a reliability diagram and a Brier score reported before any threshold is applied.

DEFENSIBLE REPLY

The headline is ROC-AUC, which is invariant to any monotone recalibration, so calibration cannot change the reported number — only the operating-point numbers, which the notebook already labels as being at threshold zero point five.

The encoder saw the test set, and nobody counted

EVIDENCE

Cell eleven: for c in X dot select underscore dtypes include object dot columns, X of c equals LabelEncoder dot fit underscore transform. That fit runs on all one million two hundred fifteen thousand eight hundred ninety rows, before cell sixteen splits. Section thirteen's defence reads: on an all-numeric corpus that step is a no-op. No cell prints the number of object-dtype columns, so the reader cannot check whether the corpus was all-numeric. Across the series, thirty-one of thirty-six notebooks fit a LabelEncoder before the split.

SCIENTIFIC CONSEQUENCE

A defence that the reader cannot verify from any printed output is not a defence. And the sibling notebook on Edge-IIoTset, one notebook later in the same series, contains a guard block whose own comment explains that some captures serialise the same numeric value differently per class and that this turns an innocuous protocol field into a perfect label proxy purely through text formatting. That hazard was discovered and the guard was written; it is not in this notebook.

REQUIRED REPAIR

Fit the encoder inside the training fold only, with an explicit unseen-category bucket for test values, and print the count and names of the columns encoded so the no-op claim is checkable.

DEFENSIBLE REPLY

Five columns are encoded: Proto with eight values, sDSb with twelve, dDSb with six, Cause with three, State with eleven. Over one point two million rows every one of those values lands in both halves of a shuffled split with probability one to more decimal places than either of us cares about, so fitting the encoder inside the training fold returns exactly the mapping I have. The leak you are charging is zero, arithmetically.

ORDERED CORRECTION

This is the ONE notebook of the four where A4 has factual content, so re-mechanise rather than withdraw: 'Section 13's defence is a dangling conditional. This corpus is not all-numeric - Proto, sDSb, dDSb, Cause and State are object-dtype and are encoded across all 1,215,890 rows before the split - and no cell prints that, so the reader cannot tell the sentence does not apply. The remedy is the print, not the refit: the five cardinalities are 8, 12, 6, 3 and 11, so a train-fold-only fit would return the identical mapping and the transductive leak is nil.' Note for the pack: this same charge is filed FATAL against nb06 and nb07, where the encoder loop provably iterates an empty list. Fixing the severity ordering matters more than this filing does.

Atlas proof

```
docs/course/security-ml-notebooks/nb05-5g-nidd.ipynb · Cell index 11,
`LabelEncoder().fit_transform`; cell index 25, disclosed limitation; zero-based
manifest cell IDs: 11=bb927ac5, 25=9747478f
```

Credited to one party, downloaded from another

EVIDENCE

The section-three provenance table gives Source as Kaggle humera one one slash five g dash nidd dash dataset, and the abstract credits Samarakoon and colleagues, two thousand twenty-two. Those are different parties. Cell eleven downloads whatever that account currently hosts and asserts only that the row count is at least one million. No hash, no manifest, no comparison against the authors' own distribution appears anywhere.

SCIENTIFIC CONSEQUENCE

The reported quantity is a property of a file, and the notebook cannot establish which file. A re-upload may be filtered, de-duplicated, re-ordered or partial, and the sixty-nine point two percent duplicate rate is exactly the kind of property a re-upload can change.

REQUIRED REPAIR

Download from the authoritative distribution, or compute a checksum of the mirror and compare it against the authors' published artifact, and print the result.

DEFENSIBLE REPLY

The loader reads one million two hundred fifteen thousand eight hundred ninety rows by forty-six features, which matches the published corpus size, and the external probe re-read the same shape and reproduced the single-feature AUC to four decimals — so the mirror is at minimum the right corpus.

No number can be placed beside a prior result

EVIDENCE

The section-two related-approaches table lists exactly one prior result — Samarakoon and colleagues, two thousand twenty-two, baseline machine learning — with the caveat in-distribution single testbed. No cell reproduces their split, their feature set, or their metric. The table is explicitly labelled: these are not measurements reproduced on this exact corpus. So the notebook has one comparison point and declines to occupy it.

SCIENTIFIC CONSEQUENCE

A benchmark number that cannot be placed beside any prior number on the same protocol is not a scientific result; it is a local observation. Reporting zero point eight seven four five one four with no comparable baseline leaves the reader unable to say whether this is good, bad, or the same as everyone else's.

REQUIRED REPAIR

Run the dataset authors' published protocol and report your figure beside theirs in the same table, with the differences in preprocessing enumerated.

DEFENSIBLE REPLY

The table says on its face that these are not measurements reproduced on this corpus. I did not place a number beside a prior result because I said I was not going to.

ORDERED CORRECTION

Replace with the far stronger version available here. Samarakoon et al. report high-nineties accuracy on 5G-NIDD using the full column set, including Attack Tool; this notebook drops Attack Tool, Seq and Offset and reaches 0.874514. The comparison is not merely absent - it is the one experiment that would decompose the gap into leakage columns versus detection, and it is affordable in one cell.

An A awarded for the wrong absence

EVIDENCE

Cell twenty computes the grade from literal thresholds: F if the single-feature AUC is at least zero point nine nine nine, D at zero point nine nine, C at zero point nine five, B at zero point eight five, otherwise A. The single-feature AUC is zero point six six five seven, so it scores A. Section thirteen then says: that says the split leaks, not that features are clean; the single-feature check separately scores A. But cell twenty-two shows the de-duplicated corpus reaching one point zero zero zero zero zero zero, which is perfect separability with no single column responsible.

SCIENTIFIC CONSEQUENCE

The letter A is awarded for the absence of the one failure mode this corpus does not have — a single near-sufficient column — while the failure mode it manifestly does have, many redundant columns jointly achieving perfect separation, has no letter and no threshold. A grading scheme that cannot see its own corpus's dominant pathology is a label, not a measurement, and it is quoted in the conclusion as though it carried information.

REQUIRED REPAIR

Report the underlying quantities with confidence intervals and drop the letters, or add a sub-check whose cutoffs are sensitive to multi-feature redundancy — for example the AUC after dropping the top k features, for k up to ten.

DEFENSIBLE REPLY

You are comparing an A computed on 1.2 million rows against a 1.000000 computed on 377 thousand de-duplicated rows and calling the mismatch my error. I never ran the single-feature scan on the de-duplicated corpus, so 'perfect separability with no single column responsible' is your inference, not my result. And my composite grade is F - the sub-check reported exactly what it was built to report and said so.

ORDERED CORRECTION

Drop the 1.000000 evidence, which is not commensurable with the A. The durable charge is construct validity: 'The audit has two sub-checks and neither responds to multi-feature redundancy. Add the AUC after dropping the top k features, for k up to ten, as a third sub-check.' That is the REQUIRED's second clause and it stands on its own.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 303,973 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

De-duplication removed most of my corpus here, so the movement you see is dominated by a real effect, not by split noise. The criticism still applies — I never measured the nuisance term — but it is at its weakest on this notebook, not its strongest.

Closing. So the ruling is not that you failed to notice. You noticed everything. You graded your own data, you told the reader that the held-out number is not a generalisation estimate, and you published zero point eight seven four five one four in the abstract under a title that says intrusion detection on a 5G network. The grouped split was twenty lines. The label-disagreement count was three. Neither was run, and a detector that alarms on forty-seven percent of normal traffic went to press with a ten out of ten.

IoT/IoT Intrusion Detection: the Edge-IIoTset Dataset

Edge-IIoTset, two million two hundred nineteen thousand two hundred one rows by thirty-two features from a single I o T slash I I o T testbed with fixed per-role attacker and victim addresses, positive rate zero point two seven two zero, pulled from a Kaggle re-upload.

HEADLINE

LightGBM, held-out ROC-AUC zero point nine nine three one six seven, accuracy zero point nine six nine nine eight, three-fold cross-validated zero point nine nine two nine plus or minus zero point zero zero zero two.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 3 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 3 critical (FATAL)
- 19 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be generous here, because this is among the two or three cleanest notebooks in the whole set. The contamination is zero point zero three three, the trust grade is a real A, and the false-positive rate is zero point zero zero zero four — one hundred and eighty false alarms in four hundred three thousand nine hundred eleven benign flows, which is an operationally defensible number that most deployed systems would envy. It is also the only notebook of the four I hold that writes a named twenty-four-column exclusion list, dropping the source and destination hosts, the A R P I P v four fields and the H T T P D N S and M Q T T payloads, and the only one that adds a numeric-string coercion guard against text-formatting label proxies. That is real construct-validity work and I am not going to pretend otherwise.

EXAMINER RULING

This is careful work on a genuinely clean corpus and it still fails, because section one states that an I I o T detector must generalise and that an in-distribution score cannot show it, and then the notebook reports an in-distribution score and stops.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

1

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

ORAL TOP 5

Section one refutes section thirteen

EVIDENCE

Section one, cell two, verbatim: I I o T devices are constrained and long-lived, so a detector must generalize — exactly what an in-distribution score cannot show. Section thirteen, verbatim: the split is random, not temporal or entity-grouped, every number above therefore measures in-distribution separability only. Between those two sentences sits zero point nine nine three one six seven, reported as the result, with a three-fold cross-validated zero point nine nine two nine plus or minus zero point zero zero zero two.

SCIENTIFIC CONSEQUENCE

This is the master fault in its purest form anywhere in the thirty-six-notebook set. The author states the requirement, states that the chosen method cannot meet it, and then meets nothing and reports the number. At doctoral level naming a fatal limitation in the opening paragraph and again in the closing paragraph is not a defence — it is a confession delivered twice with the claim published in between. Notebook thirty-six in this same series performs cross-distribution transfer, so the capability sits in the same codebase and was not applied.

REQUIRED REPAIR

Either an entity-grouped or temporal holdout with the drop reported as the headline, or a train-on-Edge-I I o T set, test-elsewhere transfer run, or the claim withdrawn and the notebook retitled as a study of the corpus.

2

CRITICAL

UNTESTED

VERIFIED

FEATURES ENCODE THE GENERATOR, NOT THE ADVERSARY

ORAL TOP 5

A raw TCP sequence number separates the classes

EVIDENCE

Cell twenty prints: best single-feature AUC equals zero point seven two two nine, feature t c p underscore s e q. That is the raw TCP sequence-number field lifted straight from the capture. Section two's own related-approaches table already names the hazard: against Ferrag and colleagues' baselines it records the caveat in-distribution, protocol fields can leak. A grep of all twenty-seven cells finds no experiment that groups features by provenance and refits.

SCIENTIFIC CONSEQUENCE

A sequence number is a property of the sending stack and the capture, not of an intrusion. That it alone separates attack from benign at zero point seven two two nine is a construct-validity finding about the corpus, and the notebook prints it and moves on. The headline of zero point nine nine three one six seven is built on thirty-two columns of which this is one; nobody established that the rest are any different in kind, because the experiment that would establish it — drop the apparatus columns as a block, refit, report — was never run.

REQUIRED REPAIR

Partition the surviving thirty-two features into behaviour-derived and apparatus-derived, drop the apparatus block, refit, and report the AUC on behaviour-only features beside the headline.

The grouping key was identified and then deleted

EVIDENCE

Section three, verbatim: we also explicitly drop the endpoint-identity and payload columns — i p dot s r c underscore host, i p dot d s t underscore host, the A R P I P v four fields — and then, in a testbed capture the attacker and victim addresses are fixed per role, so keeping those columns lets the model read who is talking instead of what the traffic does. Cell eleven's DROP list contains all twenty-four of them. But read what DROP actually does: feat equals the columns of d f not in DROP, then X equals d f of feat dot copy. The exclusion removes those columns from the feature matrix and leaves them in the data frame. The source host is still sitting in d f at cell sixteen, where train underscore test underscore split is called with stratify on y and no group argument. And two cells later, cell eighteen performs exactly the retrieval a grouped split would need: f a m underscore t e equals d f dot l o c of X-test dot index, comma, family. Across all thirty-six notebooks in this series, GroupKFold and GroupShuffleSplit appear zero times.

SCIENTIFIC CONSEQUENCE

Availability is not an inference here; it is demonstrated. The host identity was in the file. The author knew it identified who was talking and wrote that sentence down. The column was still in scope at the moment of the split, and the exact pattern for pulling a d f column by held-out index appears two cells later in this same notebook. Every ingredient for a grouped split was present and assembled. The rows were split at random anyway, so the same host lands on both sides of the boundary. The model cannot use the address, but it can use everything correlated with the address, which in a fixed-role testbed is most of the remaining thirty-two columns. Zero point nine nine three one six seven therefore measures within-host recall, not detection.

REQUIRED REPAIR

GroupShuffleSplit or GroupKFold keyed on i p dot s r c underscore host — the column already extracted, already named, already dropped — with the drop in held-out AUC reported as the headline.

Atlas proof

docs/course/security-ml-notebooks/nb06-edge-iiotset.ipynb · cells 4, 11, and 25; zero-based manifest cell IDs: 4=0ee5602e, 11=5111be4b, 25=d200f186

Two of every three MITM flows are missed

EVIDENCE

Cell eighteen prints: worst per-family recalls, M I T M zero point three three nine. One minus zero point three three nine is zero point six six one — the model misses roughly two of every three man-in-the-middle flows. The same cell prints a false-positive rate of zero point zero zero zero four, one hundred and eighty benign flows flagged out of four hundred three thousand nine hundred eleven. The threshold that produced both is the literal constant in the line `pred equals p b greater than zero point five`. A grep of all twenty-seven cells returns zero for threshold selection and zero for a cost matrix.

SCIENTIFIC CONSEQUENCE

The notebook has an enormous unspent false-positive budget — one hundred and eighty alarms — and a headline attack class it detects a third of the time, and it never connects the two. Lowering the threshold would trade some of that budget for M I T M recall, and no cell explores it. Section thirteen even says that gap, not the aggregate score, is the operationally important result, and then reports zero point nine nine three one six seven as the result. This is why the operational-admissibility axis scores two out of five here even though the false-positive rate is the best of the four notebooks I hold: the axis grades whether an operating point, a cost model and an alert budget exist, not whether the number is flattering.

REQUIRED REPAIR

A stated cost matrix in which a missed man-in-the-middle carries an explicit cost in units of false alarms, a threshold chosen to minimise expected cost, and a curve showing M I T M recall against false-positive rate as the threshold moves.

DEFENSIBLE REPLY

Run your own base-rate arithmetic on my number. At one percent prevalence my 0.0004 gives about four hundred false alarms per million flows against nearly nine thousand true ones - precision above ninety-five percent. The threshold charge you are making is the same one you make against notebooks with a 0.4663 false-positive rate, and it does not land the same way here. What does land is that I never plotted MITM recall against alert volume.

ORDERED CORRECTION

Split the two halves. Drop the FPR half entirely (it exonerates) and keep only: 'MITM recall is 0.339 at a threshold nobody chose, and no curve shows whether it recovers at a lower one.' Then fold fault 12's base-rate ask in here rather than filing it separately, since both resolve to the same computation.

Two point two million loaded, one hundred twenty thousand used

EVIDENCE

Cell eleven sets N R O W S to None with the comment read the file whole, loads two million two hundred nineteen thousand two hundred one rows, and asserts len of df is at least one million. Cell sixteen holds out five hundred fifty-four thousand eight hundred one and subsamples the training pool to one hundred and twenty thousand. Subtract: two million two hundred nineteen thousand two hundred one minus five hundred fifty-four thousand eight hundred one leaves one million six hundred sixty-four thousand four hundred in the training pool. One hundred and twenty thousand are used. One million five hundred forty-four thousand four hundred rows were read into memory and shown to no model. That is five point four percent of the corpus reaching the learner. The code comment justifies it: because these learners saturate long before then on this data.

SCIENTIFIC CONSEQUENCE

The saturation claim is what licenses discarding ninety-three percent of the available training data, and a grep of all twenty-seven cells for learning underscore curve returns zero. There is no plot, no table and no printed number anywhere showing AUC as a function of training-set size. The subtitle sells two point two million records; the experiment is one hundred and twenty thousand.

REQUIRED REPAIR

A learning curve to saturation across at least five training-set sizes on a log scale, or the effective n stated in the subtitle in place of the corpus size.

DEFENSIBLE REPLY

The assert gates the corpus, not the experiment. And my cross-validation fits on about twenty-six thousand seven hundred rows for 0.9929 against 0.993167 at a hundred and twenty thousand. That is a four-and-a-half-fold change in training size for three ten-thousandths. My held-out set is five hundred and fifty-four thousand rows.

ORDERED CORRECTION

Retitle to 'Saturation is asserted in a comment and never plotted', and acknowledge the two in-notebook points at $n = 26,667$ and $n = 120,000$ - the ask is to extend a two-point curve, which is fair and cheap.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You plotted the diagnostic and declined to read it

EVIDENCE

Cell eleven prints a positive rate of zero point two seven two zero, so benign flows outnumber attacks roughly two point seven to one. Cell eighteen imports precision underscore recall underscore curve, computes prec and rec, and plots them with the title Precision-Recall. A grep of all twenty-seven cells for average underscore precision returns zero. Across all thirty-six notebooks in the series a precision-recall curve is plotted thirty-six times and its area is computed zero times. Cell sixteen ranks by ROC-AUC.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to the false-positive count when negatives dominate, which is the regime this corpus is in. The notebook drew the right diagnostic, which means it cannot claim not to have known which one mattered, and then declined to read it. The correct sentence is not that there is no precision-recall curve — there is one, on axis two of cell eighteen — it is that its area was never computed.

REQUIRED REPAIR

average underscore precision underscore score with a bootstrap confidence interval reported beside ROC-AUC, and precision quoted at a fixed recall.

DEFENSIBLE REPLY

Compute what you are asking for. My precision at threshold 0.5 is 0.9987 and my recall is 0.8908; the average precision you want is about 0.999 against a no-skill line of 0.272. It is a real omission and it changes nothing you would read differently.

ORDERED CORRECTION

Keep the REQUIRED but downgrade the severity and say what the number would be, so the charge reads as a reporting standard rather than a concealed failure. The genuinely uninformative panel on this notebook is per-family, not aggregate.

Atlas proof

docs/course/security-ml-notebooks/nb06-edge-iiotset.ipynb · cell 18; zero-based manifest cell IDs: 18=78aea385

Recalls to three decimals, no denominators

EVIDENCE

Cell eighteen prints: M I T M zero point three three nine, Uploading zero point six eight eight, D D o S underscore H T T P zero point six nine four, Fingerprinting zero point seven one seven, X S S zero point seven three one, Password zero point seven three two. Not one count appears. The only thing the reader learns about how many M I T M positives that zero point three three nine rests on is that there were at least five, because the code contains if c n t is less than five, continue — four would have been silently dropped. A grep of all twenty-seven cells for bootstrap returns zero and for confidence interval returns zero.

SCIENTIFIC CONSEQUENCE

Section thirteen elevates the M I T M number to the operationally important result. An operationally important result with an unknown and possibly single-digit denominator is not a result. And zero point three three nine to three decimals implies a precision the sample cannot support at any plausible n .

REQUIRED REPAIR

Print every family with its positive count, attach a Wilson or Clopper-Pearson interval to every recall, and a bootstrap interval to the held-out AUC.

DEFENSIBLE REPLY

MITM is one thousand two hundred and fourteen rows, about three hundred in the held-out quarter. A Wilson interval on 0.339 at that n runs from 0.29 to 0.39. My weakest number is established to within five points. You are right that I did not print the denominator, and that is a one-line fix.

ORDERED CORRECTION

State the counts rather than 'at least five': MITM 1,214, Fingerprinting 1,001, XSS 15,915, Uploading 37,634, DDoS_HTTP 49,911, Password 50,153. Naming the two genuinely small ones (1,214 and 1,001) is a sharper charge than the floor argument.

The cross-validation validates a different model

EVIDENCE

Cell sixteen fixes every hyperparameter and varies none: RandomForest sixty estimators, XGBoost eighty estimators at max depth six, LightGBM eighty estimators, LogisticRegression max iterations three hundred. A grep of all twenty-seven cells for any hyperparameter search returns zero; across the series it is zero of thirty-six. Cell twenty-four then sets `c v X` to `X t r dot i l o c` up to forty thousand. Three folds over forty thousand rows means each fit sees roughly twenty-six thousand six hundred sixty-seven rows. The headline model was fitted on one hundred and twenty thousand.

SCIENTIFIC CONSEQUENCE

The plus or minus zero point zero zero zero two that section thirteen quotes as evidence of stability is the fold spread of a model trained on about a fifth of the reported one, computed on the same data from which the winner was selected. It bounds the variability of a different estimator, and it is quoted as if it validated the headline.

REQUIRED REPAIR

Nested cross-validation — an inner loop selecting hyperparameters, an outer loop estimating performance — run on the training set that produces the headline.

DEFENSIBLE REPLY

Nested cross-validation removes selection optimism. I select nothing - four fixed configurations, no search. The inner loop would iterate over an empty grid. If you want the CV to validate the pipeline I report, tell me to run it at a hundred and twenty thousand rows.

ORDERED CORRECTION

Replace the REQUIRED with 'run the cross-validation at the headline's $n = 120,000$ and state that the ledger's ± 0.0002 is a spread at $n = 26,667$ '. Then add the selection leak this filing misses, which is the stronger form: `best_name` is chosen by ROC-AUC on `Xte` and every reported diagnostic is computed on that same `Xte`, with no separate validation split anywhere.

Best model declared with no paired test

EVIDENCE

Cell sixteen prints LightGBM at zero point nine nine three one six seven and XGBoost at zero point nine nine two seven two zero. The difference is zero point zero zero zero four four seven. The line best model colon LightGBM is printed on that basis. A grep of all twenty-seven cells for McNemar returns zero; across the series it is zero of thirty-six. Every downstream figure in the notebook — the confusion matrix, the per-family recalls, the false-positive rate of zero point zero zero zero four, the ablation — is computed on the model that difference selected.

SCIENTIFIC CONSEQUENCE

Declaring a winner is a claim, and a claim needs a test. Two learners separated in the fourth decimal on a single split are not distinguishable without one, and the whole results section is conditioned on a selection nobody validated.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, or overlapping bootstrap confidence intervals on the two AUCs.

DEFENSIBLE REPLY

Four hundred forty-seven millionths is more than twice the three-fold cross-validation spread of plus or minus zero point zero zero zero two, so the ordering is at least not obviously inside the noise, and LightGBM also leads on accuracy, zero point nine six nine nine eight against zero point nine six nine two five.

A security detector never meets an adversary

EVIDENCE

A grep of all twenty-seven cells returns zero hits for adversarial, zero for evasion, and zero for any perturbation budget; across all thirty-six notebooks the count is the same. Section two cites Sommer and Paxson, two thousand ten — a paper about why closed-world machine-learning scores rarely survive deployment, false-positive cost among its central reasons — and no experiment below that citation responds to it.

SCIENTIFIC CONSEQUENCE

The features carrying zero point nine nine three one six seven are protocol fields an attacker sets: sequence numbers, reset flags, packet timing. An I I o T attacker who reads Ferrag's paper knows exactly which columns this corpus exposes. Untested against adaptation, the score describes an adversary who does not know a detector exists — which section one already told us is the wrong adversary for constrained, long-lived devices.

REQUIRED REPAIR

An evasion experiment under a stated perturbation budget applied to the attack class, with worst-case per-family recall reported alongside the clean number.

DEFENSIBLE REPLY

Define the budget over what an attacker can actually do - padding, delay, session splitting - and recompute the fields from the modified traffic. Perturbing `tcp.flags` and `tcp.len` and `udp.time_delta` independently produces a row that no packet could have produced, and whatever recall falls out of that describes my feature space, not an adversary.

ORDERED CORRECTION

Require the budget be defined over realisable traffic transformations with fields recomputed, not over the 32 columns directly. Note the charge is filed identically against all thirty-six notebooks, so it argues for one adversarial-robustness notebook in the series.

Redundant separability diagnosed, never measured

EVIDENCE

Cell twenty-two: dropping t c p underscore s e q moves the AUC from zero point nine nine three one six seven to zero point nine nine one five nine zero, a loss of one thousand five hundred seventy-seven millionths. The external probe went further and dropped the model's top two, t c p underscore connection underscore r s t and then t c p underscore s e q, and the AUC held at zero point nine nine one six. Section eleven's preamble already writes the correct reading: if it barely moves, common on simulated corpora, that is not vindication, it means the classes are separable by many redundant features because the attack and benign distributions barely overlap, and that is its own generation artifact. Section thirteen repeats it. No cell measures overlap.

SCIENTIFIC CONSEQUENCE

The notebook writes down the diagnosis and then declines to run the one measurement that would convert prose into a finding. Class-conditional overlap is testable in a dozen lines. Writing the correct interpretation and printing an AUC instead is disclosure standing in for measurement, and it is the reason a reader cannot tell from this notebook whether Edge-I I o T set is a hard problem or a well-separated simulation.

REQUIRED REPAIR

Nearest-neighbour distance from each attack row to the nearest benign row and back, or a two-sample divergence estimate, reported as a distribution rather than a sentence.

DEFENSIBLE REPLY

Your own citation gives the ablation delta as fifteen hundred millionths - the largest in the four notebooks you hold - and then calls it 'barely moves'. Section eleven's sentence is a template for the collapse case on simulated corpora. Edge-IIoTset is a physical testbed with real MQTT and Modbus devices.

ORDERED CORRECTION

Drop the 'simulated corpora' framing for this notebook and restrict the charge to the one unmeasured sentence in section 13. Mark the top-two-drop figures (0.9916) as external-probe output, not notebook output.

An operating point measured at the wrong prior

EVIDENCE

Cell eleven prints a positive rate of zero point two seven two zero. Cell eighteen prints a false-positive rate of zero point zero zero zero four, one hundred and eighty benign flows flagged of four hundred three thousand nine hundred eleven. No cell states a deployment base rate, and no cell converts zero point zero zero zero four into alerts per day or alerts per analyst. Section two's own related-approaches table names the hazard — for typical deep-learning N I D S work it records that the benign-majority base rate inflates accuracy — and applies it to other people's papers rather than to this one.

SCIENTIFIC CONSEQUENCE

An operating point stated without the prior it was measured under is not transferable. A testbed in which twenty-seven percent of all flows are attacks is not a production I I o T network, and the notebook never says what changes when the prior does.

REQUIRED REPAIR

State the assumed deployment base rate explicitly, and report precision and alerts per analyst per day at that base rate alongside the false-positive rate.

DEFENSIBLE REPLY

At a false-positive rate of zero point zero zero zero four the arithmetic is favourable at any plausible volume — even a hundred million flows a day yields forty thousand alarms — so performing the calculation would not change any conclusion in this notebook.

13

MAJOR UNTESTED VERIFIED NO CLASS WEIGHTING

The minority class is never weighted

EVIDENCE

A grep of all twenty-seven cells for class underscore weight returns zero and for scale underscore pos underscore weight returns zero; across all thirty-six notebooks in the series both counts are zero. Cell sixteen instantiates LogisticRegression, RandomForest, XGBoost and LightGBM with no weighting argument. The positive rate is zero point two seven two zero.

SCIENTIFIC CONSEQUENCE

Unweighted log-loss on a two point seven to one imbalance pushes the decision boundary toward the majority class, which is precisely the direction that produces an M I T M recall of zero point three three nine alongside a false-positive rate of zero point zero zero zero four. The notebook's worst operational number and its missing weighting point the same way, and nobody tested the connection.

REQUIRED REPAIR

Refit with class underscore weight set to balanced, or scale underscore pos underscore weight set to the negative-to-positive ratio, and report per-family recall before and after.

DEFENSIBLE REPLY

Class weighting and threshold movement are largely interchangeable for a probabilistic ranker, and the headline metric is ROC-AUC, which is invariant to both — so the omission changes the operating point rather than the reported result.

14

MAJOR DOCUMENTED VERIFIED SINGLE SEED FOR THE HEADLINE

One draw, reported to six decimals

EVIDENCE

Cell twenty-four prints seed equals zero. Every split uses random underscore state equals RANDOM underscore STATE. The eight-seed stability check that exists for this notebook was run by the external praxis probe, not by the notebook, and it probed the single-feature audit — the zero point seven two two nine on t c p underscore s e q — not the zero point nine nine three one six seven.

SCIENTIFIC CONSEQUENCE

Six decimal places on a single draw invites the reader to treat the sixth decimal as real, and the model-selection margin between the top two learners lives in the fourth.

REQUIRED REPAIR

At least ten seeds through the full pipeline, with mean and standard deviation reported in place of the point estimate.

DEFENSIBLE REPLY

The three-fold cross-validation reports plus or minus zero point zero zero zero two, which is small enough to make a seed-driven change in the headline's first three decimals implausible.

Fourteen families, twelve charted, six printed

EVIDENCE

The external probe reports fourteen groups with zero skipped by the count-of-five floor. Cell eighteen then computes show equals the concatenation of s r t dot head of nine and s r t dot tail of three when the series exceeds twelve entries — worst nine plus best three — and the printed line takes only the six worst. Fourteen minus twelve leaves two families that reach neither the chart nor the printed line, and the notebook prints no count of what it displayed versus what it computed.

SCIENTIFIC CONSEQUENCE

The truncation is silent. A reader cannot reconstruct the full table from any output, and cannot verify that the omitted families are the mid-ranked ones rather than something worse.

REQUIRED REPAIR

Print the complete per-family table with counts and recalls for every family, and label the chart explicitly as a truncated view.

DEFENSIBLE REPLY

The truncation keeps the worst nine, so the two omitted families are mid-ranked by construction and the conclusion's worst-case reading is unaffected — the display is conservative in the direction that matters.

Credited to one party, downloaded from another

EVIDENCE

The section-three provenance table gives Source as Kaggle s i b a s i s p r a d h a n slash edge dash i i o t s e t dash dataset. The abstract and reference one credit Ferrag and colleagues, two thousand twenty-two, I E E E Access. Those are different parties. Cell eleven downloads whatever that account currently hosts, asserts a single canonical filename and a one-million-row floor, and computes no hash and no manifest against the authors' own artifact.

SCIENTIFIC CONSEQUENCE

The reported quantity is a property of a file and the notebook cannot establish which file. A re-upload can be filtered, re-ordered, de-duplicated or partial, and the fifteen point one percent duplicate rate and zero point zero three three contamination are exactly the properties a re-upload can alter.

REQUIRED REPAIR

Download from the authors' distribution, or compute a checksum of the mirror against the published artifact, and print the comparison.

DEFENSIBLE REPLY

The loader reads two million two hundred nineteen thousand two hundred one rows by thirty-two features, the external probe independently re-read the identical shape, and the positive rate reproduces to five decimals — so the mirror is at minimum the right corpus at the right size.

No number can be placed beside a prior result

EVIDENCE

Section two's related-approaches table lists Ferrag and colleagues' deep-learning and machine-learning baselines with the caveat in-distribution, protocol fields can leak, and federated-learning studies with the caveat non-I I D splits, optimistic. The table is explicitly labelled: these are not measurements reproduced on this exact corpus. No cell reproduces the authors' split, their feature selection, or their metric.

SCIENTIFIC CONSEQUENCE

A number that cannot be placed beside any prior number on the same protocol is a local observation, not a scientific result. The reader is left unable to say whether zero point nine nine three is better, worse or identical to what Ferrag reported on the same file.

REQUIRED REPAIR

Run the dataset authors' published protocol and report your figure beside theirs in the same table, enumerating every preprocessing difference.

DEFENSIBLE REPLY

The table says on its face it is not reproduced measurement. I declined the comparison in writing rather than faking it.

ORDERED CORRECTION

Recast as an omission of an affordable experiment rather than a misrepresentation, and drop the 'no cell reproduces their split' framing, which attacks a claim the table disclaims.

The transfer notebook is thirty rows away

EVIDENCE

Section one states the requirement: I I o T devices are constrained and long-lived, so a detector must generalize. Section thirteen states the scope: every number above therefore measures in-distribution separability only. The same directory contains notebook thirty-six, titled cross-distribution transfer. A grep of this notebook's twenty-seven cells finds no second corpus, no transfer split, and no train-on-A test-on-B experiment of any kind.

SCIENTIFIC CONSEQUENCE

The excuse of unavailability is unavailable. The method exists in the same codebase, written by the same author, in the same course. Notebook thirty-six is the transfer notebook: the capability is written, tested and sitting in the same directory as this file, and it was never pointed at this corpus. The experiment was affordable and was not run.

REQUIRED REPAIR

Train on Edge-I I o T set and test on a second I o T flow corpus, and the reverse, reporting both directions and the drop from the in-distribution figure.

DEFENSIBLE REPLY

Open notebook thirty-six. Its loader comment says the shared forty-three-field NetFlow version two schema is what makes its comparison possible. My features are tshark protocol fields - mqtt.hdrflags, arp.opcode, icmp.seq_le. There is no column in common. Pointing it at my file is not thirty rows, it is a feature-alignment project, and there is no second IoT corpus in this series with my schema either.

ORDERED CORRECTION

Re-mechanise to the weaker true claim: 'No cross-corpus evaluation of any kind is attempted, and the series demonstrates the capability exists - but nb36's apparatus does not transfer here, because it operates on a shared 43-field NetFlow-v2 schema this corpus does not share.' Also note it duplicates fault 1's second alternative, so it should be a clause of fault 1 rather than an independent FATAL.

An A for the absence of a mode this corpus lacks

EVIDENCE

Cell twenty computes the grade from literal thresholds: F at a single-feature AUC of zero point nine nine nine, D at zero point nine nine, C at zero point nine five, B at zero point eight five, otherwise A; and for contamination F at zero point five, D at zero point three, C at zero point one five, B at zero point zero five, otherwise A. The values are zero point seven two two nine and zero point zero three three, so both score A. Section thirteen then writes: neither check flags a problem, so nothing here explains the score away.

SCIENTIFIC CONSEQUENCE

That last clause is the overreach. The scheme has exactly two failure modes and this corpus's dominant pathology — the ablation showing the score survives the removal of the top two features, which section eleven itself calls a generation artifact — has no cutoff and no letter. An A is awarded for the absence of a failure mode the corpus does not have, and the conclusion converts it into a general clearance. The sibling notebook on C I C-I D S two thousand seventeen gets this right and writes read that grade narrowly; this one does not.

REQUIRED REPAIR

Report the underlying quantities with confidence intervals and drop the letters, or add a sub-check whose cutoffs respond to multi-feature redundancy — for instance the AUC after dropping the top k features.

DEFENSIBLE REPLY

The scheme is defined in the cell that computes it, the two sub-checks are named, and the notebook never claims the grade is a corpus-wide verdict — a reader who reads the code sees exactly what the letter covers.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 554,801 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. None of that saves it. You wrote, in your own first section, the exact sentence that decides this defence: a detector must generalise, which is exactly what an in-distribution score cannot show. You then dropped the host-identity columns because you understood they let the model read who is talking rather than what the traffic does — and split the rows at random anyway, leaving every host on both sides of the boundary. The grouping key was in your hand. You deleted it as a feature and declined to use it as a group. That is not an oversight; it is the experiment you named and did not run.

Network Intrusion Detection on CIC-IDS2017

C I C-I D S two thousand seventeen, two million eight hundred thirty thousand seven hundred forty-three C I C Flow Meter flows by seventy features from eight daily capture files, positive rate zero point one nine seven zero, run on the original uncorrected labels and features.

HEADLINE

XGBoost, held-out ROC-AUC zero point nine nine nine nine four eight, accuracy zero point nine nine eight nine nine one, three-fold cross-validated zero point nine nine nine nine plus or minus zero point zero zero zero one.

DEFENSE

F

Selected-model estimand

DOCTORAL

6

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9.5

/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 5 critical (FATAL)
- 16 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

The provenance section here is the most honest thing in the four notebooks I hold, and I want that on the record. It names Engelen and Rosay, states flatly that it runs the original uncorrected labels and features, and tells you in the abstract that those papers stand as a limit on every number below — a known defect in the input that we did not remove. It reads its own A trust grade narrowly and says in terms that the grade does not clear the corpus. And it leads its per-family results with Infiltration at zero rather than burying it, on a detector whose false-positive rate is four hundred fifty-one alarms in five hundred sixty-eight thousand benign flows. That is genuinely good practice.

EXAMINER RULING

A held-out area under the curve of zero point nine nine nine nine four eight, on the exact files two cited papers say are mislabelled, drawn at random across eight days that the loader concatenated and threw away — and the notebook's own sentence is that this is the kind of figure those papers argue is inflated, so I am ruling with the author against the author.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Eight days concatenated and thrown away

EVIDENCE

The section-three provenance table gives Rows as approximately two point eight million flows, eight daily C S V s combined. Cell eleven reads: files equals sorted glob of the directory, then d f equals p d dot concat of a list comprehension over those files. The loader iterates a variable literally called files, and the comprehension never tags a single row with the file it came from. That omission is where the day identity is lost, and the fix sits inside the same expression: read each C S V and assign a day column equal to its filename. Cell sixteen then calls train underscore test underscore split with stratify on y, no group, and no time ordering. Across all thirty-six notebooks in this series, a temporal or time-ordered split appears zero times and GroupKFold appears zero times.

SCIENTIFIC CONSEQUENCE

C I C-I D S two thousand seventeen is a scheduled corpus: each day runs a different attack programme. That structure is not incidental — it is the design of the dataset, and it is the natural evaluation protocol. The day index was in the loader's hands, in a variable called files, inside the very expression that discarded it, and section three already tells the reader there are eight daily C S V s. Splitting at random across days means the model trains on flows from the same attack episode it is tested on, which is why zero point nine nine nine four eight is achievable at all. The correct experiment is leave-one-day-out, and it costs a groupby.

REQUIRED REPAIR

Tag every row with its source file at load time — concatenate read underscore c s v of f dot assign of day equals f, across files — then run leave-one-day-out cross-validation — train on seven days, test on the eighth, rotate — and report the per-day held-out AUC and the drop from the random-split figure.

Five nines on files two papers say are wrong

EVIDENCE

Section two cites Engelen, Rimmer and Joosen, two thousand twenty-one, found labeling errors and C I C Flow Meter feature bugs that inflate scores, and Rosay and colleagues, two thousand twenty-two, documents C I C Flow Meter implementation flaws affecting these flow features. Section three says: we load the original Kaggle C S V s, not either group's corrected re-release, those defects are therefore present in our features and labels, they are nowhere corrected, measured, or worked around in this notebook. And section thirteen says, in the author's own words, a held-out AUC of zero point nine nine nine nine four eight on original-label C I C-I D S two thousand seventeen is the kind of figure those papers argue is inflated.

SCIENTIFIC CONSEQUENCE

I am ruling with the author against the author. The notebook states that its headline is the kind of number the cited literature says is an artifact, and prints the number as the result. The corrected re-release exists — Engelen's group published it precisely so that people would stop doing this — and the notebook names it, locates it, and declines to run it. Naming a fatal limitation is a confession, not a defence, and this is the crispest confession in the four notebooks I hold.

REQUIRED REPAIR

Refit on the Engelen corrected labels and re-extracted flows, report the AUC beside the original-label figure, and make the difference between them the headline result of the notebook.

The exclusion set is the empty set

EVIDENCE

Cell eleven reads: D R O P equals list of the set containing LABEL, FAMILY, y and family, unioned with s e t of the empty list. The empty list is the entire hand-written exclusion set for this notebook. The only other filter is: i d l i k e equals the object-dtype columns whose number of unique values exceeds zero point five times the length of X. Two conditions, both required. On two million eight hundred thirty thousand seven hundred forty-three rows, a column must be string-typed and carry more than one million four hundred fifteen thousand three hundred seventy-one distinct values to be removed. No numeric identifier, no numeric port, and no numeric timestamp can ever satisfy that. Meanwhile section three lists exactly two controls the notebook claims — quote, identifier slash timestamp-like columns are dropped, and we report per-family recall rather than accuracy alone.

SCIENTIFIC CONSEQUENCE

One of the two stated controls is a filter that cannot fire on the column types at risk, and no cell prints what it removed. So the notebook cannot establish that any apparatus-coupled column was excluded from its seventy features. Contrast the sibling notebook one position earlier in this series, on Edge-I I o T set, which names twenty-four columns to drop and explains why: in a testbed capture the attacker and victim addresses are fixed per role, so keeping those columns lets the model read who is talking instead of what the traffic does. The same author wrote both. Here the list is empty.

REQUIRED REPAIR

Enumerate and print the retained feature names, hand-audit them for apparatus and identity coupling, drop the coupled block, refit, and report the AUC on behaviour-only features beside the headline.

Infiltration recall zero point zero zero zero, out of what?

EVIDENCE

Cell eighteen prints: worst per-family recalls, Infiltration zero point zero, Web Attack S q l Injection zero point five, Bot zero point seven one three, Web Attack X S S zero point nine three four, Web Attack Brute Force zero point nine nine two, F T P-Patator zero point nine nine six. Not one count appears. The only thing the reader can infer about the Infiltration denominator comes from the code: the floor is if c n t is less than five, continue, so there were at least five — because four would have been dropped the way Heartbleed was. At least five. That is the entire state of the reader's knowledge. A grep of all twenty-seven cells for bootstrap returns zero and for confidence interval returns zero; across the series it is zero of thirty-six.

SCIENTIFIC CONSEQUENCE

Section thirteen elevates this to the operationally important result — quote, at threshold zero point five the model detects none of it. A recall of zero on five positives and a recall of zero on five thousand positives are different scientific claims, and this notebook prints them identically. Work the worst case: the two-sided ninety-five percent Clopper-Pearson upper bound on zero successes in five trials is one minus zero point zero two five to the power one fifth, which is about zero point five two. On that evidence a true recall of one half is not excluded. The finding as stated cannot be defended in either direction.

REQUIRED REPAIR

Print every family's positive count, attach a Clopper-Pearson interval to every recall including the zeros, and give a bootstrap interval on the held-out AUC.

Heartbleed is deleted without a printed word

EVIDENCE

Cell eighteen contains the line: if c n t is less than five, continue, with the comment need a few positives for a meaningful recall. The external praxis probe re-ran that stage and reports fourteen groups with one skipped, and it names the skipped family: Heartbleed. No cell prints a skip count, a skip list, or the number of families evaluated. The display then narrows twice more: the chart shows at most twelve, worst nine plus best three, and the printed line takes only the six worst. So an attack family vanished at the floor, more vanished at the chart, and the conclusion quotes six.

SCIENTIFIC CONSEQUENCE

Heartbleed is one of the marquee attacks in this corpus. It disappeared from the results with no trace in any output, and the only reason anyone knows is that an external re-run found it. The floor is defensible as a policy; the silence is not. And the notebook's own conclusion presents the worst per-group recall list as, quote, the operationally important result — a list that omits the family it could not score.

REQUIRED REPAIR

Print every family with its held-out positive count. Where the count is below the floor, report the family as unscorable with its count, never omit it. Print the number of families evaluated, charted and printed at each stage.

DEFENSIBLE REPLY

Heartbleed is eleven flows in two point eight million. Three of them are in my held-out quarter. There is no floor at which that family is scoreable and no number I could have printed for it except the count. You are right that I should have printed the count and the skip. It changes nothing I reported.

ORDERED CORRECTION

Replace the external-probe citation with the corpus fact, which is stronger and independent: 'Heartbleed is 11 flows in 2,830,743; roughly 3 reach the held-out quarter and the $n \geq 5$ floor deletes the family without a word.' Then merge the counting half into fault 5, keeping only the display-arithmetic charge (evaluated, charted, printed) here.

Atlas proof

docs/course/security-ml-notebooks/nb07-cic-ids2017.ipynb; docs/course/security-ml-notebooks/assignments/evidence/SEAS8414-notebook-review-key.html · nb07 criterion 10 and probe B4

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

A winner declared on seven millionths

EVIDENCE

Cell sixteen prints XGBoost at zero point nine nine nine nine four eight and LightGBM at zero point nine nine nine nine four one. Subtract them: seven millionths. The line printed is best model colon XGBoost. Cell twenty-four then reports the three-fold cross-validated AUC as zero point nine nine nine nine plus or minus zero point zero zero zero one — the fold spread is fourteen times the margin that decided the winner. A grep of all twenty-seven cells for McNemar returns zero; across all thirty-six notebooks it is zero.

SCIENTIFIC CONSEQUENCE

Nothing distinguishes these two learners on this split. Seven millionths of AUC on seven hundred and seven thousand six hundred eighty-six held-out rows is a handful of ranking swaps. Yet the confusion matrix, the false-positive rate of zero point zero zero zero eight, the per-family recalls including Infiltration at zero, and the entire ablation table are all computed on the model that seven millionths selected. Change the seed and the results section may be about a different model.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, and a statement in the conclusion that the learners are indistinguishable if the test says so.

DEFENSIBLE REPLY

I sorted a table and took the top row. Your fourteen-times figure sets an unpaired fold spread against a paired difference on the same test set, where the two models' errors are almost perfectly correlated - which is precisely why the test you ask for is the paired one, and why the fold spread is not the yardstick. At 0.99995 nothing downstream would change either way.

ORDERED CORRECTION

Delete the '14 times the margin' comparison and keep only: section 13 writes 'Winning learner: XGBoost' unqualified where a McNemar test would almost certainly return indistinguishable. Align with nb06 fault 9, which files this identical charge at CONTESTED.

Atlas proof

docs/course/security-ml-notebooks/nb07-cic-ids2017.ipynb · Code cell 16 output; zero-based manifest cell IDs: 16=ed6e4487

An ablation table decided in the sixth decimal

EVIDENCE

Cell twenty-two prints: headline as-is zero point nine nine nine nine four eight; de-duplicated with eleven percent of rows removed zero point nine nine nine nine four six; shortcut feature dropped zero point nine nine nine nine five two. The largest gap in that column is six millionths, and the middle value is lower than both. Section thirteen then narrates them as distinct outcomes — the ablation refutes a single-feature story, dropping that feature barely moves the AUC, de-duplication lowers the AUC only slightly. Cell twenty-four's own three-fold spread is plus or minus zero point zero zero zero one, which is sixteen times the entire range of the table. And the notebook's own code comment on the rounding says six d p, a one point zero zero zero zero zero zero is a red flag, not a win.

SCIENTIFIC CONSEQUENCE

The estimator cannot resolve differences at this scale, and the notebook knows it — section thirteen says each ablation re-splits and refits, so tiny differences are re-split noise, and then interprets them anyway. A table in which every row is inside the noise floor supports no ordering and no conclusion. Reporting to a resolution the estimator does not support is how an artifact acquires the appearance of a finding.

REQUIRED REPAIR

Report each ablation setting as a mean plus or minus a standard deviation over at least ten re-splits, at the resolution that spread supports, and state explicitly which differences are resolvable.

DEFENSIBLE REPLY

Read my sentence rather than the table. Section thirteen says each ablation re-splits and refits, so tiny differences are re-split noise. That is your objection, written by me, before you made it. What I said the table shows is that the AUC barely moves, that de-duplication changes it only slightly, and that repeated rows are negligible - three statements that all mean the differences are unresolvable, which is what you want me to say.

ORDERED CORRECTION

Drop 'narrates them as distinct outcomes' - it is refuted by the notebook's own sentence. The residual charge is one directional verb applied to a two-millionth gap ('De-duplication lowers the AUC only slightly') plus the deeper point, which is the good one: at a headline of 0.999948 no ablation can show anything except collapse, so the whole ablation apparatus has no resolving power on this corpus. Keep the ten-re-splits REQUIRED.

Two point eight million loaded, one hundred twenty thousand used

EVIDENCE

Cell eleven contains assert len of d f is at least one million and loads two million eight hundred thirty thousand seven hundred forty-three rows. Cell sixteen holds out seven hundred seven thousand six hundred eighty-six and subsamples the training pool to one hundred and twenty thousand. Subtract: two million eight hundred thirty thousand seven hundred forty-three minus seven hundred seven thousand six hundred eighty-six leaves two million one hundred twenty-three thousand fifty-seven in the pool, of which one hundred and twenty thousand are used. Two million three thousand fifty-seven rows were read and shown to no model. That is four point two percent of the corpus reaching the learner. The justifying comment reads: because these learners saturate long before then on this data.

SCIENTIFIC CONSEQUENCE

The saturation claim carries the whole decision to discard ninety-four percent of the training pool, and a grep of all twenty-seven cells for learning underscore curve returns zero. No plot, no table, no printed number anywhere shows AUC against training-set size. The subtitle sells greater-than-or-equal-to one million records and the experiment is one hundred and twenty thousand.

REQUIRED REPAIR

A learning curve to saturation across at least five training-set sizes, or the effective n stated in the abstract in place of the corpus size.

DEFENSIBLE REPLY

The assert gates the corpus. And I have two points on the curve you want: about twenty-six thousand seven hundred training rows gives 0.9999, a hundred and twenty thousand gives 0.999948. At this separability the learner saturates far below either.

ORDERED CORRECTION

Retitle to 'Saturation is asserted in a comment and never plotted' and concede the two in-notebook points; the ask is to extend a two-point curve.

The scope paragraph is a confession

EVIDENCE

Section thirteen names three: the split is random, not temporal or entity-grouped, so every number measures in-distribution separability only; settling the Engelen and Rosay charge would require refitting on the corrected labels and re-extracted flows, which is not something this notebook does; and nothing computed in this notebook can confirm or rule that out. Section eleven names a fourth in advance, that a headline surviving ablation on a simulated corpus is not vindication. Four named experiments, zero run.

SCIENTIFIC CONSEQUENCE

The affordability of each is the point. The day tag is one line in the loader. The corrected label release is a download. A grouped split is twenty lines. Cross-corpus transfer exists as notebook thirty-six in this same directory. At doctoral level, identifying the experiment that would settle your central claim, writing down that you did not run it, and publishing the headline anyway is not caution — it is a confession that the reader is asked to accept in place of evidence.

REQUIRED REPAIR

Run one of the four, report its result as the headline, and demote the random-split figure to a reference number.

DEFENSIBLE REPLY

You have charged me for the temporal split, for the corrected labels, for the random split and for the overlap measurement, and then charged me a fifth time for not doing one of those four. That is one experiment set, not five faults.

ORDERED CORRECTION

Keep the E4 framing as the pack's summary charge - it is the thesis and it is well supported here - but present it explicitly as the roll-up of faults 1, 2 and 16 rather than as an independent finding, so the FATAL count is not inflated by re-filing.

You plotted the diagnostic and declined to read it

EVIDENCE

Cell eleven prints a positive rate of zero point one nine seven zero, so benign flows outnumber attacks roughly four to one. Cell eighteen imports precision underscore recall underscore curve, computes prec and rec, and plots them on axis two under the title Precision-Recall. A grep of all twenty-seven cells for average underscore precision returns zero. Across all thirty-six notebooks in this series the curve is plotted thirty-six times and its area computed zero times. Cell sixteen ranks by ROC-AUC.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to false-positive count when negatives dominate, and this is a four-to-one corpus. The notebook drew the correct diagnostic, which forecloses any claim that it did not know which one mattered, and then did not read it. The precise charge is not that there is no precision-recall curve — there is one, in the third panel of cell eighteen — it is that its area was never computed and never used to rank anything.

REQUIRED REPAIR

average underscore precision underscore score with a bootstrap confidence interval reported beside ROC-AUC, and precision quoted at a fixed recall.

DEFENSIBLE REPLY

Compute what you are asking for. My precision at threshold 0.5 is 0.9968 and my recall is 0.9981; the average precision you want is about 0.9999 against a no-skill line of 0.197. It is a real omission, and it moves nothing. The number that matters on this notebook is Infiltration at nine flows, and you have that charge already.

ORDERED CORRECTION

Downgrade and say what the number would be. The informative version on this corpus is per-family, not aggregate: precision at a fixed recall for Infiltration and Web Attack Sql Injection, which is faults 5 and 12.

Nobody moved the dial for Infiltration

EVIDENCE

Cell eighteen: pred equals p b greater than zero point five. That constant produces the confusion matrix, the false-positive rate of zero point zero zero zero eight — four hundred fifty-one benign flows flagged of five hundred sixty-eight thousand two hundred seventy-four — and every per-family recall including Infiltration at zero point zero. A grep of all twenty-seven cells returns zero for threshold selection and zero for a cost matrix; across the series both are zero of thirty-six.

SCIENTIFIC CONSEQUENCE

Four hundred fifty-one false alarms across five hundred sixty-eight thousand benign flows is an enormous unspent budget. The notebook is detecting none of an attack family while holding that budget in reserve, and no cell asks what happens to Infiltration recall at a threshold of zero point one, or zero point zero one. The operating point is the result for an asymmetric-cost detector, and here the operating point was a library default. This is also why the operational-admissibility axis scores two out of five here even though this notebook's false-positive rate is among the best of the four I hold: the axis grades whether an operating point, a cost model and an alert budget exist, not whether the number is flattering.

REQUIRED REPAIR

A stated cost matrix in which a missed Infiltration flow carries an explicit cost in units of false alarms, a threshold chosen to minimise expected cost, and a curve of Infiltration recall against alert volume across the threshold range.

The cross-validation validates a different model

EVIDENCE

Cell sixteen fixes every hyperparameter and varies none: RandomForest sixty estimators, XGBoost eighty estimators at max depth six, LightGBM eighty estimators, LogisticRegression max iterations three hundred. A grep of all twenty-seven cells for any hyperparameter search returns zero. Cell twenty-four sets `c v X to X t r dot i l o c` up to forty thousand; three folds over forty thousand rows means each fit sees roughly twenty-six thousand six hundred sixty-seven rows. The headline model was fitted on one hundred and twenty thousand.

SCIENTIFIC CONSEQUENCE

The plus or minus zero point zero zero zero one is the fold spread of a model trained on about a fifth of the reported one, computed on the same data from which the winner was selected. It bounds a different estimator than the one in the abstract and it is quoted as if it validated the headline. And with no search at all, no claim can be made that any learner here was fairly configured.

REQUIRED REPAIR

Nested cross-validation — inner loop selects, outer loop estimates — on the training set that produces the headline.

DEFENSIBLE REPLY

There is no hyperparameter search to nest. Four fixed configurations, no grid, no random search. If you want the cross-validation to validate the pipeline I report, tell me to run it at a hundred and twenty thousand rows.

ORDERED CORRECTION

Replace the REQUIRED with 'run the cross-validation at the headline's $n = 120,000$ '. Then add the selection leak this filing misses, which is the stronger version: `best_name` is chosen by ROC-AUC on `Xte` and every reported diagnostic is computed on that same `Xte`, with no separate validation split.

One draw, six decimals, a seven-millionth margin

EVIDENCE

Cell twenty-four prints seed equals zero. Every split uses random underscore state equals RANDOM underscore STATE. The eight-seed stability check for this notebook was run by the external praxis probe, not by any cell, and it probed the single-feature audit — zero point seven three nine to zero point seven four six seven on Min underscore Packet underscore Length — not the zero point nine nine nine nine four eight and not the model ranking.

SCIENTIFIC CONSEQUENCE

The margin that selected XGBoost over LightGBM is seven millionths. A quantity that small is a property of one draw, not of the learners, and every result in section nine is conditioned on it.

REQUIRED REPAIR

At least ten seeds through the full pipeline, reporting mean and standard deviation and how often each learner wins.

DEFENSIBLE REPLY

The three-fold cross-validation reports plus or minus zero point zero zero zero one and the external probe found the single-feature audit stable across eight seeds, so the headline's first four decimals are not plausibly seed-driven even if the ranking is.

A security detector never meets an adversary

EVIDENCE

A grep of all twenty-seven cells returns zero hits for adversarial, zero for evasion, and zero for any perturbation budget; across all thirty-six notebooks the counts are the same. Section two cites Sommer and Paxson, two thousand ten — closed-world machine-learning scores rarely survive deployment — and Apruzzese and colleagues, two thousand twenty-three, on where machine learning is and is not actually deployed in security practice. No experiment below either citation responds to it.

SCIENTIFIC CONSEQUENCE

The single strongest column is Min underscore Packet underscore Length at zero point seven four three one, and the model's ranked top features per the external probe are Bwd underscore Packet underscore Length underscore Std and Average underscore Packet underscore Size. Those are packet sizes. An attacker changes packet sizes by padding, at essentially zero cost and zero loss of capability. A detector whose leading features are trivially controllable by the adversary, evaluated only against an adversary who does not adapt, has produced a number about a static tool run, not about detection.

REQUIRED REPAIR

An evasion experiment under a stated perturbation budget — bounded padding and inter-packet delay on the attack class — with worst-case per-family recall reported alongside the clean figure.

DEFENSIBLE REPLY

Define the budget over what an attacker can do to traffic - padding, inter-packet delay - and recompute the aggregates from the modified flow. Perturbing seventy CICFlowMeter statistics independently produces a row no flow could have produced.

ORDERED CORRECTION

Require the budget be defined over realisable traffic transformations with aggregates recomputed. Note the charge is filed identically against all thirty-six notebooks.

The right interpretation, written and not measured

EVIDENCE

Section eleven, verbatim: if it barely moves, common on simulated corpora, that is not vindication, it means the classes are separable by many redundant features because the attack and benign distributions barely overlap, that is its own generation artifact. Cell twenty-two then shows the AUC moving by six millionths across all three settings, and the external probe drops the model's top two features with the AUC holding at zero point nine nine nine nine. Section thirteen concludes: so the separability is multi-feature, that reflects how this corpus was generated, not one leaky column. And no cell measures overlap, divergence, or nearest-neighbour margin.

SCIENTIFIC CONSEQUENCE

The notebook diagnoses its own corpus in advance, confirms the diagnosis with the ablation, states the diagnosis in the conclusion — and never runs the measurement that would turn it into a finding. That measurement is a dozen lines. Instead the reader gets a five-nines AUC in the abstract and a paragraph of prose thirteen sections later. Disclosure is standing in for measurement.

REQUIRED REPAIR

Nearest-neighbour distance from each attack row to the nearest benign row and back, or a two-sample divergence estimate, reported as a distribution, with the claim of barely-overlapping distributions either supported or withdrawn.

DEFENSIBLE REPLY

The sentence you want measured has an explanation already in my own literature review, and you have charged me for not testing it in fault two. Measuring nearest-neighbour margin would not distinguish genuine separability from the labelling artifact Engelen documents; refitting on Engelen's labels would. That is one experiment.

ORDERED CORRECTION

Merge into fault 2, or keep and restrict to the single unmeasured sentence. Mark the top-two-drop figure (0.9999) as external-probe output. Note that at 0.999948 the 'barely overlapping distributions' reading is nearly forced, unlike on nb05 where the same template sentence sits above an AUC of 0.8745.

Three prior results named, none reproduced

EVIDENCE

Section two's related-approaches table lists Sharafaldin and colleagues' random forest on C I C-I D S two thousand seventeen, Engelen and colleagues' re-labelled version, and typical deep-learning N I D S papers, each with a caveat. The table is explicitly labelled: these are not measurements reproduced on this exact corpus. No cell reproduces any of their splits, feature sets, or metrics. The references list five papers and the experiment section compares against none of them.

SCIENTIFIC CONSEQUENCE

A number that cannot be placed beside any prior number on the same protocol is a local observation. The corpus in question is the single most benchmarked flow I D S dataset in the field, so the comparison points are abundant and the notebook occupies none of them. Reporting zero point nine nine nine four eight in isolation gives the reader no way to say whether it is high, normal or symptomatic.

REQUIRED REPAIR

Reproduce at least Sharafaldin's published protocol and Engelen's corrected-label protocol and report all three figures in one table with the preprocessing differences enumerated.

DEFENSIBLE REPLY

The table disclaims itself in its own caption. And the Engelen protocol you want me to reproduce is the same experiment you already charged me for in fault two.

ORDERED CORRECTION

Strike the Engelen clause (fault 2 owns it) and keep only the Sharafaldin reproduction, which is genuinely independent and affordable.

A week of enterprise traffic that nobody sent

EVIDENCE

The abstract, cell one, reads: a widely-used flow I D S corpus covering a week of enterprise traffic and staged attacks. Section two is more careful and says realistic profiled benign traffic. Section eleven is more careful still and refers to simulated corpora. So the notebook's own three descriptions of its benign class disagree, and the loosest one is the one in the abstract, which is the sentence a reader quotes.

SCIENTIFIC CONSEQUENCE

The word enterprise carries a claim about provenance that the corpus does not support: the benign flows were synthesised from behavioural profiles, not captured from an operating enterprise. Every generalisation claim about false-positive rate rests on the benign class being representative, and this one is generated. The stricter language exists inside the notebook, which is what makes the abstract's phrasing a choice.

REQUIRED REPAIR

Use the section-two wording in the abstract — profiled benign traffic — and restrict any false-positive claim to the profile generator that produced it.

DEFENSIBLE REPLY

Section two says profiled benign traffic and section eleven says simulated corpora, so the notebook does disclose the provenance accurately in two places; enterprise traffic in the abstract is loose shorthand for the deployment setting the corpus models, not a claim of live capture.

An A on hard cutoffs the corpus never approaches

EVIDENCE

Cell twenty computes the grade from constants: F at a single-feature AUC of zero point nine nine nine, D at zero point nine nine, C at zero point nine five, B at zero point eight five, otherwise A; and for contamination F at zero point five, D at zero point three, C at zero point one five, B at zero point zero five, otherwise A. The observed values are zero point seven four three one and zero point zero four nine, so both score A — but the contamination sits zero point zero zero one below the B boundary, and the external probe's random-window recomputation gives zero point zero four seven three against the notebook's zero point zero four nine two. A shift of one thousandth in the estimate crosses the letter.

SCIENTIFIC CONSEQUENCE

A letter whose boundary is one thousandth away from the observed value, on a quantity estimated from fifty thousand rows with no confidence interval, carries no information. The grade is a rounding of a noisy statistic against an arbitrary constant, and the conclusion quotes it as a finding.

REQUIRED REPAIR

Report contamination with a binomial confidence interval and drop the letter, or state the letter together with the probability that a resample would cross the boundary.

DEFENSIBLE REPLY

The notebook explicitly instructs the reader to read that grade narrowly, states that it covers only two named failure modes, and says outright that it is not a verdict that the score is real and does not clear the corpus — so it never claims the letter is a measurement.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 707,686 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. But candour is not an experiment. You had eight days in a variable called `files` and you destroyed the boundary with `ignore_underscore_index_equals=True`. You named the corrected label `release` and did not download it. You wrote that your own headline is the kind of figure the literature calls inflated, and you printed it in the abstract to six decimal places. The band for this ruling says the reported quantity is an artifact of the file — that is not my phrasing, it is yours, one sentence earlier than mine.

IoT/IoT Network-Flow Intrusion Detection: CIC-ToN-IoT

C I C-ToN-I o T, four million eight hundred forty-seven thousand four hundred ninety-nine C I C Flow Meter flow records by sixty-eight numeric features from a single I I o T testbed, positive rate zero point five six one nine, pulled from a third-party Kaggle re-featurisation labelled V two.

HEADLINE

LightGBM, held-out ROC-AUC zero point nine nine six seven five one, accuracy zero point nine nine one zero five six, three-fold cross-validated zero point nine nine six two plus or minus zero point zero zero zero two.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9.5

/10 · self B

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 3 critical (FATAL)
- 18 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

The scoping in this notebook is exemplary and I want that said plainly. The abstract tells you that none of ToN underscore I o T's seven sensor-telemetry tables and none of its operating-system logs are read, and that nothing here is therefore evidence about telemetry-based detection — that is a restriction most authors would quietly omit. The clip caveat is written out in full before section ten, including the admission that no cell prints it. And the conclusion volunteers a defect that weakens its own audit: that the duplicate and overlap rates are upper bounds polluted by duplication the pipeline itself manufactured. Very few papers hand the reader a stick to beat them with.

EXAMINER RULING

The audit's central mitigation — that its duplicate and overlap rates are upper bounds because the pipeline's own clip manufactured the duplicates — is stated four times and measured zero times, and everything this notebook asks you to forgive rests on it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Two of every three alerts wrong, at best

EVIDENCE

Cell eighteen prints: operational false-positive rate at zero point five equals zero point zero one eight three, nine thousand seven hundred twenty-three benign flows flagged out of five hundred thirty thousand nine hundred sixty-seven. Now the base rate, and I will state the assumption out loud so nobody mistakes it for a measurement. Assume a network carrying one million flows a day at one percent attack prevalence. That is ten thousand attacks and nine hundred ninety thousand benign flows. Nine hundred ninety thousand times zero point zero one eight three is eighteen thousand one hundred seventeen false alarms. Even if this detector caught every single attack — and section thirteen says it catches sixty point two percent of man-in-the-middle and eighty-four point four percent of ransomware — you get twenty-eight thousand one hundred seventeen alerts of which ten thousand are real. Precision at best is ten thousand over twenty-eight thousand one hundred seventeen, which is zero point three five six. Roughly two of every three alerts are wrong, on a perfect detector.

SCIENTIFIC CONSEQUENCE

The corpus is fifty-six percent attacks. That prior is what makes zero point zero one eight three look small, and no production I o T network resembles it. The notebook never states a deployment prior and never performs this multiplication, so the reader is handed an operating point with no way to transfer it. Section two cites Sommer and Paxson, two thousand ten, a paper about why closed-world scores rarely survive deployment with false-positive cost among its central reasons, and nothing below that citation answers it.

REQUIRED REPAIR

An explicit cost matrix, a stated deployment base rate, a threshold chosen to minimise expected cost, and alerts per analyst per day at that threshold. A grep of all twenty-seven cells returns zero for a cost matrix and zero for threshold selection.

The mitigation is stated four times, measured zero

EVIDENCE

Section three, verbatim: the C I C Flow Meter Idle Mean, Idle Std, Idle Max and Idle Min fields are recorded on a scale that runs past the one times ten to the fifteen bound; that is a property of the source parquet, not something a cell below prints; Idle Max is the one column the clip alone removes. Section thirteen repeats it and draws the conclusion: the exact-duplicate rate zero point three zero seven, the thirty-one percent of rows the de-duplication drops, and the zero point one one four train-test overlap are therefore upper bounds that include duplication the pipeline manufactured. And then: no decomposition into manufactured versus native duplication is quoted, because no cell here measures the pre-clip rate. Cell eleven is the line in question: X equals X dot clip of minus one e fifteen and one e fifteen.

SCIENTIFIC CONSEQUENCE

This is a load-bearing claim carrying an entire defence. It converts a thirty point seven percent duplicate rate and an eleven point four percent contamination — both bad numbers — into upper bounds that the reader is asked to discount by an unknown amount. And it is unmeasured, by the notebook's own admission. The measurement is one line: compute the duplicate rate on the raw parquet before the clip, and print both. The author identified the confound, wrote three paragraphs about it, and did not run the three-second comparison that would settle it.

REQUIRED REPAIR

Compute and print the pre-clip exact-duplicate rate and pre-clip train-test overlap on the raw parquet, report the decomposition into manufactured and native duplication, and replace the flat clip with per-column scaling or a log transform so the Idle fields survive.

DEFENSIBLE REPLY

Run the decomposition you asked for. Without the clip my exact-duplicate rate is 0.127, not 0.307, and my train-test contamination is 0.008, not 0.114 - fifty-nine percent of one and ninety-three percent of the other are my own clip. My caveat is not an excuse I hid behind; it understates the size of the artifact, and the native figure grades A, not B. You are right that I did not measure it. What you cannot say is that everything I ask you to forgive rests on an unverified claim - it rests on a claim that verifies larger than I said.

ORDERED CORRECTION

Retitle from 'The mitigation is stated four times, measured zero' to 'The decomposition is three lines and it was left for someone else to run', and rewrite the verdict sentence, which currently sets the band on the refuted half. Supply the numbers so the charge lands where it belongs - on the omission, not on the credibility of the caveat: Idle Max is 100 percent above 1e15 and is exactly the column the clip removes (68 features with the clip, 69 without); Idle Mean and Idle Min 90.71 percent; Idle Std 7.47 percent; duplicate rate 0.3066 to 0.1265; contamination 0.1143 to 0.008; grade B to A. Keep the 'replace the flat clip with per-column scaling' clause - it is the fix and it changes the reported grade.

3

MAJOR

UNTESTED

DOWNGRADED

THRESHOLD FIXED AT 0.5, NO COST MODEL

ORAL TOP 5

One ransomware flow in six goes past

EVIDENCE

Cell eighteen prints: worst per-family recalls, m i t m zero point six zero two, d d o s zero point eight three three, ransomware zero point eight four four, d o s zero point nine, backdoor zero point nine nine five, scanning zero point nine nine eight. One minus zero point six zero two is zero point three nine eight — two in five man-in-the-middle flows missed. One minus zero point eight four four is zero point one five six — roughly one ransomware flow in six missed. The threshold that produced all of it is the constant in pred equals p b greater than zero point five. A grep of all twenty-seven cells returns zero for threshold selection and zero for a cost matrix; across all thirty-six notebooks in the series both counts are zero.

SCIENTIFIC CONSEQUENCE

Ransomware and man-in-the-middle are the two families in this list where a miss is not recoverable, and they are the two worst numbers. Nobody chose the point at which those misses happen; a library default did. For a detector with costs this asymmetric the operating point is the result, and this notebook reports an arbitrary one. This is also why the operational-admissibility axis scores two out of five here and two out of five on the sibling notebooks regardless of how their false-positive numbers compare: the axis grades whether an operating point, a cost model and an alert budget exist, not whether the number is flattering.

REQUIRED REPAIR

A stated cost matrix in which a missed ransomware flow carries an explicit cost in units of false alarms, a threshold selected to minimise expected cost, and curves of ransomware and man-in-the-middle recall against alert volume across the threshold range.

DEFENSIBLE REPLY

You have charged me twice for the constant 0.5. And if you want the per-family numbers taken seriously, ask about their denominators: ddos is two hundred and two rows in the whole corpus and dos is one hundred and forty-five, so two of the six recalls you quote rest on about fifty and about thirty-six held-out flows.

ORDERED CORRECTION

Delete as a separate filing and append to fault 2: 'and curves of ransomware and man-in-the-middle recall against alert volume across the threshold range'. Before quoting the six recalls anywhere, attach the denominators: mitm 517 corpus-wide (about 129 held out), ddos 202 (about 51), dos 145 (about 36), ransomware 4,790 (about 1,198).

Eleven percent of the test set is training data

EVIDENCE

Cell twenty prints an exact-duplicate row rate of zero point three zero seven and a train-test exact-row contamination of zero point one one four, graded B. Cell sixteen calls train underscore test underscore split with test size zero point two five, stratify on y, and no group argument. Section three says: one testbed, so cross-distribution generalization is untested. Section thirteen says: the split is random, not temporal or entity-grouped, every number above therefore measures in-distribution separability only. Across all thirty-six notebooks in this series, GroupKFold and GroupShuffleSplit appear zero times.

SCIENTIFIC CONSEQUENCE

Thirty point seven percent of this corpus is duplicated rows and eleven point four percent of the test set is literally training data — and that is before a single near-duplicate is counted. Under an i.i.d. split over a fixed-role testbed, the remaining rows are still generated by the same handful of devices running the same handful of tools. Zero point nine nine six seven five one measures how well the model recognises its neighbours. The notebook names this defect in two separate sections and corrects it in none.

REQUIRED REPAIR

GroupShuffleSplit or GroupKFold keyed on the source device or capture session, with the drop in held-out AUC reported as the headline rather than as a scope note.

DEFENSIBLE REPLY

Eleven percent of my test set duplicates my training set on the matrix I built, and I built the duplicates: without the clip the figure is 0.008. A grouped split would not touch it; deleting one line would. And you are asking me to group on a source device or a capture session in a file that has seventy-eight CICFlowMeter statistics and one attack-name column. There is no IP, no port, no timestamp and no flow identifier anywhere in it.

ORDERED CORRECTION

Re-mechanise to two separate, smaller charges. (a) 'The 0.114 is self-inflicted: it falls to 0.008 without the clip. The pipeline manufactures the leakage it then grades itself B for' - which is fault 1's finding, so consider merging. (b) 'The mirror was chosen without a group key. Moustafa's original ToN_IoT network CSVs carry src_ip, dst_ip and ts; dhoogla's V2 re-featurisation strips all three, so the entity-grouped experiment was foreclosed at the download line, not at the split line.' That is CORRECTION 3 Tier 2 and it is a defensible CONTESTED charge about mirror selection.

Four point eight million loaded, one hundred twenty thousand used

EVIDENCE

Cell eleven contains assert len of d f is at least one million and loads four million eight hundred forty-seven thousand four hundred ninety-nine rows. Cell sixteen holds out one million two hundred eleven thousand eight hundred seventy-five and subsamples the training pool to one hundred and twenty thousand. Subtract: four million eight hundred forty-seven thousand four hundred ninety-nine minus one million two hundred eleven thousand eight hundred seventy-five leaves three million six hundred thirty-five thousand six hundred twenty-four in the pool, of which one hundred and twenty thousand are used. Three million five hundred fifteen thousand six hundred twenty-four rows were read into memory and shown to no model. Two point five percent of the corpus reaches the learner. There is a second tell in the same cell: N R O W S is set to one million five hundred thousand with the comment per-file read cap, memory bound, and then the loader calls p d dot read underscore parquet, which does not accept it. The declared memory bound is decorative.

SCIENTIFIC CONSEQUENCE

The justification for discarding ninety-seven percent of the training pool is the code comment claiming these learners saturate long before then on this data. A grep of all twenty-seven cells for learning underscore curve returns zero. No plot, no table and no printed number anywhere shows AUC as a function of training-set size. The subtitle sells greater-than-or-equal-to one million records; the experiment is one hundred and twenty thousand, and the number four million eight hundred forty-seven thousand four hundred ninety-nine describes the download.

REQUIRED REPAIR

A learning curve to saturation across at least five training-set sizes on a log scale, or the effective n stated in the abstract in place of the corpus size, and a read cap that either functions or is deleted.

DEFENSIBLE REPLY

The read cap is dead and you caught it - it is a comment that describes nothing, and it should be deleted. It also costs nothing: the file is read whole, which is what I want. On the subsample, my cross-validation fits on twenty-six thousand seven hundred rows for 0.9962 against 0.996751 at a hundred and twenty thousand, and my held-out set is one point two million rows.

ORDERED CORRECTION

Split the two. Keep the dead-NROWS finding as its own crisp **CONTESTED** line - it is a clean verified code defect and the only one of its kind in the four packs. Retitle the subsampling half to 'Saturation is asserted in a comment

and never plotted' and concede the two in-notebook points.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

7

MAJOR

MEASURED

DOWNGRADED

MODEL RANKING WITHOUT A SIGNIFICANCE TEST

The winner is the less accurate model

EVIDENCE

Cell sixteen prints LightGBM at accuracy zero point nine nine one zero five six and ROC-AUC zero point nine nine six seven five one; XGBoost at accuracy zero point nine nine one three nine zero and ROC-AUC zero point nine nine six six seven three. Subtract the AUCs: seventy-eight millionths. Subtract the accuracies the other way: XGBoost leads by three hundred thirty-four millionths. Cell twenty-four reports the three-fold spread as plus or minus zero point zero zero zero two, which is more than twice the margin that chose the winner. A grep of all twenty-seven cells for McNemar returns zero; across the series it is zero of thirty-six.

SCIENTIFIC CONSEQUENCE

The two learners disagree about which one is better depending on the metric, and the notebook resolves the disagreement by hard-coding a sort on ROC-AUC. That is not a result; it is a tie-break rule. Every downstream number — the confusion matrix, the false-positive rate of zero point zero one eight three, the per-family recalls including ransomware at zero point eight four four, the whole ablation table — is computed on the model that seventy-eight millionths selected.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two learners, with the conclusion stating that they are indistinguishable if the test says so, and the results section reported for both if they are.

DEFENSIBLE REPLY

I sorted a table by ROC-AUC and took the top row; I never claimed LightGBM beats XGBoost. And your yardstick compares an unpaired fold spread on a forty-thousand-row subsample against a paired difference on the same one-point-two-million-row test set, where the two models' errors are nearly perfectly correlated. That is why the paired test you ask for is the right one, and why the fold spread is not the comparison.

ORDERED CORRECTION

Keep the title - 'the winner is the less accurate model' is a good and true line - but drop the fold-spread comparison and reduce to: section 13 writes 'Winning learner: LightGBM' unqualified where the two learners are indistinguishable, and XGBoost is ahead on accuracy by more than four times the AUC margin. Align severity with nb06 fault 9, which files this at CONTESTED.

You plotted the diagnostic and declined to read it

EVIDENCE

Cell eighteen imports precision underscore recall underscore curve, computes prec and rec, and plots them on the third axis under the title Precision-Recall. A grep of all twenty-seven cells for average underscore precision returns zero. Across all thirty-six notebooks in this series a precision-recall curve is plotted thirty-six times and its area computed zero times. Cell sixteen ranks by ROC-AUC alone.

SCIENTIFIC CONSEQUENCE

Precision is the quantity an analyst experiences, and it is the quantity the base-rate arithmetic makes decisive here. The area under the drawn curve is one function call from the line that drew it. The notebook plotted the right diagnostic and then declined to read it.

REQUIRED REPAIR

average underscore precision underscore score with a bootstrap confidence interval reported beside ROC-AUC, and precision quoted at a fixed recall such as zero point nine.

DEFENSIBLE REPLY

Attacks are fifty-six percent of my corpus. The curve you want integrated already has the majority class on the positive axis, so its no-skill line is 0.562 and its area is close to uninformative. And on my numbers the false-positive rate is more sensitive to a false positive than precision is, not less - a false positive is one part in five hundred and thirty-one thousand of the rate and one part in six hundred and eighty-one thousand of the precision. The class you should ask about is benign.

ORDERED CORRECTION

Replace the REQUIRED with: 'A precision-recall panel computed with BENIGN as the positive class, with average_precision_score and a bootstrap interval, and benign precision quoted at a fixed benign recall - the rare-class operating point, which no panel in this notebook establishes.' Do not ask for average_precision on the attack class at a 0.5619 positive rate.

Six recalls, three decimals, no denominators

EVIDENCE

Cell eighteen prints: m i t m zero point six zero two, d d o s zero point eight three three, ransomware zero point eight four four, d o s zero point nine, backdoor zero point nine nine five, scanning zero point nine nine eight. Not one count appears. The only inference available about any denominator is that it was at least five, because the code contains if c n t is less than five, continue. A grep of all twenty-seven cells for bootstrap returns zero and for confidence interval returns zero; across the series both are zero of thirty-six. The held-out AUC of zero point nine nine six seven five one likewise carries no interval.

SCIENTIFIC CONSEQUENCE

Section thirteen elevates m i t m at zero point six zero two to the place where detection is thinnest, which is an operational claim about a quantity with an unknown and possibly small denominator. And d o s printed as zero point nine flat, against neighbours carrying three decimals, is the visual signature of a small denominator — nine out of ten looks exactly like that. The reader cannot tell.

REQUIRED REPAIR

Print every family with its held-out positive count, attach a Wilson or Clopper-Pearson interval to every recall, and give a bootstrap interval on the held-out AUC.

The cross-validation validates a different model

EVIDENCE

Cell sixteen fixes every hyperparameter and varies none: RandomForest sixty estimators, XGBoost eighty estimators at max depth six, LightGBM eighty estimators, LogisticRegression max iterations three hundred. A grep of all twenty-seven cells for any hyperparameter search returns zero; across the series it is zero of thirty-six. Cell twenty-four sets `c v X to X t r dot i l o c` up to forty thousand, so three folds over forty thousand rows means each fit sees roughly twenty-six thousand six hundred sixty-seven rows. The headline model was fitted on one hundred and twenty thousand — and the corpus holds four point eight million.

SCIENTIFIC CONSEQUENCE

The plus or minus zero point zero zero zero two quoted as evidence of stability is the fold spread of a model trained on about a fifth of the reported one, computed on the same data from which the winner was selected. It bounds a different estimator than the one in the abstract. With no search at all, nothing establishes that any of the four learners was fairly configured before the ranking was drawn.

REQUIRED REPAIR

Nested cross-validation — an inner loop that selects hyperparameters and an outer loop that estimates performance — on the training set that produces the headline.

DEFENSIBLE REPLY

There is no search to nest. Four fixed configurations. If you want the cross-validation to validate what I reported, tell me to run it at a hundred and twenty thousand rows.

ORDERED CORRECTION

Replace the REQUIRED with 'run the cross-validation at the headline's $n = 120,000$ '. Add the selection leak the filing misses, which is the stronger version: `best_name` is chosen by ROC-AUC on `Xte` and every reported diagnostic is computed on that same `Xte`, with no separate validation split.

One draw, six decimals, a seventy-eight-millionth margin

EVIDENCE

Cell twenty-four prints seed equals zero. Every split uses random underscore state equals RANDOM underscore STATE. The eight-seed stability check that exists for this notebook was run by the external praxis probe, not by any cell, and it probed the single-feature audit — zero point eight one one zero to zero point eight one seven zero on Bwd underscore I A T underscore Min — not the zero point nine nine six seven five one and not the ranking.

SCIENTIFIC CONSEQUENCE

Seventy-eight millionths chose which model produced every number in section nine. A margin that small is a property of one draw, not of the learners, and it is reported to six decimal places as though the sixth digit were real.

REQUIRED REPAIR

At least ten seeds through the full pipeline, reporting mean and standard deviation and how often each learner wins.

DEFENSIBLE REPLY

The three-fold cross-validation reports plus or minus zero point zero zero zero two and the external eight-seed probe found the audit statistics stable, so the headline's leading decimals are not plausibly seed-driven even though the ranking may be.

The exclusion set is the empty set

EVIDENCE

Cell eleven reads: D R O P equals list of the set containing LABEL, FAMILY, y and family, unioned with s e t of the empty list. That empty list is the entire hand-written exclusion set. The only other guard is i d l i k e equals the object-dtype columns whose distinct count exceeds zero point five times the length of X. Two conditions, both required. On four million eight hundred forty-seven thousand four hundred ninety-nine rows a column must be string-typed and carry more than two million four hundred twenty-three thousand seven hundred forty-nine distinct values to be removed — and the provenance table says the features are sixty-eight numeric flow statistics. The filter cannot fire. Meanwhile the single strongest column is Bwd underscore I A T underscore Min at AUC zero point eight one three two, the highest single-feature figure of the four notebooks I hold, and it is a backward inter-arrival timing statistic.

SCIENTIFIC CONSEQUENCE

A minimum backward inter-arrival time records how fast the traffic generator's send loop ran. An attacker changes it by inserting delay, at no cost and no loss of capability. That such a column alone reaches zero point eight one three two is a construct-validity finding about the apparatus, and no cell partitions the sixty-eight features by provenance and refits. Contrast the sibling notebook two positions earlier in this series, on Edge-I I o T set, which names twenty-four columns to drop and explains that in a testbed capture the attacker and victim addresses are fixed per role. Same author, same week; here the list is empty.

REQUIRED REPAIR

Enumerate and print the sixty-eight retained feature names, hand-audit them for apparatus and timing coupling, drop the coupled block, refit, and report the AUC on behaviour-only features beside the headline.

A security detector never meets an adversary

EVIDENCE

A grep of all twenty-seven cells returns zero hits for adversarial, zero for evasion, and zero for any perturbation budget; across all thirty-six notebooks the counts are identical. Section two cites Sommer and Paxson, two thousand ten, a paper about why closed-world machine-learning scores rarely survive deployment, and no experiment below that citation responds to it.

SCIENTIFIC CONSEQUENCE

The strongest single column is a minimum inter-arrival time and the model's ranked top features per the external probe are Fwd underscore Packets underscore Length underscore Total and Bwd underscore Header underscore Length. Those are packet counts, packet sizes and timing — the three things an attacker controls most cheaply. A ransomware operator who pads and paces traffic changes all of them. Reporting zero point nine nine six seven five one with no evasion experiment describes an adversary who does not know a detector exists.

REQUIRED REPAIR

An evasion experiment under a stated perturbation budget — bounded padding and inter-packet delay applied to the attack class — with worst-case per-family recall reported alongside the clean figure.

DEFENSIBLE REPLY

Define the budget over what an attacker can do to traffic and recompute the aggregates. Perturbing sixty-eight CICFlowMeter statistics independently produces a row no flow could have produced, and the recall that falls out describes my feature space, not an adversary.

ORDERED CORRECTION

Require the budget be defined over realisable traffic transformations with aggregates recomputed. Note the charge is filed identically against all thirty-six notebooks.

Redundant separability diagnosed, never measured

EVIDENCE

Cell twenty-two shows the AUC moving from zero point nine nine six seven five one to zero point nine nine six seven four zero when the strongest single feature is dropped — a loss of eleven millionths — and to zero point nine nine five zero eight zero when thirty-one percent of rows are removed. The external probe drops the model's own top two features and the AUC holds at zero point nine nine six five. Section eleven's preamble writes the right reading in advance: if it barely moves, common on simulated corpora, that is not vindication, it means the classes are separable by many redundant features because the attack and benign distributions barely overlap. Section thirteen repeats it. No cell measures overlap, divergence or nearest-neighbour margin.

SCIENTIFIC CONSEQUENCE

The claim that the attack and benign distributions barely overlap is a testable statement about the joint distribution and it is the real finding this corpus supports. The notebook writes it three times and measures it never, and instead prints a five-nines figure in the abstract. That is disclosure standing in for measurement, and it leaves the reader unable to tell whether C I C-ToN-I o T is a hard problem or a well-separated capture.

REQUIRED REPAIR

Nearest-neighbour distance from each attack row to the nearest benign row and back, or a two-sample divergence estimate, reported as a distribution, with the barely-overlapping claim either supported or withdrawn.

DEFENSIBLE REPLY

The thirty-one percent you cite is not the corpus's duplication, it is mine: without the clip it is twelve point six percent. So the ablation row you are reading is the wrong experiment, which is the same finding as your fault one - and it is my finding, written in section thirteen.

ORDERED CORRECTION

Merge into fault 1 or restrict to the single unmeasured sentence in section 13. Mark the top-two-drop figure (0.9965) as external-probe output. Do not cite the 31 percent de-duplication row as evidence about the corpus - it is 12.6 percent natively.

The importance chart cannot support a claim

EVIDENCE

Cell eighteen reads: if hasattr best comma feature underscore importances underscore, then imp equals best dot feature underscore importances underscore, plotted as the top twelve. That is LightGBM's default split-count importance. A grep of all twenty-seven cells for permutation underscore importance returns zero, and for SHAP returns zero — the four apparent hits are the string dot shape.

SCIENTIFIC CONSEQUENCE

Split-count importance rewards columns with many distinct values, and on a corpus of sixty-eight continuous flow statistics — inter-arrival times, byte totals, header lengths — that bias is severe and systematic. A chart built on it cannot be read as an account of what the model learned, and it is the only explanation artifact the notebook produces.

REQUIRED REPAIR

Permutation importance computed on the held-out split, or SHAP values, with the impurity chart either replaced or labelled as a training-set diagnostic only.

DEFENSIBLE REPLY

No claim in section thirteen rests on the importance chart — the ablation drops the single-feature audit's winner, Bwd underscore I A T underscore Min, not the chart's leader — so the biased estimator informs a figure and nothing else.

Booij is cited and then ignored

EVIDENCE

Section two, reference two: Booij, Chiscop, Meeuwissen, Moustafa and den Hartog, two thousand twenty-two, ToN underscore I o T colon The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in I o T Network Intrusion Data Sets, I E E E Internet of Things Journal. The abstract then states that what is loaded is a C I C Flow Meter reprocessing of the network captures. No cell reconciles the family names, checks the attack-type taxonomy against Moustafa's, or compares the C I C Flow Meter feature set against the original ToN underscore I o T network features. The per-family recall list prints six lower-case names — m i t m, d d o s, d o s, ransomware, backdoor, scanning — with no mapping back to the source taxonomy.

SCIENTIFIC CONSEQUENCE

I will stand only on what this notebook prints. Reference two, from its own bibliography, is titled The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in I o T Network Intrusion Data Sets. Whatever else that paper argues, its title names the exact hazard the notebook then walks into: it adopts a third party's re-featurisation of the captures, prints six lower-case family names, and provides no mapping from those names back to the taxonomy of the corpus it credits. A citation in the literature review is not an application in the experiment, and here the cited title describes the choice the notebook made immediately afterwards.

REQUIRED REPAIR

Reconcile the C I C Flow Meter feature set and the attack-type taxonomy against Moustafa's original ToN underscore I o T network tables, print the mapping, and report per-family recall under the reconciled taxonomy.

DEFENSIBLE REPLY

My abstract states in the first three sentences that what I load is a CICFlowMeter re-featurisation of the network captures and that every feature below is a CICFlowMeter statistic, and it names what is not read - seven telemetry tables and the OS logs. The feature half of your charge is answered before section one. The taxonomy half is fair and is a piece of work I did not do.

ORDERED CORRECTION

Drop the feature-set half, which the abstract answers, and keep only the taxonomy reconciliation. Strengthen with the composition fact: the family names are printed lower-case with no mapping, and one of them - xss at 2,057,122 rows - is 75.5 percent of the positive class, which is a taxonomy question the reconciliation would have surfaced.

Credited to one party, downloaded from another, re-featurised by a third

EVIDENCE

The section-three provenance table gives Source as Kaggle d h o o g l a slash c i c t o n i o t, in parentheses C I C-ToN-I o T-V two. The abstract and reference one credit Moustafa, two thousand twenty-one. Those are different parties, and V two is a designation the notebook never explains — no cell states what distinguishes it from V one or from the C I C's own release. Cell eleven downloads whatever that account currently hosts and asserts only a one-million-row floor. No hash, no manifest, no comparison against an authoritative artifact appears anywhere.

SCIENTIFIC CONSEQUENCE

Every number here is a property of a file, and the notebook cannot establish which file. A V two designation on a mirror most often means someone cleaned, filtered or de-duplicated it — and the duplicate rate of zero point three zero seven and the overlap of zero point one one four are precisely the statistics such an operation changes. The notebook's central defence already concerns manufactured duplication; a re-processing step it never characterises is a second source of exactly that.

REQUIRED REPAIR

State what V two changes relative to the published artifact, compute a checksum of the download, and compare it against the authoritative distribution, printing the result.

DEFENSIBLE REPLY

The loader reads four million eight hundred forty-seven thousand four hundred ninety-nine rows by sixty-eight features and the external probe independently re-read the identical shape and positive rate, so whatever V two is, it is stable and reproducible across independent runs.

The transfer notebook is in the same directory

EVIDENCE

Section three, verbatim: one testbed, so cross-distribution generalization is untested. Section six lists the same as a scope limit and section thirteen repeats it. The same directory contains notebook thirty-six, titled cross-distribution transfer, alongside three further flow-corpus notebooks by the same author.

SCIENTIFIC CONSEQUENCE

The excuse of unavailability is unavailable. A cross-distribution transfer notebook already exists in this codebase, written by this author, for this course. And the nearest transfer target needs no bespoke feature alignment at all: notebook seven loads C I C-I D S two thousand seventeen, which is extracted by the same C I C Flow Meter tool and therefore speaks the same feature vocabulary as this corpus. The question was named as untested three times and not run.

REQUIRED REPAIR

Train on C I C-ToN-I o T and test on a second I o T flow corpus, and the reverse, reporting both directions and the drop from the in-distribution figure.

DEFENSIBLE REPLY

Notebook thirty-six's own loader comment says the shared forty-three-field NetFlow version two schema is what makes its comparison possible. My columns are Bwd IAT Min and Init Fwd Win Bytes and Fwd Seg Size Min. It has none of them. You cannot run it on my file.

ORDERED CORRECTION

Re-mechanise, and it gets STRONGER rather than weaker here - unlike nb06, this notebook does have schema-adjacent siblings. Replace the nb36 premise with: 'The series contains five other CICFlowMeter corpora - nb02, nb03 and nb04 on CIC-IDS2018, nb07 on CIC-IDS2017 and nb26 on CIC-DDoS2019 - whose feature names overlap this one substantially. A CICFlowMeter-to-CICFlowMeter transfer is available inside the author's own series and no transfer of any kind was attempted.' That is the claim the closing already gestures at with 'four sibling notebooks with an identical feature contract'; make it the mechanism instead of nb36.

Atlas proof

docs/course/security-ml-notebooks/nb08-cic-ton-iot.ipynb · Cells 1, 2, 4, and 25;
zero-based manifest cell IDs: 1=65a8460b, 2=a1c3ec4f, 4=7bbb911d, 25=fb542e9c

A letter grading the pipeline, quoted as grading the data

EVIDENCE

Cell twenty computes the grade from constants: for contamination, F at zero point five, D at zero point three, C at zero point one five, B at zero point zero five, otherwise A. The observed value is zero point one one four, so B. Section thirteen then concedes: all of it is computed on the preprocessed matrix X, never on the raw parquet, and that matters here; grade B grades this pipeline's split, not C I C-ToN-I o T itself. The heading on the output nevertheless reads data trust grade colon B.

SCIENTIFIC CONSEQUENCE

The label says data trust and the notebook says it does not grade the data. That gap is the whole objection. A letter awarded on an arbitrary constant, applied to a statistic the author states is an upper bound polluted by a pipeline artifact of unmeasured size, carries no information about the corpus. The correct move — measure the pre-clip rate and grade that — is one line away and was left as an exercise.

REQUIRED REPAIR

Report contamination with a binomial confidence interval, computed on the pre-clip matrix, and drop the letter.

DEFENSIBLE REPLY

The notebook explicitly says the grade applies to this pipeline's split and not to C I C-ToN-I o T, so a reader who reaches section thirteen is not misled — the objection is to the output label, not to any claim the author actually makes.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 1,211,875 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. And then the stick is never used. You identified the wrong guard, named the right one — per-column scaling or a log transform — wrote three paragraphs about which columns it corrupts, and left the run as an exercise on the grounds that the printed numbers must remain the numbers this pipeline produced. Reproducing a defective pipeline exactly is not a defence of its output. You had four point eight million rows, four sibling notebooks with an identical feature contract, and a transfer notebook in the same directory; you trained on one hundred and twenty thousand rows from one testbed and shipped a detector that misses two of every five man-in-the-middle flows.

Network Intrusion Detection on UNSW-NB15

UNSW-NB15, the 2015 corpus built to replace KDD99 — described in its own abstract as, quote, a hybrid of real benign traffic and synthesized attacks, end quote — pulled here from a third-party Kaggle re-upload as four headerless CSV parts, two million five hundred and forty thousand and forty-seven rows, attack rate zero point one two six five.

HEADLINE

LightGBM, held-out ROC-AUC zero point nine nine nine six seven one, accuracy zero point nine nine two six eight seven, with a three-fold cross-validated ROC-AUC of zero point nine nine nine five plus or minus zero point zero zero zero one.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self D

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 2 Honesty / scope

- 23 live faults
- 6 critical (FATAL)
- 17 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me start with what is genuinely good here, because there is a lot of it. You printed a majority-class baseline of zero point eight seven three five inside the ranking table, so nobody can mistake a high accuracy for a result. You found your own strongest single feature, you named it, and then you did the harder thing and re-ran the model without it. You graded your own data a D and published the D. And the de-duplication ablation moved the number — nought point nine nine nine six seven one down to nought point nine nine eight seven three one after twenty-one percent of rows came out — and you reported the movement instead of burying it. That is more self-scrutiny than most of the published UNSW-NB15 literature contains. The praxis pack gave you nine and a half out of ten for the quality of that audit, and on the pack's terms that was the right score. I am grading a different thing: whether the claim survives.

EXAMINER RULING

You measured how separable one file is from itself, you wrote down every reason that is not the same as detecting an intrusion, and then you printed zero point nine nine nine six seven one in the abstract anyway — so I am rejecting the claim, not the craft.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The clock was in your hand and you dropped it

EVIDENCE

Cell eleven types out all forty-nine column names by hand, because the raw files ship headerless. Two of those names are Stime and Ltime — flow start time and flow last time. A handful of lines later the DROP list removes both. So the notebook proves the timestamps exist in the act of throwing them away. Cell sixteen then calls `train_test_split` with test size zero point two five and stratify on y. That is a shuffle. Your own section-thirteen conclusion says it: quote, the split is random, not temporal or entity-grouped, end quote.

SCIENTIFIC CONSEQUENCE

An intrusion detector is deployed forward in time. It is trained on what happened up to Tuesday and it must work on Wednesday. A shuffled split lets a flow recorded at ten oh three in the morning predict a flow recorded at ten oh four, from the same burst of the same synthesised attack. Nothing in zero point nine nine nine six seven one distinguishes learning what an exploit looks like from learning what that afternoon looked like. And because you dropped the timestamps rather than never having them, you cannot say the experiment was unavailable. It was available, it was four lines, and it was declined.

REQUIRED REPAIR

Sort the frame on Stime. Train on the earlier fraction, test on the later fraction, hold the ratio at seventy-five twenty-five so the comparison is like for like. Print both numbers side by side and let the drop be the finding. If there is no drop, you have learned something important and you can say so.

Your contamination number is a floor, not a measurement

EVIDENCE

Follow the variable. In cell sixteen, Xtr starts as the seventy-five percent training half and is then reassigned — quote, if len of Xtr is greater than one hundred and twenty thousand, Xtr comma underscore comma ytr comma underscore equals train_test_split, end quote. From that line onward Xtr is the subsample. Cell twenty then builds its lookup table from exactly that variable: underscore t r keys equals set of map of tuple over Xtr. Now do the arithmetic from your own printed line. You materialised two million five hundred and forty thousand and forty-seven rows and held out six hundred and thirty-five thousand and twelve. That leaves a training half of one million nine hundred and five thousand and thirty-five rows. You trained on, and therefore checked against, one hundred and twenty thousand of them. Six point three percent.

SCIENTIFIC CONSEQUENCE

Contamination is monotone in the size of the reference set. Search a bigger training set and you find more matches, never fewer. So zero point one three three is a floor. And you know the true corpus-wide duplication is far higher, because the line directly above it prints an exact-duplicate row rate of zero point two zero six. Then the floor was fed into a hard cutoff — the code reads, C if contamination is at or above zero point one five, B otherwise — and zero point one three three landed a B by seventeen thousandths. That unearned B is then spent in the conclusion, which says, quote, train-test overlap separately scores B, so overlap is not the issue here, end quote. A lower bound was converted into a letter and the letter was used to dismiss the concern the lower bound could not rule out.

REQUIRED REPAIR

Build the key set from the full training half before the subsample line, or state in the printed output that the figure is overlap against the rows actually fitted and is a lower bound on split leakage. Then re-grade. And measure it on the whole held-out set, not the first fifty thousand rows of six hundred and thirty-five thousand and twelve — that window is seven point nine percent of the test set.

ct_state_ttl is a fact about the machines

EVIDENCE

Cell twenty prints, quote, best single-feature AUC equals zero point nine nine one eight, feature colon ct_state_ttl, end quote. Read that name against the column list you typed out by hand in cell eleven. That list contains state, and it contains sttl and dttl — source and destination time to live — as columns in their own right. So ct_state_ttl is, by its name and by the company it keeps in your own list, a count keyed on connection state and on TTL. And TTL is set by the sending operating system and decremented once per router in the path. Your own abstract describes the corpus as, quote, a hybrid of real benign traffic and synthesized attacks, end quote — so by construction the two classes have different origins, and TTL is exactly the kind of thing that differs between origins.

SCIENTIFIC CONSEQUENCE

Zero point nine nine one eight from one column means a single number that is not about attacker behaviour separates ninety-nine percent of the ordering. An attacker sets TTL with one socket option. Change it and the feature is gone at zero cost, no capability required, no detection risk. So the notebook has not measured intrusion detection; it has measured how distinguishable two sets of source machines were on a testbed in 2015. That is a fact about the lab, and the lab has been decommissioned for a decade.

REQUIRED REPAIR

Partition the forty-one columns into behaviour features and provenance features — TTL-derived, window-size-derived, host-count-derived — and report the held-out AUC on the behaviour set alone. That number, whatever it is, is the one that means something. Report it in the abstract instead of zero point nine nine nine six seven one.

Source addresses removed, then never grouped on

EVIDENCE

The DROP list in cell eleven reads, in order: Label, attack_cat, y, family, srcip, dstip, sport, dsport, Stime, Ltime. The comment on that line says ids slash time slash labels. So you identified srcip as an identifier and you removed it. Then cell sixteen splits at random on the row index. No GroupKFold, no GroupShuffleSplit, no groups argument appears anywhere in this notebook.

SCIENTIFIC CONSEQUENCE

Network flows are not independent draws. One infected host emits thousands of near-identical flows in a burst. A random split scatters that burst across train and test, so a model can score a test flow by recognising its siblings from training rather than by recognising an exploit. That is the mechanism behind an exact-duplicate rate of zero point two zero six, and it is why the number does not transfer. You removed the grouping variable, which proves you knew it was a grouping variable, and then split as though the rows were independent.

REQUIRED REPAIR

Keep srcip out of the features and pass it as groups to GroupShuffleSplit. Every flow from one source address lands on one side. Report the held-out AUC under that split next to zero point nine nine nine six seven one. That is roughly twenty lines and it is the experiment your own conclusion nominates.

The winner is separated by fourteen millionths

EVIDENCE

The cell-sixteen table reads LightGBM at zero point nine nine nine six seven one and XGBoost at zero point nine nine nine six five seven. The gap is fourteen millionths — zero point zero zero zero zero one four. Cell twenty-four then prints, quote, LightGBM three-fold CV ROC-AUC equals zero point nine nine nine five plus or minus zero point zero zero zero one, end quote. Your own uncertainty figure is one ten-thousandth. That is seven times the gap you used to pick a winner. Random Forest sits at zero point nine nine nine six zero four, inside the same band.

SCIENTIFIC CONSEQUENCE

Every downstream number in this notebook — the confusion matrix, the false-positive rate of zero point zero zero three seven, the per-family recalls, the entire ablation table — is computed on best, and best was selected by a comparison your own standard deviation cannot resolve. Re-run with a different seed and a different learner plausibly wins, and the whole results section changes model. The phrase best model, printed in cell sixteen, is not a finding. It is the output of an argmax over noise.

REQUIRED REPAIR

McNemar's test on the paired predictions of LightGBM and XGBoost on the same held-out rows, with the p-value printed. Or bootstrap the held-out AUC a thousand times per model and print the overlap of the intervals. If they overlap, print a tie and carry all three tree models forward.

DEFENSIBLE REPLY

You cannot use my cross-validation standard deviation as the yardstick here and then tell me in fault nineteen that the same number is a different estimand which must never be quoted as the uncertainty of the held-out score — pick one. And on the metric you say I should have used, LightGBM's accuracy of 0.992687 against XGBoost's 0.992411 on 635,012 held-out rows is 4,644 errors against 4,819, a margin of 175 net errors; under the most adversarial possible assumption, that the two models' error sets are disjoint so the discordant count is the full 9,463, McNemar gives chi-square 3.24 and $p = 0.072$, and under any realistic overlap it is p below 0.01. My winner leads on both columns.

ORDERED CORRECTION

Delete the sentence 'That is seven times the gap you used to pick a winner' and the appeal to the CV standard deviation entirely — it is incompatible with fault 19. Re-mechanise to what survives: no paired test and no bootstrap interval on the held-out AUC was computed, so the ranking is asserted rather than tested. Note also that on accuracy the ordering is unchanged (LightGBM 0.992687, RandomForest 0.992458, XGBoost 0.992411), which is not true on nb10 or nb12 and must not be carried across from them.

Atlas proof

docs/course/security-ml-notebooks/nb09-unsw-nb15.ipynb · Cell 17, comparison table; zero-based manifest cell IDs: 17=6609c824

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You cited the official split and did not run it

EVIDENCE

The section-two literature list credits Moustafa and Slay two thousand and sixteen for, quote, statistical analysis and the train slash test split, end quote. The reference list carries it at number two. The same section's caveat table then contains the row, quote, Modern DL on UNSW-NB15, in-distribution; official split has known overlap, end quote. So the notebook records that the official partition exists and that it is defective. It then uses neither the official partition nor a corrected one. Cell eleven concatenates UNSW-NB15 parts one through four and cell sixteen shuffles.

SCIENTIFIC CONSEQUENCE

Zero point nine nine nine six seven one cannot be placed beside any published UNSW-NB15 result. Not one number in the literature was computed on this split, because this split exists only in this notebook. A benchmark score that cannot be compared to the benchmark is not a benchmark score. And the excuse — that the official partition has known overlap — is an argument for reporting both, not for reporting neither.

REQUIRED REPAIR

Run the official Moustafa and Slay partition and print that AUC in the same table, labelled. Then, separately, run your grouped or temporal split. Three numbers, one table: literature-comparable, defensible, and the difference between them.

DEFENSIBLE REPLY

The official partition is not a split of my file: it is a separately engineered 45-column release of 175,341 plus 82,332 rows, a different feature set on a tenth of the data, which my own `assert len(df) >= 1_000_000` gate excludes as a primary corpus. I agree it belongs in the table as a labelled comparator. It cannot be the experiment.

ORDERED CORRECTION

Reword REQUIRED to make explicit that the official partition is an additional comparator on a different 45-feature schema, not a re-split of the 49-column raw files, and that its numbers are not directly comparable to the four-learner table. Also note that section 2's caveat table already states these are 'not measurements reproduced on this exact corpus', so the notebook does not claim literature comparability — the charge is that it should have earned it, not that it falsely asserted it.

Atlas proof

docs/course/security-ml-notebooks/nb09-unsw-nb15.ipynb · Cells 4, 12, and 17; literature review, four-file loader, and random split; zero-based manifest cell IDs: 4=11d18fae, 12=23ffbce0, 17=6609c824

Two and a half million in the title, one hundred and twenty thousand in the fit

EVIDENCE

The subtitle of the notebook reads, quote, two point five M records, via Kaggle, end quote. Cell sixteen prints, quote, materialized two million five hundred and forty thousand and forty-seven rows, trained on one hundred and twenty thousand, stratified subsample, end quote. One hundred and twenty thousand over two million five hundred and forty thousand and forty-seven is four point seven percent.

SCIENTIFIC CONSEQUENCE

The scale claim in the title is a claim about a download, not about an experiment, and a reader who takes the two point five million as the sample size of the result is reading the notebook exactly as it is written. Worse, no learning curve was drawn, so there is no evidence for the stated justification that these learners saturate long before then. Saturation was asserted, not shown. If the score is already saturated at one hundred and twenty thousand rows, that is itself the most interesting sentence in the notebook and it is unmeasured.

REQUIRED REPAIR

A learning curve: fit at ten thousand, thirty thousand, one hundred thousand, three hundred thousand, one million rows and plot held-out AUC against n . Either the curve is flat, which proves the saturation claim, or it is not, which invalidates the subsample. Then put the effective n — one hundred and twenty thousand — in the abstract next to the AUC. And credit where it is due: cell sixteen carries an explicit disclosure comment saying every headline below is a subsample number. That is the right instinct. It is in a code comment, and the title still says two point five million.

DEFENSIBLE REPLY

The disclosure is not buried in a comment: it prints in the output of cell sixteen, and section eight states in prose that every score is a subsample number. And the quantity you are attacking, the held-out ROC-AUC, is measured on 635,012 rows — the entire 25% test half, none of it subsampled. Only the fit is 120,000.

ORDERED CORRECTION

Delete 'It is in a code comment' — quote the printed line instead. Keep the residual, which is the real one and is unanswerable: no learning curve was drawn, so 'these learners saturate long before then' is asserted and not shown, and the effective training n does not appear in the abstract.

A D that sits eighteen ten-thousandths from a C

EVIDENCE

Cell twenty computes the sample as X dot sample of sixty thousand with random_state one — that is one draw, sixty thousand rows out of two million five hundred and forty thousand and forty-seven, two point four percent of the corpus. The grading line then reads: F if the AUC is at or above zero point nine nine nine, D if at or above zero point nine nine, C if at or above zero point nine five. The printed value is zero point nine nine one eight. It clears the D boundary by eighteen ten-thousandths.

SCIENTIFIC CONSEQUENCE

No interval was computed for that zero point nine nine one eight, so nobody — including the author — can say whether a second draw would land on the other side of zero point nine nine and turn the notebook's headline grade from D to C. A letter whose distance from its own boundary is smaller than its unreported sampling spread carries no information. And this letter is load-bearing: it is quoted in the abstract, in the conclusion, and in the course's summary of the notebook.

REQUIRED REPAIR

Bootstrap the single-feature AUC across at least two hundred resamples, print the mean and the ninety-five percent interval, and state the grade only if the whole interval falls inside one band. If it straddles a cutoff, print the interval and drop the letter.

DEFENSIBLE REPLY

Do the arithmetic CORRECTION 6 requires. The audit sample is 60,000 rows at a 12.65% attack rate, so 7,590 positives and 52,410 negatives; Hanley-McNeil gives a standard error of 0.00073 at an AUC of 0.9918, a 95% half-width of 0.0014, against a distance of 0.0018 to the D-C boundary. The interval does not cross. And because best_auc is a maximum over forty-one features, any selection bias pushes the estimate up and away from the cutoff, not toward it. My D is stable.

ORDERED CORRECTION

Re-mechanise to the durable form CORRECTION 6 prescribes: not 'this letter is unstable' — it is not — but 'hard cutoffs applied to a point estimate with no interval are not a measurement.' The method critique survives; the specific instability claim must be withdrawn or it hands the student a refutation with arithmetic attached.

You drew the right curve and refused to read it

EVIDENCE

Cell eighteen computes `prec` comma `rec` comma `underscore` equals `precision_recall_curve` of `yte` comma `pb`, and plots it as the third panel titled Precision-Recall. That is the whole use of it. There is no `average_precision_score` anywhere in this notebook. Nothing reads the area, nothing prints it, nothing ranks a model on it. Meanwhile the class balance from your own printed numbers: six hundred and thirty-five thousand and twelve held-out rows, of which five hundred and fifty-four thousand six hundred and ninety-one are benign — so eighty thousand three hundred and twenty-one are attacks.

SCIENTIFIC CONSEQUENCE

ROC-AUC is a ranking statistic that is insensitive to the absolute count of false positives when negatives dominate. It is precisely the metric that stays near one while an operator drowns. You put the metric that would have shown that on the page as a picture and then declined to compute its area. The correct criticism is not that the diagnostic is missing. It is that the diagnostic is present, was drawn, and was not read.

REQUIRED REPAIR

`average_precision_score` on the held-out probabilities, printed with a bootstrap interval, reported in the abstract beside or instead of the ROC-AUC. And precision at a fixed operationally meaningful recall — say ninety-five percent — printed as a number.

DEFENSIBLE REPLY

Read the panel you say I ignored together with the numbers I printed. Held-out is 635,012 rows with 80,321 attacks; I print 2,065 false positives and an accuracy of 0.992687, which fixes 2,578 false negatives, so precision at my operating point is 0.974 and recall is 0.968. The failure mode you describe — ROC near one while precision collapses — is not the failure mode of this corpus, and the confusion matrix panel prints the counts that show it.

ORDERED CORRECTION

Drop the rhetorical claim about drowning operators; on this corpus precision at the reported threshold is 0.974 and the counts to compute it are printed. Keep the narrow, true residual: average precision is never computed, so the area under the curve that was drawn is never reported, and no metric in the notebook carries an interval.

Six per-family recalls, not one positive count

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, Fuzzers zero point six three eight, Analysis zero point eight five eight, Shellcode zero point nine four three, Exploits zero point nine eight three, DoS zero point nine eight nine, Backdoors zero point nine nine three, end quote. The loop that produces them reads for fam comma cnt in fam_te where yte equals one dot value_counts, and cnt is used only in the guard if cnt is less than five, continue. The count is computed, tested, and thrown away unprinted.

SCIENTIFIC CONSEQUENCE

The conclusion nominates these recalls as, quote, the honest evidence, end quote, and then names Fuzzers at zero point six three eight as, quote, where detection is thinnest. Shellcode and Backdoors are small classes in this corpus, and a recall of zero point nine four three on a small denominator has an interval wide enough to overlap zero point six three eight. Without the counts, the ranking of the families by recall — which is the notebook's stated honest result — cannot be checked at all.

REQUIRED REPAIR

Print each family with its positive count and a Clopper-Pearson binomial interval on its recall. Then the sentence about which family is thinnest becomes a claim a reader can test instead of an ordering of six bare decimals.

DEFENSIBLE REPLY

You claim a recall of 0.943 on a small denominator has an interval wide enough to overlap 0.638. It does not. Shellcode is 1,511 rows in this corpus, about 378 in the held-out quarter; the exact Clopper-Pearson interval on 0.943 is 0.913 to 0.963, and Fuzzers on 6,062 positives is 0.626 to 0.650. The ordering the conclusion relies on does not dissolve — the smallest scored group, Worms, is not even in the printed six.

ORDERED CORRECTION

Withdraw the sentence beginning 'Shellcode and Backdoors are small classes in this corpus, and a recall of 0.943 on a small denominator has an interval wide enough to overlap 0.638.' Replace it with the true and still-strong version: no denominator and no interval is printed for any of the six, so a reader cannot check the ordering the conclusion calls 'the honest evidence' — which happens to hold on this corpus but is unverifiable from the page. On nb10 the same charge is devastating and should carry the FATAL instead.

Nobody chose zero point five

EVIDENCE

Cell eighteen reads pred equals pb greater than zero point five, cast to int. The printed line is, quote, operational FALSE-POSITIVE RATE at zero point five equals zero point zero zero three seven, two thousand and sixty-five benign flagged of five hundred and fifty-four thousand six hundred and ninety-one, end quote. The threshold appears in the label because it is the default, not because it was chosen. There is no cost matrix, no expected-cost minimisation and no threshold sweep in this notebook.

SCIENTIFIC CONSEQUENCE

In intrusion detection the cost of a miss and the cost of a false alarm differ by orders of magnitude, and they differ differently at three in the morning than at three in the afternoon. When costs are asymmetric the operating point is the result — it is what an analyst actually experiences. Reporting the recall on Fuzzers as zero point six three eight is reporting the recall on Fuzzers at an arbitrary threshold. Move it and that number moves. You have drawn a precision-recall curve that contains every alternative operating point and then reported one point on it, chosen by scikit-learn.

REQUIRED REPAIR

State a cost ratio — even a stated assumption, say fifty to one — build the two-by-two cost matrix, and select the threshold that minimises expected cost on a validation fold. Print the chosen threshold, print the FPR and per-family recalls at it, and show the sensitivity of the answer to the ratio.

DEFENSIBLE REPLY

The number in my abstract is a ROC-AUC. It is invariant to the threshold, so nothing you have said about 0.5 touches it. What you have correctly attacked is the operating-point block, and every line of it is printed with '@0.5' in the label precisely so no reader mistakes it for a chosen point.

ORDERED CORRECTION

Scope the charge to the operating-point block rather than to the headline, and merge the remedy with fault 12 — one cost-model-and-threshold-selection cell answers both.

The number you thresholded is not a probability

EVIDENCE

Cell eighteen takes best dot predict_proba of Xte, column one, and compares it to zero point five. The word calibration does not occur in this notebook. There is no CalibratedClassifierCV, no isotonic fit, no Platt scaling, no reliability plot.

SCIENTIFIC CONSEQUENCE

Boosted-tree outputs are systematically miscalibrated, and the direction of the miscalibration depends on the class prior — here zero point one two six five. So the zero point five cut is not the median of the posterior; it is an arbitrary point on a monotone but unknown transformation of it. Every count in the confusion matrix, the two thousand and sixty-five false positives, and every per-family recall inherit that. And the notebook never claims otherwise, which means the error is silent rather than disclosed.

REQUIRED REPAIR

Fit isotonic or Platt calibration on a held-out calibration fold, print a reliability diagram and a Brier score before and after, and only then apply a threshold.

DEFENSIBLE REPLY

Calibration is a monotone transformation. It leaves every ROC-AUC in my table bit-for-bit unchanged, it leaves the model ranking unchanged, and it leaves the precision-recall curve unchanged. The one thing it can move is the operating point — which is fault eleven's charge, and which threshold selection on a held-out fold fixes without any calibration at all. There is no quantity in this notebook that calibration would correct.

ORDERED CORRECTION

Either withdraw as a duplicate of fault 11 or re-mechanise narrowly: the notebook writes ``predict_proba(...) > 0.5``, which treats a boosted-tree score as a posterior probability, and that conceptual claim is unsupported even though no reported number depends on it.

Zero point three seven percent, multiplied out

EVIDENCE

Cell eighteen prints a false-positive rate of zero point zero zero three seven — two thousand and sixty-five benign flows flagged out of five hundred and fifty-four thousand six hundred and ninety-one. The literature section cites Sommer and Paxson two thousand and ten, the paper whose central argument is that the base rate is what defeats network anomaly detectors in practice. That citation is never applied to this notebook's own number.

SCIENTIFIC CONSEQUENCE

Let me do the arithmetic you owed, and I am going to state my assumption out loud so it can be checked. Take a mid-size enterprise seeing ten million benign flows a day — that figure is my assumption, it is not in your notebook. Ten million times zero point zero zero three seven is thirty-seven thousand false alarms per day. Divide across a three-analyst shift and that is over twelve thousand alerts each. The word operational in your printed line is doing work the number cannot support. You cite the paper that says this and you do not do the multiplication.

REQUIRED REPAIR

Print alerts per analyst per day at your chosen operating point, for a stated flow volume, in the results cell. Then state the FPR your alert budget can absorb and read the threshold that achieves it off the precision-recall curve you already drew.

DEFENSIBLE REPLY

You supplied the traffic volume yourself and told the room it was your assumption, which is fair — but it means the arithmetic you say I owed is arithmetic only you can do. And at my printed operating point precision is 0.974, so 97.4% of the alerts in your thirty-seven thousand would be true positives.

ORDERED CORRECTION

The stronger attack is available and should replace this one: Sommer & Paxson's argument is about the base rate of attacks, not about FPR times volume. This corpus carries a 12.65% attack rate — three to four orders of magnitude above any production network — so it is the base rate, not the false-positive rate, that makes 'operational' unsupportable, and the corrected precision at a realistic prior collapses even though the FPR does not move. That version is unanswerable and needs no invented volume.

Twenty point six percent is the part you could see

EVIDENCE

Cell twenty computes `dup_rate` as one minus `X dot drop_duplicates dot shape` over the length of `X` — exact rows only. Contamination is a set membership test on tuples of numpy round to six decimals — exact rows only. Printed: exact-duplicate row rate zero point two zero six, contamination zero point one three three. No nearest-neighbour distance, no near-duplicate threshold, no minimum-distance distribution across the split boundary anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A scan burst emits thousands of flows that differ in one byte count or one microsecond of duration. None of them are exact duplicates. All of them are the same event. Removing exact duplicates cost you twenty-one percent of rows and only zero point zero zero zero nine four of AUC — which does not show that duplication is unimportant, it shows that the exact-duplicate filter removed the wrong thing. The near-duplicates that carry the leakage survived the de-duplication ablation intact, which means the ablation cannot support the conclusion drawn from it.

REQUIRED REPAIR

Fit a nearest-neighbour index on the training rows, compute for each held-out row its distance to the nearest training row, and plot that distribution. Then re-run the de-duplication ablation with a distance threshold, not an equality test, and report the AUC at several thresholds.

Two labels on one row, and you kept the first

EVIDENCE

Cell twenty-two reads Xdd equals X dot drop_duplicates, then ydd equals y indexed by Xdd dot index. Pandas keeps first by default. The conclusion states the behaviour plainly: quote, the de-duplication variant keeps the first label when a feature vector appears twice, end quote. The disagreement rate itself is never computed and never printed.

SCIENTIFIC CONSEQUENCE

When one feature vector carries both label zero and label one, that is a hard bound on any classifier's achievable accuracy — the Bayes error of the corpus as featurised. It is arguably the single most informative statistic you could extract from a corpus with a twenty point six percent duplicate rate. Discarding it makes the de-duplicated AUC of zero point nine nine eight seven three one optimistic by an unmeasured amount, and it removes the one number that would let a reader judge whether zero point nine nine nine six seven one is even attainable in principle.

REQUIRED REPAIR

Group by the feature vector, count the groups whose labels are not unanimous, and print that fraction. That is your noise floor. Report it next to the headline. If it is near zero, say so and the ablation stands; if it is not, the ablation is measuring something else.

DEFENSIBLE REPLY

The quantity you call the single most informative statistic in the corpus is 706 rows out of 2,540,047 — 0.028 percent — spread over 453 non-unanimous groups. And it was bounded before either of us computed it: my classifier attains 0.992687 held-out accuracy, and empirical error is an upper bound on Bayes error, so the irreducible floor could not have exceeded 0.73% however the duplicates were resolved. It is worth printing. It cannot change a conclusion.

ORDERED CORRECTION

Withdraw 'arguably the single most informative statistic you could extract' and 'makes the de-duplicated AUC optimistic by an unmeasured amount' — the amount is measurable at 0.028% and is bounded above by the achieved error rate a priori. The residual is a legitimate hygiene ask: print the noise floor so the reader does not have to derive its bound.

Nine families named, six recalls printed

EVIDENCE

Cell two lists the task's classes: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Nine. Cell eighteen prints srt dot head of six — the six lowest. Generic, Reconnaissance and Worms do not appear in the printed dictionary, in the ledger, or in the conclusion. The same loop carries a silent guard: if cnt is less than five, continue. The number of families it skipped is never printed.

SCIENTIFIC CONSEQUENCE

The conclusion elevates per-family recall to, quote, the honest evidence, end quote, and then shows two-thirds of it. And because the skip count is never printed, absence from the printed list and failure to be scored look identical from the page — a reader cannot tell a family that did well from a family the guard removed. An honest lead metric that omits its own coverage is not auditable.

REQUIRED REPAIR

Print every family with its positive count and its recall, and print explicitly how many families the guard skipped and which. A family with fewer than five positives should be printed as unscorable, never omitted.

DEFENSIBLE REPLY

Look at the panel, not just the print statement. Because len(srt) is ten and the branch only truncates above twelve, `show` is the whole series: every scored family, Generic and Reconnaissance and Worms included, is drawn in 'Per-family recall (worst first)'. And the guard you say hides families skipped none — the smallest group, Worms, has 174 rows in the corpus and about 44 in the held-out quarter, well clear of five.

ORDERED CORRECTION

Strike 'do not appear in the printed dictionary, in the ledger, or in the conclusion' down to the printed dictionary and the ledger only, and strike 'a reader cannot tell a family that did well from a family the guard removed' — the figure distinguishes them. What survives: the printed text and the ledger show six of ten, the guard's skip count is never printed even though it happens to be zero, and the corpus's 'Backdoor'/'Backdoors' spelling split silently creates a spurious eleventh group that the notebook never reconciles.

A security detector with no adversary in it

EVIDENCE

There is no perturbation, no evasion budget, no adversarial example, no feature-space attack anywhere in the twenty-seven cells. The full evaluation is a stratified random split of a static file. Meanwhile cell twenty tells you the attack surface: one column, `ct_state_ttl`, alone achieves zero point nine nine one eight.

SCIENTIFIC CONSEQUENCE

The threat model of an intrusion detection system is a human being who will read this notebook. That person will observe that the model leans on a TTL-derived counter, will pin TTL to sixty-four, will pad packets and will jitter timing — all of it free, none of it requiring a new exploit. The worst case is what a security claim must report, and here the worst case is not merely unreported, it is uncomputed. The forty-one features contain no attack cost model, so nothing in this notebook distinguishes a signal an attacker cannot cheaply change from one they can.

REQUIRED REPAIR

State a perturbation budget in feature space — for instance, any generator-coupled column may be moved to any value the protocol permits — and report the worst-case held-out recall under that budget. That number, not zero point nine nine nine six seven one, is the security claim.

DEFENSIBLE REPLY

My conclusion scopes the claim in terms: 'every number above therefore measures in-distribution separability only.' That is not a disclaimer attached to a robustness claim; it is a statement that no robustness claim is made. You are asking me to report the worst case for a claim I did not enter.

ORDERED CORRECTION

Re-aim at the gap the scope sentence does not close: the notebook is titled 'Network Intrusion Detection', leads its abstract with 0.999671, and the scope sentence sits in section 13. A reader who stops at the abstract acquires an operational reading the notebook never retracts prominently. That is an editorial charge, not a missing-experiment charge, and it should not carry FATAL.

A biased ranking, plotted as an explanation

EVIDENCE

Cell eighteen reads, if `hasattr` best feature importances_, then `imp` equals best dot feature importances_ — the LightGBM built-in — and plots the top twelve under the title top importances slash absolute coefficient. There is no `permutation_importance` call and no SHAP anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The corpus is full of high-cardinality continuous columns — byte counts, jitter, load, inter-packet times — and impurity attribution systematically inflates exactly those relative to low-cardinality ones. So the ranking a student reads off that bar chart is partly an artifact of column cardinality rather than of predictive contribution. When the notebook's central question is which columns carry the score, using the biased estimator to answer it undercuts the answer.

REQUIRED REPAIR

Permutation importance computed on held-out data, or SHAP values, printed with the impurity ranking beside it so the divergence is visible.

DEFENSIBLE REPLY

My answer to 'which columns carry the score' is the single-feature ROC-AUC sweep in section ten — model-free, computed on the data, not on the model's split counts. The importance bar chart is a diagnostic picture and it is cited nowhere in the ledger, in the conclusion or in the ablation. You are attacking an estimator I did not use to make the claim.

ORDERED CORRECTION

Correct the technical detail too: LightGBM's `feature_importances_` defaults to `importance_type='split'`, a split count, not gain or impurity decrease. The cardinality-bias direction is right, the named quantity is not. Keep the charge as: an unlabelled, biased ranking is plotted under a title a student will read as an explanation.

The cross-validation runs on a third of the training subsample

EVIDENCE

Cell twenty-four reads `cvX` comma `cvy` equals `Xtr` dot `iloc` up to forty thousand, `ytr` up to forty thousand — and remember `Xtr` is already the one-hundred-and-twenty-thousand-row subsample. So the printed zero point nine nine nine five plus or minus zero point zero zero zero one is a three-fold estimate on forty thousand rows, two point one percent of the training half of one million nine hundred and five thousand and thirty-five. The four learners in cell sixteen are instantiated with fixed arguments — `n_estimators` eighty, `max_depth` six — and no search object appears in the notebook.

SCIENTIFIC CONSEQUENCE

Two separate failures compound. First, defaults were never tuned, so the model comparison is a comparison of four libraries' defaults, not of four learners. Second, the CV figure that appears in the abstract and in the validity ledger does not estimate the uncertainty of the headline — different sample size, different folds, different estimand — yet it is quoted as though it does, and it is the only dispersion figure in the notebook. Section four calls this cell reproducibility and robustness; it establishes neither.

REQUIRED REPAIR

Nested cross-validation: an inner loop that selects hyperparameters and an outer loop that estimates generalisation, both on the pipeline actually reported. If that is too slow at one hundred and twenty thousand rows, say so and shrink honestly — but do not report a forty-thousand-row three-fold spread as the robustness of a six-hundred-and-thirty-five-thousand-row held-out score.

One seed, one split, one number

EVIDENCE

Cell ten sets `RANDOM_STATE` equals zero with the comment, quote, single seed used everywhere, end quote. That seed drives the split in cell sixteen, the subsample, all four model fits, the ablation re-splits in cell twenty-two and the folds in cell twenty-four. Cell twenty draws its audit sample with `random_state` equals one — a second fixed seed, one draw. No loop over seeds appears anywhere.

SCIENTIFIC CONSEQUENCE

The winner was chosen on a gap of fourteen millionths, the trust grade sits eighteen ten-thousandths from its cutoff, and the contamination figure sits seventeen thousandths from its own. Every one of those margins is smaller than a plausible seed-to-seed shift, and not one of them has been probed across seeds. A comment saying single seed used everywhere is a statement of the problem, correctly identified, and then left in place.

REQUIRED REPAIR

Loop the entire reported pipeline over at least ten seeds. Print mean and standard deviation for the held-out AUC, the winning model name, the trust grade and the contamination rate. If the winner changes across seeds, that is the finding and it belongs in the abstract.

DEFENSIBLE REPLY

You assert that every one of those three margins is smaller than a plausible seed-to-seed shift. Two are not. The single-feature AUC of 0.9918 sits 2.46 standard errors above the 0.99 boundary on a 60,000-row draw. The contamination estimate of 0.133 on 50,000 held-out rows has a standard error of 0.0015 against a distance of 0.017 to its cutoff — eleven standard errors. Only the model-ranking margin is genuinely fragile, and that is fault four's charge, not a third instance.

ORDERED CORRECTION

Delete 'Every one of those margins is smaller than a plausible seed-to-seed shift' and keep only the model-ranking margin, which is the one that is. The durable residual is method, not letter: a single seed is used for the split, the subsample, all four fits, both ablation re-splits and the CV folds, and no reported quantity carries a seed-to-seed dispersion.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 635,012 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So here is the ruling. This notebook is an unusually honest description of an experiment that does not test its own hypothesis. You wrote, in your own conclusion, that the split is random and not entity-grouped, and that every number therefore measures in-distribution separability only — and that sentence is correct, and it is fatal, and having written it you printed zero point nine nine nine six seven one in the abstract anyway. At doctoral level a named limitation is not a defence. It is a confession that the experiment was understood, costed, and declined. Eight out of thirty. Reject. Come back with a temporal split, a source-address grouping, and the official partition, and the same four learners will give you a number I can actually argue with.

Botnet Detection on the CTU-13 Dataset

CTU-13 from the Stratosphere lab at CTU Prague — thirteen captures of real malware executed on real machines, mixed with real background traffic, read here as Argus flow records: one million five hundred and seventy-two thousand and fifty-six flows across nine features, botnet rate zero point one seven three six.

HEADLINE

XGBoost, held-out ROC-AUC zero point nine eight nine four zero five, accuracy zero point nine six four one eight two, with a three-fold cross-validated ROC-AUC of zero point nine eight six zero plus or minus zero point zero zero one zero.

DEFENSE

F

Selected-model estimand

DOCTORAL

10

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self D

D1 1 External validity	D2 1 Statistical inference	D3 3 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 3 Honesty / scope

- 23 live faults
- 9 critical (FATAL)
- 14 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be emphatic about what this notebook has that the simulated corpora do not. These are real malware binaries, executed on real machines, generating real command-and-control traffic against real background noise. Neris, Rbot, Virut, Menti, Sogou, Murlo and NSIS-dot-ay were not scripted by a traffic generator. That is a genuine construct advantage over notebook one and over the testbed corpora, and it shows in the honest numbers: your strongest single feature reaches only zero point eight two seven one, a long way from a shortcut, and your de-duplication ablation actually moved the headline — zero point nine eight nine four zero five down to zero point nine seven two seven nine six — and your conclusion responded by saying, quote, so the headline is inflated by repeated rows, and zero point nine seven two seven nine six is the honest number, end quote. That is a claim revised in the face of evidence. It is the single best moment in the thirty-six notebooks. Then you also told the room, unprompted, that your negative class contains unverified Background traffic and that the false-positive rate is therefore an upper bound on nuisance rather than a measured error. All of that is right. And none of it is the same as having run the experiment.

EXAMINER RULING

This is the best corpus in your set and the most instructive failure in it: you built the grouping variable with your own hands, printed it, correctly refused to let it be a feature, and then split at random across it anyway.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Per-family recall is per-capture recall

EVIDENCE

Cell eleven prints the map in the source: SCEN_MALWARE, one Neris, two Neris, three Rbot, four Rbot, five Virut, six Menti, seven Sogou, eight Murlo, nine Neris, ten Rbot, eleven Rbot, twelve NSIS-dot-ay, thirteen Virut. Family comes from that dictionary, not from the label string. Further down in that same cell the comment reads, verbatim, quote, scenario is capture identity, it maps one to one to the malware family, so it MUST NOT be a feature, end quote — capitals in the original — and scenario is duly placed in DROP alongside StartTime, SrcAddr, DstAddr, Sport and Dport. Then cell sixteen calls `train_test_split` with a random state and stratification on `y`. Not one capture is held out.

SCIENTIFIC CONSEQUENCE

You wrote MUST NOT be a feature in capital letters. That sentence is a statement that scenario is identity, and identity is exactly what a grouped split exists to control. Because the split is random within captures, every held-out flow comes from a capture the model has already trained on, from the same infected host talking to the same command-and-control server in the same hour. The model can score a test flow by recognising its neighbours. And a botnet detector's entire job is the case you never test: a capture it has never seen. Worse, Neris spans scenarios one, two and nine, Rbot spans three, four, ten and eleven, and Virut spans five and thirteen — so those three families are structurally guaranteed to appear on both sides of any split you draw. Zero point nine eight nine four zero five measures recognition of thirteen specific captures, not detection of seven malware families.

REQUIRED REPAIR

Leave-one-capture-out. Thirteen folds, each holding out one scenario entirely, reporting AUC per held-out capture and the mean across folds. You already have the scenario integer on every row — you computed it with a regular expression in cell eleven. Passing it as the `groups` argument to `GroupKFold` is one line.

Atlas proof

docs/course/security-ml-notebooks/nb10-ctu13-botnet.ipynb · cell 11 SCEN_MALWARE mapping and DROP list; zero-based manifest cell IDs: 11=9e5b34e1

Thirty-five point three percent is the floor

EVIDENCE

Trace the variable. Cell sixteen reassigns Xtr to the stratified subsample when the training half exceeds one hundred and twenty thousand rows. Cell twenty then builds its lookup set from that same Xtr: underscore t r keys equals set of map of tuple over Xtr. Now the arithmetic from your own printed lines. You materialised one million five hundred and seventy-two thousand and fifty-six flows and held out three hundred and ninety-three thousand and fourteen. That leaves one million one hundred and seventy-nine thousand and forty-two training rows, and you checked against one hundred and twenty thousand of them. Ten point two percent. The printed result is, quote, TRAIN slash TEST exact-row contamination equals zero point three five three, end quote — thirty-five point three percent of the tested window.

SCIENTIFIC CONSEQUENCE

Contamination is monotone in the size of the set you search. Search ten times more training rows and you find more matches, never fewer. So thirty-five point three percent is a floor, and the corpus-wide duplicate rate of zero point four one nine tells you where the ceiling probably sits. The number is then compared to a hard cutoff — D at zero point three, F at zero point five — and it landed a D. On the full training half it may well land an F, which is your own code's label for, quote, heavy contamination, end quote. The grade that drives the whole trust assessment was computed on a tenth of the evidence and rounded in the notebook's favour by construction.

REQUIRED REPAIR

Build the key set from the full training half before the subsample reassignment, and measure it across the whole held-out set rather than the first fifty thousand rows of three hundred and ninety-three thousand and fourteen. Print both figures — overlap with the rows actually fitted, and overlap with the split — and grade on the second.

The negative class is not known benign

EVIDENCE

Cell eleven sets y to one wherever the label string contains the word Botnet, case-insensitive; everything else becomes zero. Section three quotes the Stratosphere Laboratory's own definition of Background as, quote, all the rest of traffic that we don't know what it is for sure, end quote. And then cell eighteen prints, quote, operational FALSE-POSITIVE RATE at zero point five equals zero point zero one six two, five thousand two hundred and fifty benign flagged of three hundred and twenty-four thousand seven hundred and seventy-two, end quote — the word benign, in the printed output, describing traffic the corpus authors will not certify. The Normal-versus-Background split is never counted and never printed.

SCIENTIFIC CONSEQUENCE

Two things break at once. Every flow the model flags may be a genuine botnet flow sitting unlabelled in Background, so zero point zero one six two is not an error rate and calling it operational is not supportable. And on the training side, some class-zero rows are plausibly undetected botnet flows, which means the model is fitted against noisy negatives that pull it toward whatever the hidden malicious traffic looks like. The majority baseline of zero point eight two six four inherits the same defect. You state both mechanisms, in the notebook, with precision — and then you publish the number that depends on them not being true.

REQUIRED REPAIR

Restrict the negative class to rows whose label string says Normal, print how many rows that leaves, and recompute the false-positive rate against verified-clean traffic only. Report the Background count separately as an unlabelled pool. If the verified-Normal class is too small to support the estimate, that is the finding and it belongs in the abstract.

Atlas proof

docs/course/security-ml-notebooks/nb10-ctu13-botnet.ipynb; docs/course/security-ml-notebooks/assignments/evidence/SEAS8414-praxis-findings-pack.html · cells 4, 11, 18 and 25; findings-pack nb10 criterion 2; zero-based manifest cell IDs: 4=ea138814, 11=9e5b34e1, 18=6a32c729, 25=ba58b0cb

Thirty-two millionths apart, and the runner-up is more accurate

EVIDENCE

The cell-sixteen table: XGBoost roc-auc zero point nine eight nine four zero five, accuracy zero point nine six four one eight two. Random Forest roc-auc zero point nine eight nine three seven three, accuracy zero point nine six eight one nine seven. The AUC gap is thirty-two millionths — zero point zero zero zero zero three two. Cell twenty-four prints the spread: zero point nine eight six zero plus or minus zero point zero zero one zero. One thousandth. That is thirty-one times the gap. And on accuracy the ordering reverses — Random Forest wins by zero point zero zero four zero one five, which is a hundred and twenty-five times the AUC gap that decided the contest.

SCIENTIFIC CONSEQUENCE

The selection rule is a single line — sort by roc_auc, take the first row — and it resolved a difference far below its own measured noise. Everything downstream is then computed on best: the confusion matrix, the false-positive rate of zero point zero one six two, all six per-family recalls, and every row of the ablation table including the honest number zero point nine seven two seven nine six. Change the seed and Random Forest plausibly wins, and the entire results and ablation sections are recomputed on a different model. Best model, as printed, is an argmax over noise with a hundred and twenty-five to one evidence pointing the other way on the other metric.

REQUIRED REPAIR

McNemar's test on the paired predictions of XGBoost and Random Forest, p-value printed. Or bootstrap the held-out AUC and print the two intervals. If they overlap, declare a tie, say so in the table, and carry both models through the results section.

Atlas proof

docs/course/security-ml-notebooks/nb10-ctu13-botnet.ipynb · cell 16 output; zero-based manifest cell IDs: 16=409ee65a

Your own table names the harder test, and you ran the easier one

EVIDENCE

Section two's caveat table contains one row that decides this: quote, Garcia et al. two thousand and fourteen, behavioural methods — evaluated per-scenario; cross-scenario is much harder, end quote. That is your sentence, in your table, citing your reference one. Cell eleven then concatenates every binetflow file it can find into one frame, and cell sixteen shuffles it.

SCIENTIFIC CONSEQUENCE

You did not merely fail to run the published protocol. You wrote down what the published protocol is, wrote down that it is harder, and then ran the other one and reported the number without a comparison. Zero point nine eight nine four zero five cannot be placed beside any figure in Garcia et al., because no figure in Garcia et al. was computed on a pooled random split of all thirteen captures. And the reason it cannot be placed beside them is the reason it is high.

REQUIRED REPAIR

Run the per-scenario protocol from Garcia et al. two thousand and fourteen and print those numbers in the same table as yours, labelled. Then print the cross-scenario matrix your own caveat row says is much harder. The difference between the two is the result this notebook should be reporting.

DEFENSIBLE REPLY

My caveat table is prefaced, in the notebook, with the sentence that these are drawn from the wider literature and are not measurements reproduced on this corpus. I claim no comparability to Garcia et al. and I say so. And the experiment you are asking for here is the same experiment you asked for in fault one and again in fault six — one grouped split answers all three.

ORDERED CORRECTION

Merge with fault 6, or keep only the distinct residual — that no literature-comparable number is printed anywhere — and drop the second FATAL count for the same missing experiment.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Thirteen distributions, and you pooled them

EVIDENCE

Cell eleven walks every binetflow file, tags each with its scenario integer, and concatenates. The scenario integer survives into the frame. The conclusion names the missing test explicitly: quote, whether flow features generalize across scenarios, end quote. And it is not run. There is no transfer matrix, no leave-one-out fold, no per-scenario score anywhere in the twenty-seven cells.

SCIENTIFIC CONSEQUENCE

This series contains notebook thirty-six, which builds exactly this kind of matrix on four NetFlow corpora and prints it — so the capability exists in the same repository, written by the same author, and was not applied here where the data structure hands it to you for free. Seven malware families across thirteen captures is a transfer experiment already assembled. You named it as the honest question and left the answer uncomputed while a template for computing it sat in the same directory.

REQUIRED REPAIR

A thirteen-by-thirteen or seven-by-seven transfer matrix: fit on one scenario or family, score on each of the others, print the diagonal and the off-diagonal means separately. Notebook thirty-six's cell already does this — the code is thirty lines and it is in your own repository.

DEFENSIBLE REPLY

Notebook thirty-six transfers among four NetFlow v2 corpora precisely because they share a 43-field schema; CTU-13 has nine Argus columns and shares that schema with none of them, so nothing in notebook thirty-six runs here. What you are asking for is the grouped split from fault one, reported as a matrix.

ORDERED CORRECTION

Keep the mechanism as written — it is one of the few E2 faults in the set that respects CORRECTION 4 — but merge it with fault 1 and count the missing per-capture evaluation once, reported as a 13x13 matrix, rather than three times.

StartTime removed, never ordered on

EVIDENCE

The DROP list in cell eleven reads Label, y, family, scenario, StartTime, SrcAddr, DstAddr, Sport, Dport, under the comment ids slash time slash labels. StartTime is therefore present on every row and is discarded. Cell sixteen shuffles. The conclusion confirms: quote, the split is random, not temporal or entity-grouped, end quote.

SCIENTIFIC CONSEQUENCE

Botnet flows are time-clustered — your own loader comment says so, in the sentence explaining why the read cap matters: quote, botnet flows are a time-clustered minority within each capture, end quote. A shuffled split therefore scatters a single burst of command-and-control beacons across train and test. The model learns the burst, not the beacon. Detection in the field means detecting the next burst, and no number in this notebook estimates that.

REQUIRED REPAIR

Within each capture, order by StartTime and split at a time boundary — train on the first portion of the infection, test on the later portion. Report the drop from zero point nine eight nine four zero five. You wrote the sentence about time-clustering yourself; act on it.

One point five million in the subtitle, one hundred and twenty thousand in the fit

EVIDENCE

The subtitle reads, quote, model comparison plus validity audit on CTU-13 real botnet captures, greater than or equal to one M flows, end quote. Cell sixteen prints, quote, materialized one million five hundred and seventy-two thousand and fifty-six rows, trained on one hundred and twenty thousand, stratified subsample, end quote. One hundred and twenty thousand over one million five hundred and seventy-two thousand and fifty-six is seven point six percent. And there is an assert in cell eleven enforcing the one-million floor — the floor gate is on the download, not on the experiment.

SCIENTIFIC CONSEQUENCE

The one-million claim is a claim about a file. The assert that enforces it checks the length of the dataframe, not the size of the fit. A reader who takes the scale of the corpus as the scale of the evidence is reading the notebook as written. And no learning curve was drawn, so the justification in the code comment — that these learners saturate long before then — is asserted and not shown.

REQUIRED REPAIR

A learning curve from ten thousand to one million rows with held-out AUC on the vertical axis. Then put the effective n of one hundred and twenty thousand into the abstract next to zero point nine eight nine four zero five, where the one point five million currently sits.

DEFENSIBLE REPLY

The line 'trained on 120,000 (stratified subsample)' prints in my output, and section eight says in prose that every score below is a subsample number. And the quantity you are attacking is measured on 393,014 held-out rows — the entire test half.

ORDERED CORRECTION

Keep only the residual, which is unanswerable: no learning curve exists, so 'these learners saturate long before then' is asserted and never measured, and the effective fitted n is absent from the abstract.

Seventeen point three six percent is a setting, not a base rate

EVIDENCE

Cell eleven sets `READ_CAP` to one million five hundred thousand rows scanned per scenario, then for each capture takes every botnet flow and, quote, non equals d where not isbot dot sample of min of not isbot sum comma one hundred thousand, end quote. The printed line calls the result what it is: quote, botnet rate, bounded sample, zero point one seven three six, end quote. And the code comment states the true figure: quote, a bounded sample, not the full population, it also means the botnet rate below is a bounded-sample rate, not the natural roughly one to two percent base rate, end quote.

SCIENTIFIC CONSEQUENCE

The majority-class baseline of zero point eight two six four — the bar the entire results section is judged against — is therefore a loader argument, not a property of botnet traffic. At the natural rate the notebook itself names, one to two percent, the trivial baseline is above ninety-eight percent, and an accuracy of zero point nine six four one eight two would be below it. Every accuracy figure in the cell-sixteen table would flip its meaning. You wrote the sentence that says this, in a comment, and then published the table that depends on nobody applying it.

REQUIRED REPAIR

Re-weight the held-out set to the natural base rate, or subsample positives to reach it, and print the majority baseline and the model accuracies at that rate. State the resulting baseline in the abstract next to the score, because at one to two percent positives the accuracy column stops meaning anything at all.

A letter with no interval, fifty-three thousandths from the next band

EVIDENCE

Cell twenty grades contamination with the line, F if at or above zero point five, D if at or above zero point three, C if at or above zero point one five. The printed value is zero point three five three, measured on the first fifty thousand of three hundred and ninety-three thousand and fourteen held-out rows — twelve point seven percent of the test set — and against ten point two percent of the training rows. It cleared the D boundary by fifty-three thousandths and sits one hundred and forty-seven thousandths below F. No interval is printed for it.

SCIENTIFIC CONSEQUENCE

The letter D is quoted in the abstract, in the ledger and in every summary of this notebook, and it is a point estimate from one window with no dispersion attached. Because the reference set is a tenth of the training half — see fault two — the true value is above zero point three five three, and the distance to F is smaller than the distance the correction is likely to move it. A grade whose band cannot be resolved is a decoration, not a measurement, and it is being used as one.

REQUIRED REPAIR

Bootstrap the contamination estimate across resampled held-out windows, print the mean and interval, and assign a letter only when the entire interval falls in one band. Otherwise print the interval and drop the letter.

DEFENSIBLE REPLY

You say the distance to F is smaller than the distance the correction will move it. Measure both. The reference-set correction takes 0.353 to 0.4478 — nine and a half points — and the distance to F is fourteen and seven tenths. And the sampling spread you say is unreported has a standard error of 0.0021 on fifty thousand rows, which is twenty-five standard errors from the nearest cutoff. My D is not close to anything.

ORDERED CORRECTION

Re-mechanise to the durable form CORRECTION 6 prescribes: not 'this band is unresolved' — it is resolved, in both directions — but 'a letter assigned by a hard cutoff to a point estimate with no interval is not a measurement.' As written this fault hands the student two numbers that refute it.

You plotted the precision-recall curve and never read it

EVIDENCE

Cell eighteen calls `precision_recall_curve` on the held-out probabilities and plots it as panel three, titled Precision-Recall. That is its entire use. `average_precision_score` appears nowhere in this notebook. Nothing computes the area, prints it, or ranks a model with it. Meanwhile the ranking in cell sixteen sorts on `roc_auc`, and the abstract leads with zero point nine eight nine four zero five.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to the absolute number of false positives when negatives dominate — and in deployment they dominate fifty to a hundred to one, by the notebook's own estimate of the natural rate. So ROC-AUC is precisely the statistic that stays near one while precision collapses. The correct criticism is not that the diagnostic is absent. It is on the page, as a picture, and it was not read.

REQUIRED REPAIR

Print `average_precision_score` with a bootstrap interval and lead with it. Add precision at a fixed recall — ninety percent, say — as a number. And recompute both after re-weighting the negative class to the natural base rate.

DEFENSIBLE REPLY

At the balance I report, precision is 0.919 and recall 0.871, and the counts to compute both are printed on the confusion-matrix panel. The scenario you describe, ROC near one while precision collapses, does not occur at 17.36% positives — it occurs at the natural rate, which is the criticism you already made in fault nine.

ORDERED CORRECTION

Merge the base-rate half into fault 9 and keep only the narrow, true residual here: the area under the curve that was drawn is never computed, and no metric in the notebook carries an interval.

Sogou at zero point one eight eight, with no denominator

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, Sogou zero point one eight eight, NSIS-dot-ay zero point two eight four, Neris zero point seven four seven, Murlo zero point eight six six, Virut zero point eight eight five, Menti zero point nine eight four, end quote. The loop reads for fam comma cnt in value_counts, and cnt is used only in the guard if cnt is less than five, continue. It is never printed. Nowhere in the notebook is a confidence interval computed on anything.

SCIENTIFIC CONSEQUENCE

The conclusion says of Sogou, quote, the weakest group sits at zero point one eight eight, so the model misses most of it. That gap, not the aggregate score, is the operationally important result, end quote. I agree with that sentence — which is exactly why the number underneath it needs a denominator. Sogou is one capture, scenario seven, and if the held-out positives number in the low hundreds then the interval around zero point one eight eight is wide enough to overlap NSIS-dot-ay at zero point two eight four, and the ordering the conclusion relies on dissolves. Your most important result is the one with the least support and the only one with no error bar.

REQUIRED REPAIR

Print each family's held-out positive count beside its recall, with a Clopper-Pearson binomial interval. Then the sentence about which family is weakest becomes checkable.

The operating point is a library default

EVIDENCE

Cell eighteen reads pred equals pb greater than zero point five. The printed label carries the threshold — false-positive rate at zero point five equals zero point zero one six two — because it is the default, not because it was selected. There is no threshold sweep, no expected-cost calculation, no cost matrix in these twenty-seven cells.

SCIENTIFIC CONSEQUENCE

The cost of missing a botnet beacon and the cost of paging an analyst about a DNS lookup are not the same cost, and under asymmetric costs the operating point is the result. Sogou's recall of zero point one eight eight — the number the conclusion calls the operationally important result — is Sogou's recall at zero point five. Lower the threshold and it rises, at some price in false positives you have not costed. You have already drawn a curve containing every alternative operating point and reported the one scikit-learn handed you.

REQUIRED REPAIR

State a cost ratio explicitly, build the two-by-two cost matrix, choose the threshold that minimises expected cost on a validation fold, and print it. Then report the FPR and all per-family recalls at that threshold, plus how the choice shifts if the ratio doubles.

DEFENSIBLE REPLY

The number in my abstract is a ROC-AUC and it is invariant to the threshold, so nothing about 0.5 touches it. The block you are attacking prints '@0.5' in the label of every line precisely so no reader mistakes it for a chosen point.

ORDERED CORRECTION

Scope to the operating-point block, and note that Sogou's 0.188 is the example that actually bites — it is three of sixteen at 0.5, and moving the threshold on sixteen positives moves it in steps of 0.0625.

One point six two percent, multiplied out

EVIDENCE

Cell eighteen prints zero point zero one six two — five thousand two hundred and fifty flagged out of three hundred and twenty-four thousand seven hundred and seventy-two. Section two cites Sommer and Paxson two thousand and ten, the paper whose central claim is that base rates defeat network anomaly detection in practice, and lists it at reference three. The citation is never applied to this notebook's own figure.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic you owed, and I am stating my assumption aloud so it can be checked: take a network seeing ten million benign flows a day — that number is mine, not yours. Ten million times zero point zero one six two is one hundred and sixty-two thousand false alarms a day. And remember fault four: because Background is unverified, you cannot even say those hundred and sixty-two thousand are mistakes. So the word operational is attached to a rate that has neither been multiplied out nor validated against confirmed-clean traffic. You cite the paper that says this will happen and you do not do the multiplication.

REQUIRED REPAIR

Print alerts per analyst per day at your chosen operating point for a stated flow volume, in the results cell. Then state the alert budget the deployment can absorb and read the required threshold off the precision-recall curve you already drew.

DEFENSIBLE REPLY

You supplied the traffic volume yourself and said so. The criticism that survives without it is the one you already made in fault nine.

ORDERED CORRECTION

Merge into fault 9. Sommer & Paxson's argument is about the base rate of attacks, not about false-positive rate times volume, so fault 9's re-weighting demand is the correct application of the citation and this one is a weaker restatement.

Trained at seventeen percent, deployed at one

EVIDENCE

The four models in cell sixteen are instantiated with fixed constructor arguments and none of them carries `class_weight` or `scale_pos_weight` — the terms appear nowhere in the notebook. The loader's own comment names the mismatch: quote, not the natural roughly one to two percent base rate, end quote.

SCIENTIFIC CONSEQUENCE

A classifier fitted at seventeen point three six percent positives and cut at zero point five carries a decision boundary tuned to a prior that is roughly ten to seventeen times the deployment prior. Deployed, it over-predicts the positive class systematically, and the false-positive rate rises above zero point zero one six two rather than staying there. So the one figure this notebook labels operational is optimistic in the specific direction that matters, by an amount nothing in the notebook estimates.

REQUIRED REPAIR

Set `scale_pos_weight` from the target deployment prior, or correct the predicted log-odds by the log ratio of training prior to deployment prior, and report the false-positive rate under the corrected prior. Both are one line.

DEFENSIBLE REPLY

A false-positive rate is a rate over negatives. Change the prevalence and you change how many negatives arrive, not what fraction of them my model flags. My 0.0162 does not rise when the prior falls — my precision does, and that is a different number which I also did not report.

ORDERED CORRECTION

Replace 'the false-positive rate rises above 0.0162 rather than staying there' with the correct consequence: FPR is invariant to the prior at a fixed threshold; positive predictive value is what degrades, from 0.919 at the manufactured 17.36% to roughly 0.45 at the 1-2% rate the notebook names. The defect — no class weighting, no prior correction, a decision boundary fitted to a loader argument — survives intact and is worth keeping.

Forty-one point nine percent is only the identical ones

EVIDENCE

Cell twenty computes `dup_rate` as one minus `drop_duplicates` over length, and contamination as a set-membership test on tuples rounded to six decimals. Both are equality tests. The printed values are zero point four one nine and zero point three five three. There is no nearest-neighbour distance, no near-duplicate threshold, no distance distribution across the split boundary anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Command-and-control beacons are near-identical by design — the same query, the same size, the same interval, differing by microseconds of duration and a byte or two of payload length. None of those are exact duplicates. All of them are the same behaviour. So the de-duplication ablation, which removed forty-two percent of rows and cost you zero point zero one six six zero nine of AUC, removed only the flows that happened to round identically and left the beacons that carry the real leakage sitting on both sides of the split. That means the honest number you correctly identified — zero point nine seven two seven nine six — is itself still optimistic.

REQUIRED REPAIR

Fit a nearest-neighbour index on the training rows and plot, for each held-out row, the distance to its nearest training neighbour. Then re-run the de-duplication ablation with a distance threshold rather than an equality test, at several thresholds, and report the curve.

Seven families recovered, six recalls printed

EVIDENCE

Cell eleven prints, quote, botnet families recovered, Menti, Murlo, NSIS-dot-ay, Neris, Rbot, Sogou, Virut, end quote. That is seven. Cell eighteen prints srt dot head of six, and the six that appear are Sogou, NSIS-dot-ay, Neris, Murlo, Virut and Menti. Rbot is not in the printed dictionary, not in the ledger and not in the conclusion. The same loop also carries the guard if cnt is less than five, continue, whose skip count is never printed.

SCIENTIFIC CONSEQUENCE

Rbot spans four of the thirteen captures — scenarios three, four, ten and eleven — which makes it the family most heavily represented on both sides of a random split and therefore the one whose recall the split most inflates. It is the single most informative family in the corpus for the question this notebook asks, and it is the one that does not appear. And because the skip count is never printed, a reader cannot distinguish a family that scored well from a family that fell below the count guard.

REQUIRED REPAIR

Print all seven families with their held-out positive counts and recalls, and print explicitly how many the count guard skipped and which. A family below the guard should be printed as unscorable, never omitted.

DEFENSIBLE REPLY

Cell eleven prints the seven families it recovered by name, and the recall panel draws seven bars, Rbot included — the branch that truncates only fires above twelve. A reader who compares those two counts knows the guard skipped nothing.

ORDERED CORRECTION

Strike 'a reader cannot distinguish a family that scored well from a family that fell below the count guard' — the seven-in, seven-plotted comparison distinguishes them. Keep the strong part, which is correct and specific: Rbot spans four of thirteen captures, is therefore the family a random split inflates most, and is the one family the printed text and the ledger omit.

Nine columns of Argus bookkeeping

EVIDENCE

Cell eleven prints, quote, loaded one million five hundred and seventy-two thousand and fifty-six flows x nine features, end quote. Cell twenty prints, quote, best single-feature AUC equals zero point eight two seven one, feature colon State, end quote. State is the Argus connection-state string — the flags-and-status bookkeeping of the connection — integer-encoded by LabelEncoder in cell eleven and fed to the model as a number.

SCIENTIFIC CONSEQUENCE

An integer-encoded categorical imposes an ordering that does not exist: the encoder assigns State strings to integers alphabetically, so a tree learner is splitting on alphabetical position. Whatever zero point eight two seven one is measuring, it is not a monotone property of connection state. And State is under the attacker's control at essentially zero cost — a botnet that completes its handshakes differently changes the column outright. Nine columns is a very thin construct on which to claim botnet detection, and the single strongest of them is the one an adversary can most cheaply move.

REQUIRED REPAIR

One-hot encode State rather than integer-encoding it, and report the AUC on behaviour-only features — duration, byte and packet counts — with the protocol bookkeeping columns removed. That number is the transferable claim.

DEFENSIBLE REPLY

My own audit grades that column A. The single-feature check exists to catch a near-sufficient feature and it did not fire, which is the correct outcome and I printed it.

ORDERED CORRECTION

Keep the construct-validity charge — nine columns of Argus accounting is a thin basis for a botnet-detection claim, and the ordinal encoding of `State` is a genuine modelling error — but drop any implication of a shortcut, since the printed single-feature grade is A. Note also that one-hot encoding would likely raise the single-feature figure, not lower it, so the remedy is not a mitigation.

Malware that adapts, evaluated as if it did not

EVIDENCE

There is no perturbation, no evasion budget, no adversarial example and no robustness experiment in the twenty-seven cells. The evaluation is one shuffled split of one static file. Meanwhile the corpus itself is seven malware families collected between two thousand and eleven and two thousand and thirteen.

SCIENTIFIC CONSEQUENCE

Botnet authors iterate against detectors as a matter of business. Every feature this model uses — flow duration, byte counts, packet counts, protocol state — is under the operator's direct control through padding, jitter and connection shaping, and none of those cost the operator anything. So the reported zero point nine eight nine four zero five is a best case against thirteen fixed captures of decade-old malware, and the worst case, which is the number a security claim must report, is not merely unreported but uncomputed.

REQUIRED REPAIR

State a perturbation budget — for instance, duration may be jittered by up to twenty percent and byte counts padded upward by up to thirty percent, both of which a botnet can do for free — and report the worst-case recall under it. That is the security claim; zero point nine eight nine four zero five is not.

DEFENSIBLE REPLY

My conclusion states that every number above measures in-distribution separability only. That is not a caveat attached to a robustness claim; it is a statement that no robustness claim is entered.

ORDERED CORRECTION

Re-aim at the editorial gap the scope sentence does not close — the title, the abstract and the course summary all read as detection capability, and the scope sentence is the last paragraph of section 13 — rather than at a missing experiment.

The robustness cell measures a different experiment

EVIDENCE

Cell twenty-four reads `cvX` comma `cvy` equals `Xtr` dot `iloc` up to forty thousand — and `Xtr` is by then the one-hundred-and-twenty-thousand-row subsample, so the printed zero point nine eight six zero plus or minus zero point zero zero one zero is a three-fold estimate on forty thousand rows, three point four percent of the training half. The four learners in cell sixteen carry fixed constructor arguments — `n_estimators` sixty and eighty, `max_depth` six — and no search object appears in the notebook. Section twelve is titled reproducibility and robustness.

SCIENTIFIC CONSEQUENCE

The comparison in cell sixteen is therefore a comparison of four libraries' default settings, not of four learners, and the winner-by-thirty-two-millionths in fault five is a comparison of defaults resolved below noise. Separately, the only dispersion figure in the notebook — plus or minus zero point zero zero one zero — estimates the variability of a forty-thousand-row three-fold experiment and is quoted in the abstract as if it characterised a three-hundred-and-ninety-three-thousand-row held-out score. Different `n`, different folds, different estimand.

REQUIRED REPAIR

Nested cross-validation on the reported pipeline: inner loop selects hyperparameters, outer loop estimates generalisation, and both run at the size actually reported. If that is too slow, shrink the experiment and say so — but do not label a forty-thousand-row spread as the robustness of the headline.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 393,014 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So the ruling. This is the strongest corpus in the set and the clearest demonstration of the problem, and those two facts are connected. You had real malware, you had thirteen natural distributions, you constructed the grouping variable yourself, you printed it, and you wrote MUST NOT be a feature in capital letters next to it — and then you split at random across it and published zero point nine eight nine four zero five. Ten out of thirty. Reject. The distance between this notebook and a publishable result is one argument to GroupKFold and thirteen folds, and you already have every input that call requires.

NetFlow-Standardized Intrusion Detection: NF-BoT-IoT-V2

NF-BoT-IoT-v2 — the Bot-IoT testbed corpus re-processed by Sarhan and colleagues into the standard NetFlow version two schema, pulled here from a third-party Kaggle mirror: thirty million four hundred and twenty thousand and eighty-six rows across forty-one features, with a positive rate of zero point nine nine five seven.

HEADLINE

XGBoost, held-out ROC-AUC zero point nine nine nine nine zero two — call it one — accuracy zero point nine nine nine six eight two on a seven million six hundred and five thousand and twenty-two row held-out split, with a three-fold cross-validated ROC-AUC of zero point nine nine nine seven plus or minus zero point zero zero zero two.

DEFENSE

F

Selected-model estimand

DOCTORAL

5

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 1 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 1 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 6 critical (FATAL)
- 15 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me be scrupulous about this, because you have something here the other notebooks do not. The praxis pack gave this ten out of ten, and on the pack's own terms that was correct. Your exact-duplicate rate is zero point zero zero zero and your train-test contamination is zero point zero zero zero — genuinely clean, both of them, and I am conceding that fully because two of your sibling notebooks have duplicate rates above forty percent. You also refused, in cell twenty-one, to read a surviving ablation as vindication, in these words: quote, if it barely moves, that is not vindication, end quote. That is the correct epistemology and almost nobody writes it down. But now hold the concession up to the light. NF-BoT-IoT-v2 is not a fresh capture. It is Bot-IoT, re-processed through a NetFlow converter. A duplicate rate of zero point zero zero zero proves that the conversion collapsed exact repeats. It does not prove the flows are independent — and I have the evidence that they are not, and it is in your own repository.

EXAMINER RULING

The answer to this notebook's central question is printed, in your own repository, in notebook thirty-six, on this exact Kaggle dataset — a model trained on BoT-IoT scores zero point three three one five on ToN-IoT — and this notebook publishes zero point nine nine nine nine zero two and points at the door.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Notebook thirty-six already answered you: zero point three three one five

EVIDENCE

Your section-thirteen conclusion reads, quote, the honest next step is train-on-one slash test-on-another. That is a test we did not run here, so we make no transfer claim. Notebook thirty-six runs that test on four NetFlow-standardized corpora and prints the transfer matrix, end quote. So open notebook thirty-six. Its loader dictionary reads SETS equals BoT-IoT colon d-hoogla slash nfbotiotv2 — character for character the same Kaggle reference as your cell eleven. Its transfer matrix is built with rows indexed by the corpus trained on and columns by the corpus tested on; the figure axis labels say so, y-label trained on, x-label tested on. The BoT-IoT row reads: zero point nine nine nine nine on itself, zero point three three one five on ToN-IoT, zero point six one two zero on UNSW-NB15, zero point seven eight six two on CSE-CIC-IDS2018. And its summary line prints, quote, cross-corpus off-diagonal mean zero point five four three four, min zero point three three one five, worse-than-random cells colon five of twelve, end quote.

SCIENTIFIC CONSEQUENCE

Zero point three three one five is not a degraded score. It is substantially below chance — a model trained on your corpus, applied to a sibling corpus built by the same authors from the same schema, orders the classes worse than a coin. And the whole justification for NF-BoT-IoT-v2's existence, stated in your own section one, is that, quote, the standard feature set was proposed to enable cross-dataset comparison, end quote. So the notebook's stated purpose is transferability, the test of transferability was run in the same repository on the same file by the same author, it returned a number below chance, and this notebook's abstract reports zero point nine nine nine nine zero two with no mention of it. Deferring to a later notebook is not a limitation when the later notebook already exists and already disagrees with you. It is a citation you declined to read.

REQUIRED REPAIR

Print notebook thirty-six's BoT-IoT row in this notebook's validity ledger, beside zero point nine nine nine nine zero two, in the same table. Then rewrite the abstract so the leading number is the transfer figure and the in-distribution figure is the footnote, because that is the direction the evidence points.

Ninety-nine point five seven percent attack is a script setting

EVIDENCE

Cell eleven prints, quote, loaded thirty million four hundred and twenty thousand and eighty-six rows x forty-one features; positive rate zero point nine nine five seven, end quote. Cell sixteen prints, quote, MAJORITY-CLASS BASELINE accuracy equals zero point nine nine five seven, end quote — and reads it out as the bar to beat. Now count the negatives. Cell eighteen prints one thousand one hundred and three benign flagged of thirty-two thousand three hundred and fifty-nine. So the entire benign class in a seven million six hundred and five thousand and twenty-two row held-out set is thirty-two thousand three hundred and fifty-nine rows — zero point four three percent of it. There are seven million five hundred and seventy-two thousand six hundred and sixty-three attack rows against thirty-two thousand three hundred and fifty-nine benign ones. Two hundred and thirty-four to one, in favour of attack.

SCIENTIFIC CONSEQUENCE

The majority-class baseline of zero point nine nine five seven is achieved by a box that says the word attack and nothing else, forever, and never looks at the traffic. That box has an accuracy your notebook prints as the standard every model must clear. On a real network the ratio is inverted by three to five orders of magnitude. Everything that follows inherits it: the accuracy column is meaningless, the ROC-AUC of zero point nine nine nine nine zero two is computed against a negative class that barely exists, and the false-positive rate rests on thirty-two thousand rows. This is not a detection problem with an unusual balance. It is a testbed in which the generator was left running, and the resulting ratio is a configuration value.

REQUIRED REPAIR

Re-weight or subsample the held-out set to a stated deployment base rate — one attack in a thousand flows, one in ten thousand — and report accuracy, precision, recall and the majority baseline at that rate. State the assumed rate in the abstract. Until that is done, no number in this notebook describes detection.

LightGBM at zero point seven two, and nobody said anything

EVIDENCE

The cell-sixteen table, bottom rows: LightGBM, accuracy zero point nine eight three three four six, roc-auc zero point seven two zero four eight four. MajorityBaseline, accuracy zero point nine nine five seven. So LightGBM's accuracy is below the trivial baseline by zero point zero one two three five four — it is worse than the box that always says attack. The conclusion's treatment is this sentence: quote, on some of these corpora a learner, the linear model, or LightGBM, lands well below the others or even below the majority-accuracy baseline, end quote. On some of these corpora. It happened on this one, the number is zero point seven two zero four eight four, and that number appears nowhere outside the table.

SCIENTIFIC CONSEQUENCE

Now put it beside notebook twelve. Same template, byte-identical cells sixteen through twenty-four, same hyperparameters, same forty-one-field NetFlow schema, sibling corpus from the same authors. On NF-ToN-IoT-v2 LightGBM scores zero point nine nine eight six nine nine and finishes second. On NF-BoT-IoT-v2 the identical configuration returns zero point seven two zero four eight four. That is a swing of point two seven eight across two files that are supposed to share a feature space precisely so they can be compared. That collapse is not a footnote — it is direct experimental evidence that these corpora are not interchangeable, and it is the strongest such evidence in your notebook. It is present in your printed table, it contradicts the schema's entire rationale, and the conclusion converts it into a hedge that names no number.

REQUIRED REPAIR

Name it in the conclusion with its number and investigate it. A learner that collapses on one corpus and wins on its sibling is the most interesting result in the notebook. And flag any sub-baseline row in the table itself — a model below the majority baseline is a diagnostic, not a ranking entry.

Zero duplicates proves the converter ran, not that the flows are independent

EVIDENCE

Cell twenty prints, quote, exact-duplicate row rate, whole corpus, equals zero point zero zero zero; TRAIN slash TEST exact-row contamination equals zero point zero zero zero, end quote, and grades the second an A. Both are equality tests: `dup_rate` is one minus `drop_duplicates` over length, and contamination is a set-membership test on tuples rounded to six decimals. Your section-three provenance table states the origin: quote, Bot-IoT, re-featured to NetFlow v2, end quote. And the ablation row in cell twenty-two is labelled de-duplicated, zero percent rows removed — the de-duplication experiment removed nothing, so it tested nothing.

SCIENTIFIC CONSEQUENCE

Both statistics are true and I concede them without reservation. But they answer a question nobody is asking. NF-BoT-IoT-v2 was produced by running the original Bot-IoT packet captures through a NetFlow converter, and a converter that aggregates packets into flow records with continuous byte counts, durations and throughputs will not emit two bit-identical rows even when two flows are the same beacon from the same device one second apart. So zero point zero zero zero is a fact about float precision in the conversion pipeline, not a fact about independence. The flows behind it come from a small fixed set of IoT devices running scripted attacks, which means the near-duplicate structure that actually leaks across a random split is untouched, unmeasured, and hidden precisely because the exact-duplicate rate came back clean. The A grade on contamination is the notebook's own evidence being used against it.

REQUIRED REPAIR

Fit a nearest-neighbour index on the training rows and plot the distance from every held-out row to its nearest training neighbour. If that distribution has mass near zero, the zero-duplicate rate is an artifact of continuous-valued conversion and must be reported as one. Then re-run the de-duplication ablation with a distance threshold instead of an equality test, so it removes something and therefore tests something.

Four tenths of one percent

EVIDENCE

Cell sixteen prints it in one line: quote, materialized thirty million four hundred and twenty thousand and eighty-six rows, trained on one hundred and twenty thousand, stratified subsample, held-out seven million six hundred and five thousand and twenty-two, end quote. One hundred and twenty thousand divided by thirty million four hundred and twenty thousand and eighty-six is zero point zero zero three nine four. Four tenths of one percent. Put it the other way: you discarded ninety-nine point six percent of the corpus and reported the remainder as a thirty-million-row result. And the assert in cell eleven that enforces the greater-than-one-million floor is checking the length of the dataframe, not the size of the fit — the honesty gate guards the download.

SCIENTIFIC CONSEQUENCE

The greater-than-or-equal-to-one-million claim in the subtitle describes what was downloaded. It has no relationship to the evidence. And unlike your sibling notebooks, where the subsample is five or eight percent, four tenths of one percent means the training set is smaller than the rounding error on the corpus size. Worse, of the twenty-two million eight hundred and fifteen thousand and sixty-four rows in the training half you actually fitted on half of one percent, and no learning curve exists anywhere in the notebook to show that the score has saturated. Saturation is asserted in a code comment and never measured.

REQUIRED REPAIR

A learning curve out to at least three million rows with held-out AUC on the vertical axis, so the saturation claim becomes a measurement. And put the effective n — one hundred and twenty thousand — into the abstract, in the same sentence as the AUC, where thirty million currently stands.

DEFENSIBLE REPLY

The quantity you are attacking is measured on 7,605,022 held-out rows. Only the fit is 120,000, and that prints in my output and is stated in section eight.

ORDERED CORRECTION

Replace the generic scale complaint with the specific one, which is far worse and which you do not make: at a positive rate of 0.9957 the 120,000-row stratified fit contains about 511 benign flows, drawn from the 97,078 available in the training half. The reported model saw five hundred negatives. That is the number that should appear in the abstract, and it is the mechanism behind fault 4 and fault 20.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

A handful of IoT devices, split at random

EVIDENCE

Cell sixteen calls `train_test_split` with test size zero point two five, stratified on `y`. `GroupKFold`, `GroupShuffleSplit` and the `groups` argument do not appear anywhere in the twenty-seven cells. Your own conclusion states the consequence: quote, the split is random, not temporal or entity-grouped. Every number above therefore measures in-distribution separability only, end quote.

SCIENTIFIC CONSEQUENCE

Bot-IoT is a testbed. Its attack traffic was generated by scripts running on a small number of machines against a small number of simulated IoT services, which means a single scripted DDoS run contributes hundreds of thousands of near-identical rows. Scatter that run across a random split and the model scores held-out rows by recognising their siblings. That is a sufficient mechanism for zero point nine nine nine zero two on its own, and it is the mechanism notebook thirty-six's zero point three three one five confirms: what survives the shuffled split does not survive a change of corpus. You wrote the sentence identifying this and then reported the number it invalidates.

REQUIRED REPAIR

Group on device identity or on attack session and split with `GroupShuffleSplit` so no run appears on both sides. Report the AUC under that split beside zero point nine nine nine zero two. That is the number the abstract should carry.

DEFENSIBLE REPLY

Open the parquet. There are forty-three columns and not one of them is an address, a device identifier or a session identifier — the mirror stripped `IPV4_SRC_ADDR` and `IPV4_DST_ADDR` from Sarhan's published forty-three-feature schema. You are prescribing a `groups` argument for a column that is not in the file.

ORDERED CORRECTION

Re-mechanise as a two-step charge and merge with fault 9: the authoritative NF-BoT-IoT-v2 release carries `IPV4_SRC_ADDR` and `IPV4_DST_ADDR`, this mirror does not, and the notebook selected the mirror without checking — so the grouped split was made impossible by a provenance decision that is itself undisclosed. That is a true and checkable charge. 'Group on device identity and split with `GroupShuffleSplit`' as written is refuted by opening the file.

You never print which forty-one columns the model sees

EVIDENCE

Cell eleven prints one number and no names: forty-one features. The DROP list is exactly four entries — Label, Attack, y and family — all label-derived. Not one identifier column is named for removal. Compare notebook nine, whose DROP list explicitly names srcip, dstip, sport, dsport, Stime and Ltime, and notebook ten, which names StartTime, SrcAddr, DstAddr, Sport and Dport. Here the only other guard reads, quote, idlike equals columns where nunique is greater than zero point five times the length of X, end quote — a text column is dropped only if it has more distinct values than half the corpus. On thirty million four hundred and twenty thousand and eighty-six rows that threshold is over fifteen million distinct values.

SCIENTIFIC CONSEQUENCE

That guard removes a primary key. It cannot remove host identity, because host identity in a testbed has a small number of distinct values and sails through a fifteen-million threshold untouched, gets integer-encoded by the LabelEncoder two lines below, and becomes a feature. And here is the part that ends the argument: neither you nor I can tell from this PDF whether that happened, because the surviving column names are never printed. A pipeline whose feature set cannot be enumerated from its own output is not auditable, and an unauditable pipeline cannot support a claim to four decimal places. Everything downstream — the single-feature scan, the importance plot, the ablation — reports names for individual columns while never reporting the set they were drawn from.

REQUIRED REPAIR

Print the full sorted list of the forty-one feature names in cell eleven. Name every identifier column explicitly in DROP, as notebooks nine and ten do, rather than relying on a cardinality heuristic. And print which columns idlike and the constant filter removed, and why.

DEFENSIBLE REPLY

Open the file. The only non-numeric column in that parquet is Attack, and Attack is in DROP. There is nothing for the LabelEncoder to encode and no host identity to sail through your threshold, because the mirror ships no address column at all.

ORDERED CORRECTION

Delete the entire 'host identity ... becomes a feature' paragraph — it describes a latent defect of the shared template, not an event in this notebook. What survives is smaller but true: the feature list is never printed, so the reader cannot enumerate the model's inputs from the notebook's own output, and the DROP list names no identifier because the loader relies on a cardinality heuristic that on 30.4 million rows fires only above 15.2 million distinct values. Keep that at CONTESTED and stop claiming 'neither you nor I can tell', because the student can open the parquet in the room and read out forty-three column names.

You cite Sarhan and you download from someone else

EVIDENCE

Section three's provenance table gives the source as Kaggle d-hoogla slash nfbotiotv2 and the origin as, quote, Bot-IoT, re-featured to NetFlow v2, end quote, citing Sarhan et al. two thousand and twenty-two, arXiv two one zero one point one one three one five. Cell eleven downloads from that Kaggle reference. There is no hash check, no file listing, no row-count comparison against the authors' own release, and no statement of which version of the mirror was pulled.

SCIENTIFIC CONSEQUENCE

The uploading account is not the crediting author. Every number in this notebook — thirty million four hundred and twenty thousand and eighty-six rows, forty-one features, a positive rate of zero point nine nine five seven, and a duplicate rate of exactly zero — is a property of that mirror, and none of them has been checked against what Sarhan and colleagues published. The duplicate rate of exactly zero on thirty million rows is itself a signal that the mirror may be pre-cleaned relative to the original release, and the provenance table does not say so because nobody checked. In a course whose whole argument is that provenance determines what a score means, downloading the file from a stranger and citing the authors is the specific error being taught against.

REQUIRED REPAIR

Record the mirror's file hashes and row counts, compare them to the authors' published distribution, and print the comparison. If they differ, report the difference. If the authoritative release is unavailable, say so in the provenance table in those words.

Three point four one percent, multiplied out

EVIDENCE

Cell eighteen prints, quote, operational FALSE-POSITIVE RATE at zero point five equals zero point zero three four one, one thousand one hundred and three benign flagged of thirty-two thousand three hundred and fifty-nine, end quote. Section two cites Sommer and Paxson two thousand and ten, reference five, whose central argument is that base rates defeat network anomaly detectors in the field. That citation is never applied to this number.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic you owed, and I am stating my assumption aloud so it can be checked: take a network carrying ten million benign flows a day — that figure is mine, not yours, and the multiplicand has to be the benign volume because an FPR is a rate over negatives. Ten million times zero point zero three four one is three hundred and forty-one thousand false alarms per day. And the estimate itself rests on thirty-two thousand three hundred and fifty-nine negative rows, which is the entire benign population of a seven-million-row held-out set. So the one figure in this notebook that carries the word operational is both unmultiplied and thinly estimated, and it is the worst false-positive rate of the four notebooks in front of me.

REQUIRED REPAIR

Print alerts per analyst per day for a stated benign flow volume, in the results cell. Then state the alert budget a deployment can absorb and read the required threshold off the precision-recall curve you already drew.

DEFENSIBLE REPLY

You supplied the ten million yourself and said so. And on the corpus you are multiplying, benign traffic is 0.43% of flows — the volume you assume does not describe anything in this notebook. The criticism that survives without your number is the one you already made in fault three.

ORDERED CORRECTION

Merge into fault 3. Sommer & Paxson's argument is about the base rate of attacks, not FPR times volume, so fault 3's re-weighting demand is the correct application of the citation.

An operating point nobody chose, on a class that barely exists

EVIDENCE

Cell eighteen reads pred equals pb greater than zero point five. The threshold appears in the printed label because it is the scikit-learn default, not because it was chosen. There is no threshold sweep, no expected-cost calculation, no cost matrix and no threshold-selection step anywhere in the twenty-seven cells.

SCIENTIFIC CONSEQUENCE

Two problems compound. Under asymmetric costs the operating point is the result, and zero point five is not an operating point, it is a default. And here it is applied to a negative class of thirty-two thousand three hundred and fifty-nine rows against seven million five hundred and seventy-two thousand six hundred and sixty-three positives — a prior so extreme that the model's zero point five contour sits in a region of the score distribution containing almost no benign traffic at all. Moving the threshold by any sensible amount would swing the false-positive rate substantially, and nothing here reports that sensitivity.

REQUIRED REPAIR

State a cost ratio, build the cost matrix, choose the threshold that minimises expected cost on a validation fold, print the chosen threshold, and print the false-positive rate and per-family recalls at it. Then show how far the answer moves when the ratio changes.

DEFENSIBLE REPLY

The number in my abstract is a ROC-AUC, which is invariant to the threshold. The block you are attacking prints the threshold in its own label.

ORDERED CORRECTION

Scope to the operating-point block and merge the prior argument into fault 3.

The right curve, drawn, and not read

EVIDENCE

Cell eighteen calls `precision_recall_curve` on the held-out probabilities and plots it as panel three, titled Precision-Recall. That is its entire use. `average_precision_score` appears nowhere in the notebook — nothing computes the area, prints it, or ranks a model on it. Cell sixteen sorts the table on `roc_auc`, and the abstract leads with zero point nine nine nine nine zero two.

SCIENTIFIC CONSEQUENCE

With seven million five hundred and seventy-two thousand six hundred and sixty-three positives and thirty-two thousand three hundred and fifty-nine negatives, precision is near one before the model does anything, and ROC-AUC is dominated by the ordering of a positive class that constitutes almost the entire set. Neither metric can see the only difficult decision in this corpus, which is whether a benign flow is correctly left alone. And the correct criticism is not that you lacked the right diagnostic. It is on your page, as a picture, and you did not read it. The relevant quantity here is the recall on the benign class — one minus zero point zero three four one — and it never appears as a headline anywhere.

REQUIRED REPAIR

Compute and print `average_precision_score` with a bootstrap interval, and compute it with the labels inverted so it measures the benign class, which is the minority. Report specificity — benign recall — as a first-class number in the abstract.

DEFENSIBLE REPLY

You have it backwards. ROC-AUC is a rank statistic over one positive and one negative drawn independently; it does not depend on the class ratio at all. On my split one false positive costs $3.09\text{e-}5$ of false-positive rate and $1.32\text{e-}7$ of precision — ROC-AUC is two hundred and thirty-four times more sensitive to it. The metric that cannot see the benign class here is precision, which is 0.9957 before my model does anything.

ORDERED CORRECTION

Re-mechanise to the form CORRECTION 2 prescribes and which your own REQUIRED already half-reaches: the precision-recall panel is drawn with the MAJORITY class as positive, so its area is uninformative — a trivial all-positive classifier scores 0.9957 — and the rare class is the benign one, whose operating point (specificity, benign-class average precision, benign recall) is never established anywhere. Drop 'ROC-AUC is dominated by the ordering of a positive class that constitutes almost the entire set'; that sentence loses the argument in the room.

Three recalls of exactly one, and not one denominator

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, Theft zero point nine eight two, Reconnaissance one point zero, DDoS one point zero, DoS one point zero, end quote. Three of the four are exactly one. The loop that produces them reads for fam comma cnt in value_counts, and cnt is used only in the guard if cnt is less than five, continue — never printed. There is no bootstrap, no confidence interval and no binomial interval anywhere in this notebook.

SCIENTIFIC CONSEQUENCE

A recall of exactly one point zero is either a statement that a family has no missed positives across millions of rows, or a statement that the family is too small for the estimate to mean anything, and the reader cannot tell which because the denominator is withheld. The conclusion then converts this into, quote, the weakest group sits at zero point nine eight two, which is where detection is thinnest, end quote — a sentence describing a corpus in which nothing is thin, which is itself the finding. When your worst per-family recall is zero point nine eight two, the correct inference is not that detection is strong. It is that the classes barely overlap, which is a property of the generator.

REQUIRED REPAIR

Print every family with its held-out positive count and a Clopper-Pearson interval on its recall. A recall of one point zero on a small count should be printed with its interval, which will not be tight, and the difference will be visible.

DEFENSIBLE REPLY

Reconnaissance, DoS and DDoS are 2.36 million, 13.6 million and 14.3 million rows of this corpus; their held-out recalls of 1.000 are not small-sample artifacts, they are statements that the model missed under one in two thousand. Only Theft is small, at about 579 held-out positives, and its interval on 0.982 is roughly [0.967, 0.992].

ORDERED CORRECTION

Keep the B2 defect — no denominator, no interval, anywhere in the notebook — and drop the 'reader cannot tell which' framing, which is answerable. The strongest sentence in this fault is the last one and it should lead: when the worst per-family recall on thirty million rows is 0.982, the correct inference is not that detection is strong but that the classes barely overlap, which is a property of the generator.

Six decimals on a quantity whose ablation moves upward

EVIDENCE

Cell twenty-two prints the ablation: headline as-is zero point nine nine nine nine zero two; shortcut feature dropped, SRC_TO_DST_SECOND_BYTES, zero point nine nine nine nine one one. The score rose by nine millionths after deleting the strongest single column. The comparison table alongside it reports every model to six decimals, and the winner-versus-runner-up gap in that table is three hundred and thirty-five millionths.

SCIENTIFIC CONSEQUENCE

A measurement whose value increases when you remove information is a measurement whose noise floor exceeds the effect being reported. The conclusion nonetheless narrates the movement as, quote, dropping that feature barely moves the AUC: zero point nine nine nine nine zero two arrow zero point nine nine nine nine one one, end quote — it did not barely move, it moved the wrong way, and the notebook's own How-the-audit-numbers-are-computed paragraph concedes the mechanism: quote, each ablation re-splits and refits, so tiny differences are re-split noise, end quote. Having identified that the sixth decimal is noise, the notebook keeps reporting six decimals and drawing conclusions from differences smaller than that.

REQUIRED REPAIR

Report to the resolution the estimator supports. Run each ablation across at least ten re-splits, print mean and standard deviation, and round every table to the first digit that survives. If the standard deviation is one ten-thousandth, print four decimals and stop.

DEFENSIBLE REPLY

The six decimals are there by design, and the comment on that line says why: at these scores a 1.000000 is a red flag and I want it visible. Rounding to four would hide the thing you are asking me to show.

ORDERED CORRECTION

Drop the reporting-precision half. Keep the true residual, which is stronger anyway: each ablation is run once, so no dispersion exists for any of the three settings, and the notebook diagnoses its own resolution problem ('tiny differences are re-split noise') without ever measuring it across re-splits.

A B on a two-question quiz

EVIDENCE

Cell twenty computes precisely two things and takes the worse letter: a single-feature AUC of zero point eight eight two three, graded B against a zero point eight five cutoff, and a contamination rate of zero point zero zero zero, graded A. The printed line reads, quote, data trust grade colon B, worse of the two, end quote. The single-feature figure comes from one draw of sixty thousand rows out of thirty million four hundred and twenty thousand and eighty-six — zero point two percent of the corpus — with random_state one, and no interval is computed for it. The contamination figure is measured on the first fifty thousand of seven million six hundred and five thousand and twenty-two held-out rows: zero point six six percent of the test set.

SCIENTIFIC CONSEQUENCE

The quiz asks two questions. It does not ask whether the split respects device identity, whether it respects time, whether near-duplicates cross the boundary, whether the score transfers to another corpus, or whether an adversary can move the features. Every one of those is unmeasured, and four of them are known from your own repository to fail — notebook thirty-six's zero point three three one five settles the transfer question outright. So the letter B is not a summary of trustworthiness. It is the score on two of the eight questions that matter, and it is being read across the course as the first thing.

REQUIRED REPAIR

Rename it what it is — a shortcut-and-overlap check — or extend it to cover grouping, time, near-duplicates and transfer. And attach an interval to the zero point eight eight two three before assigning any letter to it.

DEFENSIBLE REPLY

Every time that letter prints, the line beside it names the two checks it is made of, section ten defines it as the worse of the single-feature and contamination concerns, and the ledger spells out both sub-grades. It is not presented as a summary of everything.

ORDERED CORRECTION

Keep the rename demand — 'shortcut-and-overlap check' is right and cheap — and drop the implication of concealment, since the components are printed on the same line as the letter. The load-bearing version of this fault lives in nb12, where the ledger promotes two passing checks into 'nothing here explains the score away'; nb11's corresponding sentence is narrower and only claims that overlap is not the issue.

The correct reading was written and then not acted on

EVIDENCE

Cell twenty-one states the standard precisely: quote, if it barely moves — common on simulated corpora — that is not vindication. It means the classes are separable by many redundant features, because the attack and benign distributions barely overlap. That is its own generation artifact, end quote. Cell twenty-two then produces exactly that outcome: zero point nine nine nine nine zero two to zero point nine nine nine nine one one. And the conclusion writes, quote, the ablation refutes a single-feature story, end quote, and moves on. No overlap statistic is computed anywhere.

SCIENTIFIC CONSEQUENCE

You wrote down the correct interpretation before running the experiment, the experiment returned the outcome your own prose identified as the bad case, and the conclusion reported it under the heading the ablation refutes a single-feature story — which is true and irrelevant. It is not a single-feature story. It is a story in which forty-one features each independently separate the classes because the two distributions were produced by different scripts. That is a strictly worse finding than a single leaky column, because you cannot ablate your way out of it. And notebook thirty-six's zero point three three one five is what that finding looks like when you change corpus.

REQUIRED REPAIR

Measure the overlap directly: a two-sample energy or maximum-mean-discrepancy statistic between the benign and attack feature distributions, or the Bayes-error estimate from a density-ratio fit. Print it. Then rewrite the ablation paragraph so its conclusion is the one cell twenty-one already tells you to draw.

DEFENSIBLE REPLY

Read the next two sentences. My ledger says the separability is multi-feature and that it reflects how this corpus was generated, not one leaky column — which is exactly the reading section eleven prescribes. I did not move on from it; I wrote it down.

ORDERED CORRECTION

Correct the quotation and keep the residual, which is real and cheap to fix: the notebook states the generation-artifact reading in prose and never measures it. An MMD or energy statistic between the benign and attack feature distributions, or a density-ratio Bayes-error estimate, would turn a sentence into a number. That is a **CONTESTED** omission, not a **FATAL** misreading.

A byte-rate is a throttle setting

EVIDENCE

Cell twenty prints, quote, best single-feature AUC equals zero point eight eight two three, feature colon SRC_TO_DST_SECOND_BYTES, end quote. That column is bytes per second from source to destination — a volumetric rate. The four attack families in cell eighteen are Theft, Reconnaissance, DDoS and DoS, three of which are defined by their volume.

SCIENTIFIC CONSEQUENCE

Bot-IoT's attacks were produced by scripts with configured rates against a testbed with configured link capacities. A byte-rate column therefore encodes the script's throttle and the lab's bandwidth. An adversary changes it with a sleep call. And because your ablation shows the score is untouched when this column is removed, the alternative explanation is not that the model found something deeper — it is that the other forty columns carry the same generator signature. Zero point nine nine nine nine zero two is a measurement of how the traffic was made.

REQUIRED REPAIR

Partition the forty-one columns into rate-and-volume features and structural features, and report the held-out AUC on the structural set alone. And then report both under notebook thirty-six's cross-corpus protocol, because that is the only test that distinguishes generator signature from transferable signal.

DEFENSIBLE REPLY

0.8823 is my B grade, not my F grade. My single-feature check exists to catch a near-sufficient column and it reports that no column is near-sufficient. What you are actually arguing is that the signature is spread across all forty-one features, which is fault sixteen.

ORDERED CORRECTION

Merge with fault 16 or keep only the uncomputed experiment: no rate-and-volume versus structural feature partition is reported, so no number in the notebook separates generator signature from transferable signal.

A botnet detector with no botnet operator in it

EVIDENCE

There is no perturbation, no evasion experiment and no robustness budget in the twenty-seven cells. The evaluation is one stratified random split of one static file. The features carrying the score are, by the notebook's own single-feature scan and importance plot, volumetric rate statistics.

SCIENTIFIC CONSEQUENCE

Every feature this model relies on is under the attacker's direct control at zero cost: rate through throttling, duration through pacing, packet-size distributions through padding, TCP flag patterns through connection shaping. None of that requires a new exploit or a new capability. A security claim must report its worst case, and here the worst case is not merely unreported — it is uncomputed, on a corpus where the adversary's cost of evading every top feature is a configuration change.

REQUIRED REPAIR

State a perturbation budget in feature space — for example, all rate and volume columns may be scaled by any factor between zero point one and ten, which a script can do trivially — and report the worst-case held-out recall under it. That number is the security claim.

DEFENSIBLE REPLY

My conclusion states that every number above measures in-distribution separability only. No robustness claim is made, so there is no worst case I have failed to report against a claim I entered.

ORDERED CORRECTION

Re-aim at the editorial gap: the abstract leads with 0.999902 and the scope sentence is the last line of section 13.

Four families is the whole taxonomy

EVIDENCE

Cell eighteen prints exactly four families: Theft, Reconnaissance, DDoS and DoS. The guard in the same loop reads if cnt is less than five, continue, and the number of families it skipped is never printed. The conclusion then reports, quote, the weakest group sits at zero point nine eight two, which is where detection is thinnest, end quote.

SCIENTIFIC CONSEQUENCE

Four categories over thirty million rows is not a taxonomy, it is a coarse partition, and three of the four report a recall of exactly one. The per-family breakdown is the mechanism the whole series relies on to expose rare-class failure, and on this corpus it has no resolving power at all — yet the conclusion reports its output in the same confident register as notebook ten's, where the same code exposes a family at zero point one eight eight. When the diagnostic returns nothing, the notebook should say the diagnostic returned nothing.

REQUIRED REPAIR

Print every family with its positive count, print how many the guard skipped, and add a sentence to the conclusion stating that with four families and three perfect recalls the per-family diagnostic carries no information on this corpus.

DEFENSIBLE REPLY

Four is the whole Bot-IoT taxonomy, all four are printed and all four are plotted, and the guard skipped nothing — the smallest family has about 579 held-out positives against a threshold of five.

ORDERED CORRECTION

Drop the skip-count clause and keep the strong part verbatim: with four families and three recalls of exactly 1.0, the diagnostic the whole series relies on to expose rare-class failure returns nothing on this corpus, and the conclusion reports its output in the same confident register as nb10's, where the identical code exposes a family at 0.188. Your REQUIRED — a sentence in the conclusion saying the diagnostic carries no information here — is exactly right.

Zero point seven two zero four eight four is what untuned defaults look like

EVIDENCE

Cell twenty-four reads cvX comma cvy equals Xtr dot iloc up to forty thousand — and Xtr is by then the one-hundred-and-twenty-thousand-row subsample. So the printed zero point nine nine nine seven plus or minus zero point zero zero zero two is a three-fold estimate on forty thousand rows, which is zero point one eight percent of the training half. The four learners in cell sixteen carry fixed constructor arguments and no search object appears anywhere. Section twelve is titled reproducibility and robustness.

SCIENTIFIC CONSEQUENCE

You have the proof of the harm in your own table. LightGBM returns zero point seven two zero four eight four on this corpus with default settings and second place on its sibling corpus with the same settings — that is what an untuned learner on an extreme prior does, and a single round of tuning or a `scale_pos_weight` argument would very likely have fixed it. So the model comparison is a comparison of four libraries' defaults, not four learners. And separately, the only dispersion figure in the notebook estimates the variability of a forty-thousand-row three-fold experiment and is quoted as though it characterised a seven-million-row held-out score. Different n , different folds, different estimand.

REQUIRED REPAIR

Nested cross-validation on the reported pipeline — inner loop selects, outer loop estimates — with `scale_pos_weight` in the search space given a prior of zero point nine nine five seven. And run the CV at the size actually reported, or state clearly that the printed spread describes a different, smaller experiment.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 7,605,022 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Closing. So the ruling. This notebook took ten out of ten from the friendly audit, and it deserved it — the duplicate rate really is zero, the contamination really is zero, and cell twenty-one really does contain the sentence that a surviving ablation is not vindication. And none of that matters, because the question this corpus exists to answer was answered, in your own repository, in notebook thirty-six, on this exact Kaggle reference, and the answer was zero point three three one five — worse than chance, with five of twelve cross-corpus cells below random. You had that number and you published zero point nine nine nine zero two. Five out of thirty. Reject with prejudice — not because the work is careless, but because the reported quantity is a property of the file, and the file's own author already proved it.

NetFlow-Standardized Intrusion Detection: NF-ToN-IoT-V2

NF-ToN-IoT-v2 — Moustafa's ToN-IoT testbed captures re-processed by Sarhan and colleagues into the standard NetFlow version two schema, pulled here from a third-party Kaggle mirror: thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows across forty-one features, positive rate zero point seven two five eight.

HEADLINE

XGBoost, held-out ROC-AUC zero point nine nine eight eight six three, accuracy zero point nine nine zero six seven three on a three million two hundred and eighty-three thousand nine hundred and seventy-one row held-out split, with a three-fold cross-validated ROC-AUC of zero point nine nine eight two plus or minus zero point zero zero zero one.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 7 critical (FATAL)
- 14 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to give this notebook its due, because it earns more than any other in the set. It is the only one of the four in front of me that reads its own comparison table properly — it names XGBoost at zero point nine nine eight eight six three, LightGBM at zero point nine nine eight six nine nine, Random Forest at zero point nine nine seven six seven zero and Logistic Regression at zero point eight three nine four four three, checks each accuracy against the baseline of zero point seven two five eight, and states that no learner falls below it. Your strongest single feature reaches only zero point seven six three seven, the lowest and therefore the most honest of the four notebooks. Your duplicate rate is zero and your contamination is zero, and I concede both without reservation. And your literature review is the best in the set — you found Layeghy and Portmann's two thousand and twenty-three cross-domain study and you quoted its finding accurately. The praxis pack gave this ten out of ten and on the pack's terms that was right. Now watch what happens when the master frame is applied to that same literature review.

EXAMINER RULING

This is the cleanest notebook in the set by its own audit — trust grade A, zero duplicates, zero contamination — and it is the one I am rejecting hardest, because you cite the paper that says your score will not transfer, you cite the notebook in your own repository that measures how far it does not transfer, and then you print zero point nine nine eight eight six three and call the audit clean.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

You cited the finding, then published against it

EVIDENCE

Your section-thirteen conclusion reads, quote, a cross-domain study covered four NetFlow-standardized NIDS datasets. No model it tested generalised across all four, Layeghy and Portmann two thousand and twenty-three, end quote — reference six in your own list. It continues, quote, the honest next step is train-on-one slash test-on-another. That is a test we did not run here, so we make no transfer claim, end quote. Now open notebook thirty-six in the same directory. Its SETS dictionary reads ToN-IoT colon d-hoogla slash nftoni0tv2 — character for character your cell eleven's Kaggle reference. Its transfer matrix is indexed rows by trained-on and columns by tested-on; the figure's axis labels say exactly that. The ToN-IoT row reads: zero point nine nine nine zero on itself, zero point four nine one zero on BoT-IoT, zero point three eight two two on UNSW-NB15, zero point five four six eight on CSE-CIC-IDS2018. And in the other direction, a model trained on BoT-IoT scores zero point three three one five when tested on your corpus. Its summary line prints, quote, cross-corpus off-diagonal mean zero point five four three four, min zero point three three one five, worse-than-random cells colon five of twelve, end quote.

SCIENTIFIC CONSEQUENCE

Zero point three eight two two is not a degraded score. It is well below chance — the model orders the classes backwards. Zero point four nine one zero is chance exactly. And the entire stated rationale for this corpus, in your own section one, is that, quote, the standard feature set was proposed to enable cross-dataset comparison, end quote. So the notebook's purpose is transferability, the paper that measures transferability is in your reference list, the notebook that measures it is in your repository, both say it fails, and the abstract reports zero point nine nine eight eight six three with no mention of either. Deferring to a later notebook is not a limitation when the later notebook already exists and already contradicts you. This is not a gap in the work. It is a finding the work declined to carry.

REQUIRED REPAIR

Print notebook thirty-six's ToN-IoT row inside this notebook's validity ledger, in the same table as zero point nine nine eight eight six three. Then rewrite the abstract so the transfer figures lead and the in-distribution figure is the footnote, because that is the direction your own cited evidence points.

An A on a two-question quiz, spent as a clean bill of health

EVIDENCE

Cell twenty computes two things and takes the worse letter: a single-feature AUC of zero point seven six three seven, graded A because it falls below the zero point eight five cutoff, and an exact-row contamination of zero point zero zero zero, graded A. The printed line is, quote, data trust grade colon A, worse of the two, end quote. And then the conclusion writes the sentence that decides this fault: quote, neither check flags a problem, so nothing here explains the score away, end quote.

SCIENTIFIC CONSEQUENCE

That sentence is the failure, and it is worth reading twice because it is the most quotable line in the thirty-six notebooks. Two checks passed, and the conclusion promotes that into a claim about everything. The quiz does not ask whether the split respects device identity. It does not ask about time. It does not ask about near-duplicates, which is the leakage that survives a NetFlow conversion. It does not ask about adversarial robustness. It does not ask whether the score transfers — and that last one is not merely unasked, it is answered elsewhere in your own repository at zero point three eight two two. So the A is the score on two of eight questions, and it is being spent as though it were the score on all eight. A notebook that grades its own data A and cannot transfer above chance has learned that its grading instrument is too short, and that is the finding.

REQUIRED REPAIR

Rename the grade what it is — a shortcut-and-overlap check — or extend it to cover entity grouping, temporal ordering, near-duplicate distance and cross-corpus transfer. And delete the sentence nothing here explains the score away, because notebook thirty-six explains the score away.

Zero duplicates proves the converter ran

EVIDENCE

Cell twenty prints, quote, exact-duplicate row rate, whole corpus, equals zero point zero zero zero; TRAIN slash TEST exact-row contamination equals zero point zero zero zero, end quote. Both are equality tests — `dup_rate` is one minus `drop_duplicates` over length, contamination a set-membership test on tuples rounded to six decimals. Your provenance table names the origin: ToN-IoT, Moustafa two thousand and twenty-one, re-featured to NetFlow v2 by Sarhan et al. two thousand and twenty-two. And the ablation row in cell twenty-two reads de-duplicated, zero percent rows removed — that experiment removed nothing and therefore tested nothing. There is no nearest-neighbour distance and no near-duplicate threshold anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

I concede both numbers fully. But they answer a question nobody asked. A NetFlow converter emits records carrying continuous byte counts, durations, throughputs and packet-size statistics, and two records will essentially never be bit-identical even when they are the same beacon from the same device one second apart. So zero point zero zero zero is a fact about float precision in the conversion pipeline, not a fact about independence. The flows behind it come from a small fixed testbed running scripted attacks, and the near-duplicate structure that actually leaks across a shuffled split is untouched, unmeasured, and hidden precisely because the exact-duplicate rate came back clean. Your A grade on contamination is your own evidence being used against you.

REQUIRED REPAIR

Fit a nearest-neighbour index on the training rows and plot the distance from every held-out row to its nearest training neighbour. If that distribution has mass near zero, report the zero-duplicate rate as an artifact of continuous-valued conversion. Then re-run the de-duplication ablation with a distance threshold rather than an equality test, so it removes something.

Nine tenths of one percent

EVIDENCE

Cell sixteen prints it: quote, materialized thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows, trained on one hundred and twenty thousand, stratified subsample, held-out three million two hundred and eighty-three thousand nine hundred and seventy-one, end quote. One hundred and twenty thousand over thirteen million one hundred and thirty-five thousand eight hundred and eighty-one is zero point zero zero nine one three five. Nine tenths of one percent. Against the training half specifically — nine million eight hundred and fifty-one thousand nine hundred and ten rows — you fitted on one point two two percent. The assert in cell eleven that enforces the one-million floor tests the length of the dataframe, not the size of the fit.

SCIENTIFIC CONSEQUENCE

The greater-than-or-equal-to-one-million claim in the subtitle describes the download, not the evidence, and a reader who takes thirteen million as the scale of the experiment is reading the notebook exactly as written. No learning curve exists anywhere in the twenty-seven cells, so the justification offered in the cell-sixteen comment — that these learners saturate long before then — is asserted and never measured. And if it is true, it is the most interesting sentence in the notebook and it is unreported: a corpus on which a model saturates at nine tenths of one percent of the rows is a corpus with very little variety in it.

REQUIRED REPAIR

A learning curve from ten thousand to at least two million rows, held-out AUC against n , printed. Then put the effective n of one hundred and twenty thousand into the abstract in the same sentence as zero point nine nine eight eight six three.

DEFENSIBLE REPLY

The line 'trained on 120,000 (stratified subsample)' prints in my output and section eight repeats it in prose. The quantity you are attacking is measured on 3,283,971 held-out rows.

ORDERED CORRECTION

Keep only the unanswerable residual — no learning curve, so the saturation justification is asserted and never measured — and keep the good closing observation, that a corpus on which a model saturates at nine tenths of one percent of its rows is a corpus with very little variety in it. That sentence is the real finding and it deserves to be the fault.

The instrument returned the same number three times

EVIDENCE

Cell twenty-two, in full: headline as-is, zero point nine nine eight eight six three. De-duplicated, zero percent rows removed, zero point nine nine eight eight six three. Shortcut feature dropped, LONGEST_FLOW_PKT, zero point nine nine eight eight six three. Three settings, one number, six decimals. The conclusion then writes, quote, the ablation refutes a single-feature story. Dropping that feature barely moves the AUC: zero point nine nine eight eight six three arrow zero point nine nine eight eight six three, end quote — it did not barely move, it did not move at all — and, quote, so the separability is multi-feature, end quote.

SCIENTIFIC CONSEQUENCE

Section eleven tells you how to read this, in your own words: quote, if it barely moves — common on simulated corpora — that is not vindication. It means the classes are separable by many redundant features, because the attack and benign distributions barely overlap. That is its own generation artifact, end quote. You wrote the correct interpretation before running the experiment, the experiment returned the exact case your prose identified as the bad one, and the conclusion reported it under a heading that reads like exoneration. And a redundancy that deep is strictly worse than a leaky column, because you cannot ablate your way out of it — every one of the forty-one features carries the same testbed signature. Notebook thirty-six's zero point three eight two two is what that redundancy is worth off this corpus.

REQUIRED REPAIR

Measure the class overlap directly — a maximum-mean-discrepancy or energy statistic between benign and attack feature distributions, or a Bayes-error estimate from a density-ratio fit — and print it. Then rewrite the ablation paragraph to draw the conclusion your own section eleven already instructs.

DEFENSIBLE REPLY

Read the next two sentences of the ledger. I say the separability is multi-feature and that it reflects how this corpus was generated, not one leaky column. That is section eleven's reading, written down.

ORDERED CORRECTION

Correct the quotation, then keep two residuals that survive it. First, the ledger's auto-generated sentence 'Dropping that feature barely moves the AUC: 0.998863 -> 0.998863' describes a delta of exactly zero as 'barely moves', which is a template artifact producing a false statement in the notebook's own conclusion — quote that line, it is better than the one you used. Second, the generation-artifact reading is stated and never measured; an MMD or energy statistic between the class-conditional feature distributions would turn it into a number.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

A testbed of a few devices, split at random

EVIDENCE

Cell sixteen calls `train_test_split` with test size zero point two five, stratified on `y`. `GroupKFold`, `GroupShuffleSplit` and the groups argument appear nowhere in the twenty-seven cells. Your own conclusion states the consequence: quote, the split is random, not temporal or entity-grouped. Every number above therefore measures in-distribution separability only, end quote.

SCIENTIFIC CONSEQUENCE

ToN-IoT is a testbed with a small IoT device population running scripted attacks — a single scanning run or a single ransomware execution contributes tens of thousands of near-identical rows. Scatter that run across a shuffled split and the model can score a held-out row by recognising its siblings from training. That mechanism alone is sufficient to produce zero point nine nine eight eight six three, and notebook thirty-six's zero point three eight two two is the confirmation: what survives the shuffle does not survive a change of corpus. You wrote the sentence identifying the defect and reported the number it invalidates.

REQUIRED REPAIR

Group on device identity or attack session and split with `GroupShuffleSplit` so no run appears on both sides. Report the AUC under that split beside zero point nine nine eight eight six three, and let the difference be the result.

DEFENSIBLE REPLY

There is no device column in the file. Forty-three columns, forty-one of them numeric NetFlow counters, and the mirror has stripped `IPV4_SRC_ADDR` and `IPV4_DST_ADDR` from Sarhan's published schema. You are prescribing a groups argument for a column that is not there.

ORDERED CORRECTION

Re-mechanise and merge with fault 9: the authoritative NF-ToN-IoT-v2 release carries the two IPv4 address fields, this mirror does not, and the notebook selected the mirror without checking — so the grouped split was foreclosed by an undisclosed provenance decision. That version is checkable. 'Group on device identity and split with `GroupShuffleSplit`' as written is refuted by opening the parquet.

You never print which forty-one columns the model sees

EVIDENCE

Cell eleven prints a count and no names: forty-one features. DROP is exactly four entries — Label, Attack, y and family — every one of them label-derived. Not a single identifier column is named. Compare notebook nine, whose DROP list explicitly names srcip, dstip, sport, dsport, Stime and Ltime, and notebook ten, which names StartTime, SrcAddr, DstAddr, Sport and Dport. Here the only other guard reads, quote, idlike equals columns where nunique is greater than zero point five times the length of X, end quote. On thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows that threshold is over six and a half million distinct values.

SCIENTIFIC CONSEQUENCE

That filter removes a primary key. It cannot remove host identity, because host identity on a testbed has a small number of distinct values and passes a six-and-a-half-million threshold untouched, is integer-encoded by the LabelEncoder two lines below, and becomes a feature. And here is the sentence that ends the argument: neither you nor I can tell from this PDF whether that happened, because the surviving column names are never printed anywhere in the notebook. A pipeline whose feature set cannot be enumerated from its own output is not auditable, and an unauditable pipeline cannot support a claim to six decimal places. The single-feature scan, the importance plot and the ablation all name individual columns while never naming the set they were drawn from.

REQUIRED REPAIR

Print the full sorted list of the forty-one feature names in cell eleven. Name every identifier column explicitly in DROP, as notebooks nine and ten do, instead of relying on a cardinality heuristic. And print which columns idlike and the constant filter removed, and why.

DEFENSIBLE REPLY

Open the parquet. One non-numeric column, Attack, and it is in DROP. There is no host identity to sail through your threshold because the mirror ships no address column at all.

ORDERED CORRECTION

Delete the 'host identity becomes a feature' paragraph — it describes a latent defect of the shared template, not an event here — and drop 'neither you nor I can tell', because the student can open the file in the room. What survives at CONTESTED: the feature list is never printed, so the model's inputs cannot be enumerated from the notebook's own output, and the DROP list names no identifier because the loader relies on a cardinality heuristic that on 13.1 million rows fires only above 6.6 million distinct values.

You cite Moustafa and Sarhan and download from a stranger

EVIDENCE

Section three names the source as Kaggle d-hoogla slash nftoniotv2 and the origin as ToN-IoT, Moustafa two thousand and twenty-one, re-featured to NetFlow v2 by Sarhan et al. two thousand and twenty-two — references one and two. Cell eleven downloads from that Kaggle reference. There is no hash check, no file listing, no row-count comparison against either author's release, and no record of which version of the mirror was pulled.

SCIENTIFIC CONSEQUENCE

The uploading account is neither crediting author. Every number in this notebook — thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows, forty-one features, a positive rate of zero point seven two five eight, and a duplicate rate of exactly zero on thirteen million NetFlow rows — is a property of that mirror, and not one of them has been checked against what Sarhan and colleagues published. A duplicate rate of exactly zero at that scale is itself a signal that the mirror may be pre-cleaned relative to the original release, and your provenance table does not say so, because nobody checked. In a course whose central argument is that provenance determines what a score means, this is the specific error being taught against.

REQUIRED REPAIR

Record the mirror's file hashes and row counts, compare them against the authors' published distribution, and print the comparison in the provenance table. If the authoritative release is unavailable, say so in those words.

You drew the precision-recall curve and never read it

EVIDENCE

Cell eighteen calls `precision_recall_curve` on the held-out probabilities and plots it as the third panel, titled Precision-Recall. That is its entire use. `average_precision_score` appears nowhere in the notebook. Nothing computes the area, prints it, or ranks a model on it. Cell sixteen sorts the table on `roc_auc` and the abstract leads with zero point nine nine eight eight six three.

SCIENTIFIC CONSEQUENCE

The correct criticism here is not that the diagnostic is missing — a student with the PDF open can see the curve. It is that the right diagnostic is on the page, as a picture, and was not read. And this notebook needs it more than its siblings, because the class balance is inverted relative to any deployment: seventy-two point five eight percent of the corpus is attack, so precision is inflated by the prior before the model does anything. The number that would show that, average precision computed at a realistic base rate, is the one you drew and did not compute.

REQUIRED REPAIR

Print `average_precision_score` with a bootstrap interval, and print precision at a fixed recall — say ninety-five percent — as a number. Then recompute both after re-weighting the held-out set to a stated deployment base rate.

DEFENSIBLE REPLY

You call the precision-recall panel the right diagnostic and in the same breath say precision is inflated by the prior. Both cannot be true. On my corpus attacks are the majority, so average precision on that labelling starts at 0.7258 before I do anything, and ROC-AUC is the metric that is more sensitive to a false positive here, not less — by a factor of about two and a half.

ORDERED CORRECTION

Re-mechanise to the form CORRECTION 2 prescribes: the PR panel is drawn with the majority class as positive and is therefore uninformative about the rare class, and the rare class here is the benign one, whose operating point — specificity, benign-class average precision, benign recall — is never established. Replace the REQUIRED's 'print `average_precision_score`' with 'print average precision with the labels inverted, and print specificity as a first-class number', which is what you correctly asked for in nb11's fault 12.

Ransomware at zero point nine zero four, with no denominator

EVIDENCE

Cell eighteen prints, quote, worst per-family recalls, ransomware zero point nine zero four, mitm zero point nine three four, scanning zero point nine eight nine, backdoor zero point nine nine seven, dos zero point nine nine eight, injection zero point nine nine nine, end quote. The loop reads for fam comma cnt in value_counts, and cnt is used only in the guard if cnt is less than five, continue — never printed. There is no bootstrap, no confidence interval and no binomial interval anywhere in the notebook. The only dispersion figure printed is the CV spread of plus or minus zero point zero zero zero one, which measures a different experiment.

SCIENTIFIC CONSEQUENCE

The conclusion says, quote, the weakest group sits at zero point nine zero four, which is where detection is thinnest, end quote. Ransomware and man-in-the-middle are the two smallest attack classes in the ToN-IoT testbed, which is why they are the two lowest — and without their denominators nobody can tell whether the gap between zero point nine zero four and zero point nine three four is real or whether both intervals span the whole range from zero point eight to one. Your stated honest evidence is the one quantity in the notebook with the least support and the only one with no error bar.

REQUIRED REPAIR

Print each family's held-out positive count beside its recall, with a Clopper-Pearson binomial interval. Then the sentence about which family is thinnest becomes a claim a reader can check rather than an ordering of six bare decimals.

DEFENSIBLE REPLY

Ransomware is 3,357 rows and man-in-the-middle 7,723, so the held-out denominators are about 839 and 1,931. The exact intervals on 0.904 and 0.934 are roughly 0.883 to 0.923 and 0.923 to 0.945. They do not overlap, and neither spans anything like 0.8 to 1. The ordering my conclusion relies on holds.

ORDERED CORRECTION

Withdraw the sentence about intervals spanning 0.8 to 1 and keep the true residual: no denominator and no interval is printed for any of the six, so the ordering the conclusion calls the honest evidence is unverifiable from the page even though it happens to hold. The version of this charge that is devastating lives in nb10, where Sogou's 0.188 is three detections out of sixteen.

Twenty thousand eight hundred and seventeen alerts at a default nobody chose

EVIDENCE

Cell eighteen reads pred equals pb greater than zero point five, and prints, quote, operational FALSE-POSITIVE RATE at zero point five equals zero point zero two three one, twenty thousand eight hundred and seventeen benign flagged of nine hundred thousand three hundred and twenty-one, end quote. The threshold appears in the printed label because it is the scikit-learn default, not because it was selected. There is no threshold sweep, no expected-cost calculation and no cost matrix in the twenty-seven cells.

SCIENTIFIC CONSEQUENCE

In intrusion detection the cost of missing a ransomware beacon and the cost of paging an analyst about a printer are not the same cost, and under asymmetric costs the operating point is the result. Ransomware's recall of zero point nine zero four — the number the conclusion calls thinnest — is ransomware's recall at zero point five. Lower the threshold and it rises, at a price in false positives you have not costed. You have already drawn a curve containing every alternative operating point and reported the single point the library handed you.

REQUIRED REPAIR

State a cost ratio explicitly, build the two-by-two cost matrix, select the threshold that minimises expected cost on a validation fold, and print it. Then report the false-positive rate and all six per-family recalls at that threshold, and show how far the choice moves if the ratio doubles.

DEFENSIBLE REPLY

The number in my abstract is a ROC-AUC and it is invariant to the threshold. The block you are attacking prints the threshold in its own label so no reader mistakes it for a chosen point.

ORDERED CORRECTION

Scope to the operating-point block and merge the remedy with fault 13, since one cost-model-and-threshold-selection cell answers both.

Two point three one percent, multiplied out

EVIDENCE

Cell eighteen prints zero point zero two three one — twenty thousand eight hundred and seventeen flagged of nine hundred thousand three hundred and twenty-one. Section two cites Sommer and Paxson two thousand and ten, reference seven, whose central argument is that base rates defeat network anomaly detection in the field. The citation is never applied to this notebook's own figure.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic you owed, with my assumption stated aloud so it can be checked: take a network carrying ten million benign flows a day — that number is mine, not yours, and the multiplicand must be the benign volume because a false-positive rate is a rate over negatives. Ten million times zero point zero two three one is two hundred and thirty-one thousand false alarms per day. Across a three-analyst shift that is seventy-seven thousand alerts each. The word operational in your printed line is doing work the number cannot support, and the paper that predicted exactly this outcome is in your reference list, uncited at the point of use.

REQUIRED REPAIR

Print alerts per analyst per day for a stated benign flow volume, in the results cell, right beside the false-positive rate. Then state the alert budget the deployment can absorb and read the required threshold off the precision-recall curve you already drew.

DEFENSIBLE REPLY

You supplied the ten million yourself and said so. The criticism that survives without your number is the one you already made in fault twenty.

ORDERED CORRECTION

Merge into fault 20. Sommer & Paxson's argument is about the base rate of attacks, not about false-positive rate times volume, so fault 20's re-weighting demand is the correct application of the citation and this is a weaker restatement of it.

The winner changes if you rank on accuracy

EVIDENCE

The cell-sixteen table: XGBoost roc-auc zero point nine nine eight eight six three, accuracy zero point nine nine zero six seven three. LightGBM roc-auc zero point nine nine eight six nine nine. Random Forest roc-auc zero point nine nine seven six seven zero, accuracy zero point nine nine two zero one one. So on ROC-AUC the order is XGBoost, LightGBM, Random Forest; on accuracy Random Forest is highest, ahead of XGBoost by zero point zero zero one three three eight. The selection line reads sort_values on roc_auc, take the first non-baseline row. No McNemar test, no bootstrap, no interval comparison appears anywhere.

SCIENTIFIC CONSEQUENCE

Everything downstream is computed on best: the confusion matrix, the false-positive rate of zero point zero two three one, all six per-family recalls, and every row of the ablation table. That model was chosen by a rule that would have picked a different model had the sort key been the other column already printed in the same table. And the gap that decided it — one hundred and sixty-four millionths between XGBoost and LightGBM — is the same order as the notebook's own cross-validation spread of plus or minus one ten-thousandth, so the selection is at best marginal even on its own metric.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of XGBoost and LightGBM, p-value printed. Or bootstrap the held-out AUC per model and print the intervals. State in the table that the ranking is metric-dependent and show both orderings.

Six recalls printed, and the total never stated

EVIDENCE

Cell eighteen builds the recall series and prints srt dot head of six. Six families appear: ransomware, mitm, scanning, backdoor, dos and injection. The same loop carries the guard if cnt is less than five, continue, and the skip count is never printed. No cell in this notebook prints the number of distinct values in the family column, and the section-one task description does not enumerate the ToN-IoT attack classes. So from the PDF alone there is no way to determine how many families exist, how many were scored, or how many were dropped.

SCIENTIFIC CONSEQUENCE

The conclusion elevates per-family recall to the honest evidence and then reports it without a denominator of any kind — neither positive counts per family nor a count of families. ToN-IoT ships more attack classes than six, and a class that fell below the five-positive guard and a class that simply scored well are indistinguishable from this page. An honest lead metric whose coverage cannot be established is not auditable, and the fix is one print statement.

REQUIRED REPAIR

Print every family with its held-out positive count and its recall, and print explicitly how many families were scored and how many the guard skipped, by name. A family below the guard should be printed as unscorable, never omitted.

DEFENSIBLE REPLY

The recall panel draws nine bars, one per scored family, because the truncation branch only fires above twelve. And cell thirteen plots the top eight attack families by count. Coverage is on the page twice, in two figures.

ORDERED CORRECTION

Strike 'there is no way to determine how many families exist, how many were scored, or how many were dropped' — the first two are visible in the figures. What survives is narrow and true: the printed text shows six of nine, the guard's skip count is never printed (it is zero — the smallest family, ransomware, has about 839 held-out positives), and no per-family denominator appears anywhere. That is the same residual as fault 11 and the two should be merged.

The longest packet in the flow is an MTU and a script

EVIDENCE

Cell twenty prints, quote, best single-feature AUC equals zero point seven six three seven, feature colon
 LONGEST_FLOW_PKT, end quote. That is the maximum observed packet length in the flow. And look one notebook across, at the sibling corpus sharing this exact schema: notebook eleven's cell twenty prints
 SRC_TO_DST_SECOND_BYTES — a byte rate — as its own strongest column at zero point eight eight two three. On both NetFlow-standardised corpora the leading feature is a size or rate statistic of the sending host.

SCIENTIFIC CONSEQUENCE

Packet size is set by the sending stack's MTU and the tool's payload construction. On a testbed, the machines running the attack scripts and the machines producing benign traffic are different machines with different stacks and different tools, so a size statistic separates the two populations by construction. An adversary changes it by padding, which costs nothing, requires no capability, and carries no detection risk. And because your ablation shows the score is completely unchanged when this column is removed, the alternative explanation is not that the model found something deeper — it is that the other forty columns carry the same testbed signature.

REQUIRED REPAIR

Partition the forty-one columns into size-and-rate features and structural features, and report the held-out AUC on the structural set alone. Then report both under notebook thirty-six's cross-corpus protocol, which is the only test that separates testbed signature from transferable signal.

DEFENSIBLE REPLY

0.7637 is my A grade. You said so yourself in the concession: it is the lowest single-feature AUC of the four notebooks and therefore the most honest. My shortcut check ran and reported that there is no shortcut.

ORDERED CORRECTION

Drop the shortcut framing, which the printed grade refutes, and keep the uncomputed experiment: no size-and-rate versus structural feature partition is reported, so nothing in the notebook separates testbed signature from transferable signal. The cross-notebook observation — that both NF-v2 corpora lead with a size or rate statistic of the sending host — is good and worth keeping.

Ransomware and man-in-the-middle, evaluated as if they did not adapt

EVIDENCE

There is no perturbation, no evasion budget, no adversarial example and no robustness experiment in the twenty-seven cells. The evaluation is one stratified random split of one static file. The attack classes named in cell eighteen include ransomware, man-in-the-middle, backdoor and injection — every one of them the work of an operator who reads detection research.

SCIENTIFIC CONSEQUENCE

Every feature carrying this score is under the attacker's direct control at zero cost: packet size through padding, duration through pacing, rate through throttling, flag patterns through connection shaping. None requires a new exploit. A security claim must report its worst case, and here the worst case is not merely unreported, it is uncomputed — on a corpus whose top feature an adversary defeats by adding bytes to a packet.

REQUIRED REPAIR

State a perturbation budget in feature space — for instance, every size and rate column may be scaled by any factor between zero point one and ten — and report the worst-case held-out recall under it. That number, not zero point nine nine eight eight six three, is the security claim.

DEFENSIBLE REPLY

My conclusion states that every number above measures in-distribution separability only. That is a scope limitation, not a caveat attached to a robustness claim I made.

ORDERED CORRECTION

Re-aim at the editorial gap the scope sentence does not close: the abstract leads with 0.998863 and the scope sentence is the last line of section 13.

A biased ranking, plotted as an explanation

EVIDENCE

Cell eighteen reads, if hasattr best feature_importances_, imp equals best dot feature_importances_ — the XGBoost built-in — and plots the top twelve under the title top importances slash absolute coefficient. permutation_importance does not appear in the notebook, and neither does SHAP.

SCIENTIFIC CONSEQUENCE

The NetFlow version two schema is dominated by high-cardinality continuous columns — byte counts, packet counts, durations, throughputs, size statistics — and impurity attribution systematically inflates exactly those relative to low-cardinality ones such as protocol or flag fields. So the ranking a student reads off that bar chart is partly an artifact of column cardinality rather than of predictive contribution, which matters here because the notebook's central question is which columns carry the score.

REQUIRED REPAIR

Permutation importance computed on held-out data, or SHAP values, printed alongside the impurity ranking so any divergence between them is visible.

DEFENSIBLE REPLY

My answer to which columns carry the score is the single-feature ROC-AUC sweep in section ten — model-free, computed on the data, not on the model's gain accounting. The importance chart is a diagnostic picture and it is cited nowhere in my ledger, my conclusion or my ablation.

ORDERED CORRECTION

Keep it at CONTESTED and narrow it to what is true: a biased ranking is plotted under a title a student will read as an explanation, with no permutation or SHAP comparison beside it to show the divergence. Drop 'the notebook's central question is which columns carry the score', because the notebook answers that question elsewhere with an unbiased estimator.

The robustness cell measures a different, smaller experiment

EVIDENCE

Cell twenty-four reads `cvX` comma `cvy` equals `Xtr` dot `iloc` up to forty thousand — and `Xtr` is by then the one-hundred-and-twenty-thousand-row subsample. So zero point nine nine eight two plus or minus zero point zero zero zero one is a three-fold estimate on forty thousand rows, which is four tenths of one percent of the training half of nine million eight hundred and fifty-one thousand nine hundred and ten. The four learners in cell sixteen carry fixed constructor arguments — `n_estimators` eighty, `max_depth` six — and no search object appears anywhere. Section twelve is titled reproducibility and robustness.

SCIENTIFIC CONSEQUENCE

The comparison in cell sixteen is therefore a comparison of four libraries' default settings, not of four learners, and the winner in fault fourteen was selected among untuned defaults. Separately, the plus or minus one ten-thousandth is the only dispersion figure in the entire notebook and it estimates the variability of a forty-thousand-row three-fold experiment, while being quoted in the abstract as though it characterised a three-million-two-hundred-and-eighty-three-thousand-row held-out score. Different sample size, different folds, different estimand. And it is the figure I had to compare against in fault fourteen, because it is the only one you provided.

REQUIRED REPAIR

Nested cross-validation on the reported pipeline: an inner loop that selects hyperparameters and an outer loop that estimates generalisation, both at the size actually reported. If that is too slow, shrink the experiment and say so — but do not present a forty-thousand-row three-fold spread as the robustness of the headline.

Seventy-two point five eight percent attack is a lab schedule

EVIDENCE

Cell eleven prints, quote, loaded thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows x forty-one features; positive rate zero point seven two five eight, end quote. Cell sixteen prints the majority baseline at the same figure, zero point seven two five eight, and presents it as the bar every model must clear. The held-out split contains two million three hundred and eighty-three thousand six hundred and fifty attack rows against nine hundred thousand three hundred and twenty-one benign ones. The notebook does acknowledge the simulated origin in section eleven and again in the conclusion, quote, it reflects how these flows were generated, end quote — and then reports every number at the testbed's balance anyway.

SCIENTIFIC CONSEQUENCE

Almost three flows in four being an attack is a fact about when the researchers ran their scripts. On a real network the ratio is inverted by three or four orders of magnitude. Every accuracy figure in the cell-sixteen table inherits that, including the sentence in your conclusion that no learner lands below the baseline — Logistic Regression clears zero point seven two five eight at zero point eight four eight one six nine, and at a realistic base rate it would not clear anything. And this is the master frame in one line: you named the generation artifact, in the conclusion, in those words, and then published the numbers that depend on it.

REQUIRED REPAIR

Re-weight or subsample the held-out set to a stated deployment base rate — one attack in a thousand flows, one in ten thousand — and print accuracy, precision, recall and the majority baseline at that rate. State the assumed rate in the abstract beside zero point nine nine eight eight six three.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 3,283,971 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Closing. So the ruling, and it is the one I most want the room to sit with. This notebook has the cleanest audit in the set — trust grade A, zero duplicates, zero contamination, the lowest single-feature AUC of the four, and the most careful literature review any of you have written. And it lands in the worst band, because the master frame does not care how good the audit is. You found Layeghy and Portmann, you quoted their finding that no model generalised across all four NetFlow corpora, you noted that the transfer test belongs in notebook thirty-six — and notebook thirty-six is in your repository, on your Kaggle reference, and it prints zero point three eight two two. Seven out of thirty. Reject with prejudice. An A on a two-question quiz is not a clean bill of health, and the sentence nothing here explains the score away is the one line in these thirty-six notebooks I would ask you all to strike.

NetFlow-Standardized Intrusion Detection: NF-UNSW-NB15-V2

One million nine hundred eighty-six thousand seven hundred forty-five NetFlow v2 records from the Kaggle upload dhoogla/nfunswnb15v2, being UNSW-NB15 re-featured into Sarhan and colleagues' standard forty-three-field schema, forty-one features, positive rate 0.0378.

HEADLINE

LightGBM, held-out ROC-AUC 0.999502, accuracy 0.996537 on four hundred ninety-six thousand six hundred eighty-seven held-out rows, printed in cell 16 and named the winning learner in the section-13 conclusion.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self D

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 2 Honesty / scope

- 22 live faults
- 6 critical (FATAL)
- 16 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me start with what is genuinely right here, because there is a great deal of it and this is the most careful of the four notebooks I was given. The exact-duplicate rate is 0.002 and the train-test exact-row contamination is 0.001, both measured and both printed — this corpus is clean of the memorisation artifact that ruins so many intrusion benchmarks. The majority-class baseline of 0.9622 is printed above the ranking table, printed again as a row inside it, and the conclusion explicitly guards the axis by saying that for ROC-AUC the trivial baseline is 0.5 rather than that figure. The single-feature audit reproduces exactly against an independent probe, and its grade of D holds across all eight reseeds. And the scope paragraph names the specific untested question rather than hedging generically. Almost every criticism I am about to make is a criticism the notebook itself anticipated in writing.

EXAMINER RULING

This notebook selected a corpus expressly because that corpus was built to make cross-dataset comparison possible, said so in its first section, identified train-on-one and test-on-another as the honest next step, declined to run it, and published 0.999502 anyway.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The corpus was chosen for transfer, and transfer was not tested

EVIDENCE

Section 1 states the reasoning: the standard feature set was proposed to enable cross-dataset comparison, so it is the right lens for asking whether high scores are feature shortcuts or transferable signal. The section-13 conclusion then states: because the schema was built for cross-dataset use, the honest next step is train-on-one, test-on-another — that is a test we did not run here, so we make no transfer claim. The scope paragraph adds: the split is random, not temporal or entity-grouped, and every number above therefore measures in-distribution separability only. Cell 16 nonetheless prints LightGBM at 0.999502 and the conclusion names it the winning learner.

SCIENTIFIC CONSEQUENCE

The notebook set up its own falsification test in section 1, in its own words, and then in section 13 recorded that it had not performed it. Every element required was present: the schema exists precisely so that four corpora share a feature space, the sibling corpora are named in the literature review, and the whole exercise is one additional load and one additional score. At doctoral level, naming the experiment that would settle your central claim and then not running it is not a limitation section; it is a confession that the claim was known to be unsupported at the time of publication. The number 0.999502 is not wrong. It answers a question the notebook itself says is not the question.

REQUIRED REPAIR

Load NF-BoT-IoT-v2 or NF-ToN-IoT-v2 — both named in your own literature review, both in the same forty-three-field schema, both a single Kaggle call away — fit on NF-UNSW-NB15-v2, score on the other, and print the transfer number in this notebook's own conclusion beside 0.999502. Both directions.

Atlas proof

docs/course/security-ml-notebooks/nb13-nf-unsw-nb15-v2.ipynb · cells 0, 2, and 25; zero-based manifest cell IDs: 0=9d33e009, 2=503b0f20, 25=2325450b

The notebook identifies the score as a generation artifact and headlines it

EVIDENCE

Cell 22 prints: headline as-is 0.999502; de-duplicated, zero percent of rows removed, 0.999532; shortcut feature dropped, meaning MIN_TTL removed, 0.999502. Identical to six decimals. The praxis pack's probe goes one column further: dropping the top two, MIN_TTL and TCP_WIN_MAX_IN, still leaves 0.9995, and the second column is worth 0.0000. Section 11 pre-commits to the correct reading — if the AUC barely moves that is not vindication, it means the classes are separable by many redundant features because the attack and benign distributions barely overlap, which is its own generation artifact. And the conclusion honours it: the separability is multi-feature, a property of how the flows were generated, not a one-header-field shortcut.

SCIENTIFIC CONSEQUENCE

Follow the notebook's own logic to its end. If dropping the two strongest columns costs nothing, then many columns each separate the classes on their own. Distributions that many independent variables separate that cleanly are distributions that barely overlap. Distributions that barely overlap are what you get when one set of hosts runs attack tooling and another set runs background traffic in a laboratory, and they are not what an operational network looks like. So the notebook has correctly diagnosed that 0.999502 measures the generation procedure — and has then printed 0.999502 as the winning learner's score, in the ranking table, in the ablation table and in the conclusion's validity ledger. The diagnosis and the headline cannot both stand.

REQUIRED REPAIR

State the consequence numerically rather than in prose: measure the class-conditional overlap directly — the Bhattacharyya coefficient or the total-variation distance between the attack and benign feature distributions — and report it. Then report the score on a behaviour-only feature set with the flow-header and counter columns removed as a block, and make that the headline, because it is the only number in the notebook that is not a measurement of the testbed.

Sommer and Paxson cited, their arithmetic never done

EVIDENCE

Cell 18 prints: operational FALSE-POSITIVE RATE at 0.5 equals 0.0026, one thousand two hundred thirty-one benign flagged of four hundred seventy-seven thousand nine hundred seventeen. Reference five in cell 26 is Sommer and Paxson, 2010, *Outside the Closed World*, and the literature review lists it as the closed-world critique. No cell multiplies 0.0026 by any flow volume, states an alerts-per-day figure, names an analyst budget, or computes precision at a realistic operational prior.

SCIENTIFIC CONSEQUENCE

Do the arithmetic on the page. A rate of 0.0026 means about two point six false alerts for every thousand benign flows, and about two thousand six hundred for every million. A single busy enterprise segment produces benign flows in the tens of millions per day, which puts this detector in the tens of thousands of false alarms daily. That is the closed-world result: a laboratory false-positive rate that reads as excellent becomes an unworkable alert volume the moment it meets production traffic, and the paper establishing it is in this notebook's own reference list. The word operational is applied to a number that has never been placed in an operational context.

REQUIRED REPAIR

State a flow volume for a plausible deployment, multiply through, and print alerts per analyst per day at threshold 0.5. Then invert the exercise: set an alert budget, read the required threshold off the precision-recall curve you are already plotting, and report recall at that threshold. That figure, not 0.999502, is what the detector is worth in a security operations centre.

Forty-one features, and no printed list of them

EVIDENCE

Cell 11 prints only a count — loaded 1,986,745 rows by 41 features. The feature names appear in exactly two places, both of them images: the top twelve importances plotted in cell 18, and the top fifteen single-feature AUCs plotted in cell 20. Even taking those two figures at their maximum and assuming no overlap at all, at least fourteen of the forty-one columns are named nowhere in the notebook — and in practice more than fourteen, because the two figures share their top entries, MIN_TTL among them. The loader silently drops columns in three separate steps — an identifier-like filter on high-cardinality object columns, a constant-column filter, and a regular-expression rename — and prints the outcome of none of them.

SCIENTIFIC CONSEQUENCE

Construct validity is the question of whether the features measure attacker behaviour or the generator's configuration, and it cannot be asked, let alone answered, without the list. Section 3 of this notebook raises exactly that concern — it says NetFlow header fields such as ports and protocol can act as shortcuts, which the audit probes — but the reader is never told which header fields survived into the model. Three filters ran, each capable of removing or retaining a decisive column, and none reported what it did. Every claim in the notebook about what the model learned is therefore unauditably from the notebook.

REQUIRED REPAIR

Print the full feature list after preprocessing, print the names of every column each of the three filters removed and why, and print the resulting count. Four lines. Without them, section 3's shortcut discussion is asserted rather than checkable.

Nine attack families, six named, and not one denominator

EVIDENCE

Cell 18 prints: worst per-family recalls, Analysis 0.668, Fuzzers 0.948, DoS 0.971, Exploits 0.986, Generic 0.994, Backdoor 0.995. The praxis pack's probe reports nine groups with zero skipped, so the count-five floor excluded nothing — this is a display choice, not the floor. The printed line is the worst six by construction, so the three families that never appear all sit at or above Backdoor's 0.995. In the same loop, `value_counts` computes the number of positives per family, that count feeds only the count-five gate, and it is never printed. The conclusion quotes all six recalls verbatim and says the weakest group sits at 0.668, which is where detection is thinnest.

SCIENTIFIC CONSEQUENCE

There are eighteen thousand seven hundred and seventy attack flows in the held-out set — four hundred ninety-six thousand six hundred eighty-seven minus the four hundred seventy-seven thousand nine hundred seventeen benign flows named on the false-positive line — divided across nine families. The conclusion nominates 0.668 as the honest lead number and gives it to three decimals with no denominator and no interval, so a reader cannot tell whether that figure rests on fifty positives or five thousand. And the three families reported nowhere are precisely the ones scoring at or above 0.995, which is the more diagnostic finding: a near-perfect recall on a small attack family is a signature of memorisation, and it is the one result this notebook does not show.

REQUIRED REPAIR

Print all nine families with recall, positive count and a Wilson binomial interval, sorted worst first, none omitted. A near-unity recall on a small family should be flagged as a memorisation candidate rather than silently excluded for being good.

DEFENSIBLE REPLY

Your own cited probe says nine groups, zero skipped — nine is not more than twelve, so the truncation branch never fires and my figure labels every one of the nine families, including the three you say appear nowhere; open the panel and read Worms, Reconnaissance and Shellcode off the y-axis.

ORDERED CORRECTION

Strike 'three are never named anywhere' and strike the inference paragraph beginning 'And the three families reported nowhere are precisely the ones scoring at or above 0.995' — they are plotted, at 1.00, 1.00 and 1.00, in the same cell. The surviving charge is narrower and correct: `'cnt'` is computed by `value_counts`, used only for the count-five gate, and never printed, so the conclusion's nominated lead number 0.668 carries no denominator. State the denominator you can derive: Analysis holds 770 of the corpus's 75,079 attack flows, roughly 193 in the held-out set, which puts a Wilson interval of about plus or minus 0.066 on 0.668.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

The strongest feature is the minimum time-to-live

EVIDENCE

Cell 20 prints: best single-feature AUC equals 0.9932, feature MIN_TTL. The provenance section notes that NetFlow header fields can act as shortcuts. The corpus originates in UNSW-NB15, which the reference list identifies as a laboratory capture from Moustafa and Slay, 2015, and in such a capture the attack traffic is generated by a small fixed set of hosts.

SCIENTIFIC CONSEQUENCE

Time-to-live is not a behaviour. It is an integer the sending stack initialises to its own default — sixty-four, one hundred and twenty-eight, or two hundred and fifty-five depending on the operating system — decremented once per router hop. So MIN_TTL encodes which machine sent the flow and how far away it sat, and in a testbed where attack tooling runs on a handful of designated hosts, that is very nearly a lookup of the source. An AUC of 0.9932 on that column is therefore consistent with the model having learned the laboratory's floor plan. And an adversary changes it with a single sysctl, at no cost and with no change to their objective.

REQUIRED REPAIR

Report the score with the whole block of stack-and-topology-derived fields removed together — TTL fields, window-size fields, and any address or port column — rather than dropping the single strongest one. That behaviour-only number is what generalises, and it belongs beside the headline.

DEFENSIBLE REPLY

Anomalous TTL is a recognised and legitimate intrusion signal in its own right — spoofed and remotely-originated traffic frequently carries distinctive TTL profiles — so a high single-feature AUC on MIN_TTL may reflect real detectable structure rather than a testbed fingerprint, and the notebook's ablation shows the headline does not depend on it either way.

Flows from one host on both sides of the split

EVIDENCE

Cell 16 calls `train_test_split` with `test_size 0.25`, a `random_state` and a `stratify` argument, and no `groups` argument. The section-13 scope paragraph states it directly: the split is random, not temporal or entity-grouped. The praxis pack's full-source scan reports `GroupKFold`, `GroupShuffleSplit` or any entity-grouped split present in zero of the thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

In a testbed capture, the flows are not independent draws. They come in bursts from a handful of machines running a handful of tools. A random split therefore places, for essentially every held-out flow, a training flow from the same host running the same tool in the same session. The model can match a neighbour instead of learning a concept, and 0.999502 is exactly what neighbour-matching produces. The corroborating evidence is already printed: the strongest single feature is `MIN_TTL`, which is close to an identifier of the sending host.

REQUIRED REPAIR

Group on the source host and use `GroupShuffleSplit` or `GroupKFold` so that no host appears on both sides, then report the drop from 0.999502. That difference is the quantity of scientific interest and the code is twenty lines.

DEFENSIBLE REPLY

Read the columns: the file I loaded has `L4_SRC_PORT` and `L4_DST_PORT` and no IPv4 address of any kind, so the group key you want me to split on is not in the frame — what you have found is that I chose a mirror that stripped it, which is a different charge and belongs against my loader, not my split.

ORDERED CORRECTION

Replace `REQUIRED`. The demand is: fetch Sarhan and colleagues' own NF-UNSW-NB15-v2 release, which retains `IPV4_SRC_ADDR` — the notebook's own download folder ships the schema dictionary proving the field exists — run `GroupShuffleSplit` on it, and report the drop. Then state in the abstract that the mirror selected for this notebook cannot support an entity-grouped protocol at all. Keep the corroborating observation that `MIN_TTL` is near-identifying; drop the implication that a host column was available and ignored.

The only account of the model's inputs is a reconstruction, not an observation

EVIDENCE

Cell 11 prints forty-one features and no names. The friendly praxis pack, auditing provenance, accounts for that number by arithmetic: the forty-three-field NetFlow v2 schema minus the two label columns, Label and Attack. That is the only statement in existence of which columns the model saw, and it is derived from a specification rather than read off the data. Note what follows if the arithmetic is the right explanation: nothing was removed by the loader's identifier-like filter, and Sarhan and colleagues' schema is headed by the source and destination IPv4 addresses and the layer-four ports.

SCIENTIFIC CONSEQUENCE

A reader wanting to know whether a laboratory capture's attacker addresses went into the model as features cannot find out. Not from the notebook, which prints counts and figures but no list; and not from the audit, which reconstructs the list by subtraction from a specification it did not verify against the file. The possibility that the address columns are model inputs — which would explain a held-out ROC-AUC of 0.999502 entirely, since the model would need only to learn which integers are attacker hosts, and would explain why dropping the top two features costs nothing — cannot be excluded by anything printed anywhere in this notebook. An unexcludable explanation for the headline is not a minor documentation gap.

REQUIRED REPAIR

Print the feature list, and if the address and port columns are present, drop them explicitly and report the refitted score. That is the single most informative experiment available on this corpus and it is one line in the drop list.

DEFENSIBLE REPLY

The uploader's cleaned parquet has forty-three columns, forty-one features plus Label and Attack, and the two features it drops relative to the published schema are the two IPv4 address fields — so the explanation you say cannot be excluded is excluded by opening the file, and what is left of your fault is fault five.

ORDERED CORRECTION

Re-mechanise rather than withdraw, because the port half of the charge survives and is stronger than the address half: L4_SRC_PORT and L4_DST_PORT ARE model inputs, and L4_SRC_PORT is the winning model's single largest importance in cell 18. Reword to: 'the notebook prints no feature list, so a reader cannot see that the two columns the model leans on hardest are the source and destination ports.' Delete the paragraph about address columns; it is answerable by one `pq.ParquetFile(...).schema_arrow.names``.

Attack sessions are episodes in time, split at random

EVIDENCE

Cell 16 splits at random with stratification and no time ordering. The conclusion states: the split is random, not temporal or entity-grouped. The NetFlow v2 schema carries flow start and duration fields, and the praxis pack's scan reports a temporal or time-ordered split present in zero of thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

An attack in a testbed is a burst lasting minutes. Cutting through the middle of a burst at random puts flows from the same execution of the same tool on both sides of the evaluation, so the model interpolates within an event rather than generalising to a new one. The deployment question — does a detector trained on last week's traffic recognise next week's — is never posed, and the notebook says so and leaves the number in place.

REQUIRED REPAIR

Order by the flow start field, train on the earlier portion and score on the later, and print the drop from 0.999502 beside it. Report both.

DEFENSIBLE REPLY

Name the column you want me to sort on. The NetFlow v2 schema has three duration fields and no clock; there is no flow-start timestamp in the file or in the feature dictionary that ships beside it, so the temporal protocol you are prescribing does not exist for this corpus.

ORDERED CORRECTION

Per CORRECTION 3 the prescription must go. Reword the REQUIRED to: no timestamp column exists in the loaded frame or in the shipped schema document, so the split cannot be validated as temporal by any means available here; state that unfixability in the abstract rather than in a scope sentence. Do not assert that no temporal split is constructible — if the parquet preserves capture order a positional holdout would be, and neither of us can verify the row order — but do not prescribe a sort key that is not there.

A winner declared on twenty millionths of ROC-AUC

EVIDENCE

Cell 16 prints LightGBM roc_auc 0.999502, XGBoost 0.999482, RandomForest 0.999338, LogisticRegression 0.988004, then the line best model: LightGBM. The gap between first and second is 0.000020. The gap across the whole tree cluster, first to third, is 0.000164. Cell 24 prints the winner's cross-validated standard deviation as plus or minus 0.0001 — five times the first-to-second gap. There is no McNemar test, no bootstrap, and no confidence interval anywhere in the notebook; the praxis pack's scan finds none in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

Every subsequent section of this notebook is computed for LightGBM and for nothing else: the confusion matrix, all nine per-family recalls, the operational false-positive rate, the ablation table, the cross-validation. All of that follows from a lead of twenty millionths. The conclusion is candid that the three tree learners cluster from 0.999338 to 0.999502, which is the right observation, and then names one of them the winner anyway.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the three tree models plus a bootstrap interval on each AUC. At these gaps the intervals will overlap, so report a three-way tie and state that the downstream model choice was arbitrary — or carry all three through section 9.

DEFENSIBLE REPLY

My conclusion does not claim LightGBM is better — it prints the cluster, 0.999338 to 0.999502, in the sentence you are quoting from; 'winning learner' is the label for which model carries the downstream diagnostics, and you have charged the same missing resampling distribution three times in faults eleven, thirteen and seventeen.

ORDERED CORRECTION

Merge with 13 and 17 into one FATAL: 'no interval, no paired test and no search anywhere, and the sole dispersion figure is computed on an undisclosed slice.' Keep 13 as the carrier because it has the specific, verifiable disclosure failure. Retain from this filing only the consequence sentence — every downstream number in sections 9 to 12 is computed for LightGBM alone on the strength of a 0.000020 lead.

The precision-recall curve drawn and never integrated

EVIDENCE

Cell 18 calls `precision_recall_curve` on the winner's probabilities and plots it as the third panel of the diagnostics figure. The function `average_precision_score` appears in no cell of this notebook, and the strings PR-AUC and average precision appear nowhere in any cell. Every reported and ranked figure — cell 16, cell 22, the validity ledger — is ROC-AUC.

SCIENTIFIC CONSEQUENCE

You plotted the right diagnostic and then declined to read it. At a positive rate of 0.0378 the negative class outnumbers the positive by more than twenty-five to one, and ROC-AUC is correspondingly insensitive to the false-positive count: it can sit at 0.999502 while one thousand two hundred and thirty-one benign flows are flagged, which is exactly what happened. Average precision is the statistic that is not blind to that, the curve containing it is drawn on the page, and the one function call that would extract it was not made.

REQUIRED REPAIR

Call `average_precision_score` on the probability vector already in scope in cell 18, print it, bootstrap its interval over held-out rows, and re-rank the four learners on it. Put that number in the validity ledger.

DEFENSIBLE REPLY

Average precision does not depend on my threshold, so it cannot show you the 1,231 alerts my 0.5 cut produced — and the panel you say I declined to read is printed on the page: precision sits at one until recall 0.95, which puts the number you want me to compute at about 0.995 against a ROC-AUC of 0.9995. It would not have changed a word.

ORDERED CORRECTION

Delete the clause 'it can sit at 0.999502 while one thousand two hundred and thirty-one benign flows are flagged, which is exactly what happened' — that is a threshold artifact and belongs in fault 14. The surviving charge is bare and correct: a curve was drawn and its area was never taken, on a corpus at a 26:1 imbalance, and the four learners are ranked on the wrong statistic. Note the contrast for the class: the same charge is devastating on nb16, where the printed PR panel bounds average precision near 0.30 against a ROC-AUC of 0.9486, and material on nb15 at roughly 0.85 against 0.9947. On nb13 it is immaterial, and the difference is exactly what a student should learn to read off the panel.

Six-decimal point estimates and an undisclosed cross-validation slice

EVIDENCE

Cell 16 rounds every ROC-AUC to six decimals. Cell 24 prints: LightGBM three-fold CV ROC-AUC equals 0.9993 plus or minus 0.0001. The line immediately above it in the same cell reads cvX, cvy equals Xtr dot iloc up to forty thousand — the folds come from the first forty thousand rows of the one hundred and twenty thousand-row training subsample, roughly two percent of the one million nine hundred eighty-six thousand seven hundred forty-five rows loaded. That bound appears nowhere: not in the print line, not in section 12, not in the validity ledger. Notebook 14 in this very series states the identical bound explicitly in both its section 12 and its conclusion, so the disclosure exists in the template and is missing here.

SCIENTIFIC CONSEQUENCE

A reader given 0.999502 and 0.9993 plus or minus 0.0001 with no scope statement will read the second as corroborating the first across the corpus. It does not. It measures fold-to-fold stability of a fit on a two percent training slice, with duplicates left in and held-out rows excluded. And the plus or minus 0.0001 is the only dispersion number in the notebook, so it is doing rhetorical work — conveying stability — that its actual scope does not license. The sibling notebook proves the author knew how to say this.

REQUIRED REPAIR

Bootstrap the held-out AUC over resampled test rows and report that interval. Report every figure to the resolution the interval supports rather than to six decimals. And carry the forty-thousand-row bound into the printed cross-validation line, exactly as notebook 14 does.

The operating point is whatever the library defaulted to

EVIDENCE

Cell 18 sets pred equal to pb greater than 0.5, a literal, and every operational figure in section 9 — the confusion matrix, the false-positive rate of 0.0026, all nine per-family recalls including the 0.668 the conclusion nominates — is computed at that point. There is no cost matrix, no expected-cost minimisation and no threshold sweep in the notebook. The praxis pack's scan reports an explicit cost matrix in zero of thirty-six notebooks and threshold selection from a precision-recall curve in zero of thirty-six.

SCIENTIFIC CONSEQUENCE

A missed intrusion is a compromise; a false alarm is analyst time, and at two thousand six hundred per million benign flows it is a great deal of analyst time. When the costs are that asymmetric the operating point is not an evaluation detail, it is the result — and every number the conclusion calls operational is a consequence of a threshold nobody chose for this problem. The recall of 0.668 on the Analysis family in particular is a function of the threshold, not a property of the model.

REQUIRED REPAIR

State a cost model or an alert budget, sweep the threshold across the held-out set, plot expected cost or alert volume against threshold, and report recall, precision and per-family recall at the operating point that satisfies it. Report 0.5 only as a reference point.

DEFENSIBLE REPLY

Faults four, fourteen and sixteen all end with the same instruction: choose the operating point from the precision-recall curve. That is one experiment I did not run, not three.

ORDERED CORRECTION

Fold into fault 4 as its threshold-sweep clause. Keep the distinctive observation that the 0.668 Analysis recall the conclusion nominates as its honest lead is a function of the threshold rather than a property of the model — that sentence does work fault 4 does not do.

One attack in twenty-six, and nothing weighted

EVIDENCE

Cell 11 prints positive rate 0.0378, which is about one attack flow in twenty-six. Cell 16 constructs LogisticRegression with max_iter three hundred, RandomForestClassifier with sixty trees, XGBClassifier with eighty rounds and depth six, and LGBMClassifier with eighty rounds. None carries a class_weight or scale_pos_weight argument. The praxis pack's full-source scan confirms class weighting present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

An unweighted loss is minimised by getting the majority right, and the majority here is benign. The consequence is printed: the Analysis family sits at a recall of 0.668, meaning about a third of that attack family is missed at the reported operating point, and the smallest families are precisely the ones an unweighted objective is cheapest to abandon. That outcome was engineered by declining a single constructor keyword.

REQUIRED REPAIR

Refit with class_weight balanced for the forest and linear models and scale_pos_weight set to the negative-to-positive ratio, roughly twenty-five to one, for the boosted pair, and re-report every per-family recall. If Analysis moves materially off 0.668, the weakest-family finding was an artifact of the loss weighting rather than of the data.

DEFENSIBLE REPLY

My training subsample carries about 4,500 attack flows and my overall recall on the held-out set is 0.974 — that is not a loss function that abandoned the minority class. The Analysis family sits at 0.668 because it holds 770 flows in a corpus of two million, which is a rare-family problem, not a class-weight problem, and reweighting the whole positive class does not address it.

ORDERED CORRECTION

Strike 'The consequence is printed: the Analysis family sits at a recall of 0.668... That outcome was engineered by declining a single constructor keyword.' Replace with the true and defensible form: no weighting was tried and no evidence is offered either way, so the reader cannot tell whether the rare-family recalls are a property of the corpus or of the objective — which is the question fault 3's missing denominators would also have helped answer.

Uncalibrated boosted scores cut at one half

EVIDENCE

Cell 18 takes best dot predict_proba and thresholds it at 0.5, where best is LightGBM fitted on a twenty-five-to-one class ratio. There is no CalibratedClassifierCV, no isotonic fit, no Platt scaling, no reliability diagram and no Brier score anywhere in the notebook. The praxis pack's scan reports probability calibration present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

Boosted trees fitted under class imbalance emit scores that are monotone in risk but not equal to risk, and the imbalance systematically pushes them toward the majority. So 0.5 sits at an unknown point on the real risk scale, and every operational quantity in section 9 inherits that arbitrariness. It also means the score cannot be used for triage ranking or combined with any other signal, which are the two things an operational detector's output is actually for.

REQUIRED REPAIR

Calibrate on a held-out calibration split with isotonic regression, print the reliability diagram and the Brier score before and after, and choose the operating point on the calibrated scale from the precision-recall curve you are already drawing.

DEFENSIBLE REPLY

This is the third time you have asked me to pick a threshold off the PR curve.

ORDERED CORRECTION

Retain only the residual that faults 4 and 14 do not carry: an uncalibrated score cannot be used for triage ranking or fused with another signal, which are the two things an operational detector's output is for. That is a genuinely separate loss and it is one sentence.

Four fixed configurations, no search, and a winner declared

EVIDENCE

Cell 16 fixes sixty trees for RandomForest, eighty rounds and depth six for XGBoost, eighty rounds for LightGBM, and three hundred iterations for the linear model. There is no GridSearchCV, no RandomizedSearchCV, no Optuna and no nested cross-validation in any cell. The praxis pack's full-source scan reports hyperparameter search present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

The declared winner leads the runner-up by 0.000020. Ten additional boosting rounds, or a single change of learning rate or leaf constraint, would move either model by more than that many times over. So the sentence best model: LightGBM records which integers were typed into cell 16, not which algorithm is better on this corpus, and the ranking table ranks defaults rather than methods.

REQUIRED REPAIR

Nested cross-validation: an inner loop selecting each learner's hyperparameters and an outer loop estimating generalisation, so that each algorithm is reported at its own best configuration.

DEFENSIBLE REPLY

Faults eleven, thirteen and seventeen are one missing pass — nested cross-validation with bootstrap intervals — filed three times.

ORDERED CORRECTION

Merge into 13. Keep the one-line consequence: at a 0.000020 lead, 'best model: LightGBM' records which integers were typed into cell 16.

A grade of D bracketed by two cutoffs it nearly touches

EVIDENCE

Cell 20 prints: best single-feature AUC equals 0.9932, and data trust grade D. The grading line in the same cell sets F at or above 0.999, D at or above 0.99, and C at or above 0.95. The praxis pack's probe reseeds the sixty-thousand-row audit sample eight times: the values run from 0.9919 to 0.9946. The low end sits only 0.0019 above the 0.99 boundary and the high end only 0.0044 below the 0.999 boundary. Both margins are inside the pack's own 0.01 proximity tolerance, and the pack scored this criterion one half out of one for exactly that reason.

SCIENTIFIC CONSEQUENCE

The letter is the object a reader retains, and the conclusion leans on it — it says a D or F means the headline is optimistic and to treat it as a benchmark number rather than a deployment estimate. But the letter is produced by a threshold whose distance from the observed range is smaller than the spread the range itself exhibits. A modest change in the corpus, the sample size, or the feature set would move the letter to C or to F, and none of those three letters would tell the reader anything different about the corpus.

REQUIRED REPAIR

Report the eight-seed interval, 0.9919 to 0.9946, rather than a letter, and state its distance from each boundary. If a letter must be retained, widen the bands until they exceed the resampling spread.

DEFENSIBLE REPLY

Eight seeds, eight D's, and the spread is narrower than my distance to the nearest cutoff — your own probe records both facts. That is a stable grade, and you are quoting its stability as evidence of fragility.

ORDERED CORRECTION

Re-mechanise to the form CORRECTION 6 licenses: a hard cutoff applied to a point estimate with no interval is not a measurement, whatever letter it happens to return — which is a criticism of the grading instrument used across all thirty-six notebooks, not of this D. Note the contrast the class should see: on nb14 and nb16 the same instrument does flip under reseeding, and there the E3 charge is fully assertable; here it is not.

A two-hop provenance chain with no integrity check

EVIDENCE

The provenance table names the source as Kaggle dhoogla/nfunswnb15v2 and the origin as UNSW-NB15 re-featured to NetFlow v2. The literature review credits the re-features to Sarhan, Layeghy and Portmann, 2022, and the underlying capture to Moustafa and Slay, 2015. The uploading Kaggle account is neither of those parties. Cell 11 downloads the upload and reads whatever parquet files it finds, with no SHA verification, no row-count comparison against Sarhan's published figures, and no schema comparison. Note also that the loader declares NROWS as one million five hundred thousand with the comment per-file read cap, memory bound, and then never passes it to read_parquet — the declared cap is dead code and the whole file is loaded.

SCIENTIFIC CONSEQUENCE

Every number in this notebook is a claim about what a third party uploaded. UNSW-NB15 is a capture; NF-UNSW-NB15-v2 is a re-derivation of features from that capture by different authors; and dhoogla's upload is a further transformation of unknown extent — the uploader's own description of these datasets is that they are cleaned. Attributing 0.999502 to NF-UNSW-NB15-v2 and to Sarhan and colleagues, as the abstract and provenance table do, attributes it to an object this notebook never verified it was holding.

REQUIRED REPAIR

Print a SHA-256 of the downloaded files, and print the row count, column list and positive rate beside Sarhan and colleagues' published figures for NF-UNSW-NB15-v2. State exactly which of those match. Also remove or apply the dead NROWS constant, because a declared memory bound that never fires is a reproducibility hazard for anyone running this on a smaller machine.

The detector's worst case is unmeasured, and its strongest feature is free to change

EVIDENCE

There is no perturbation experiment, no stated evasion budget and no worst-case recall in any cell. Cell 20 identifies MIN_TTL as the strongest single feature at an AUC of 0.9932. The reference list carries Sommer and Paxson 2010, whose argument is that the operational environment is adversarial and non-stationary in ways laboratory evaluation does not capture.

SCIENTIFIC CONSEQUENCE

The reported recalls describe an attacker who is not trying to evade, because the attacks in UNSW-NB15 were generated by tooling run to produce labelled data. Under those maximally favourable conditions the weakest family already sits at 0.668. What that becomes against an opponent who sets an initial time-to-live, pads packets, throttles rate, or randomises inter-arrival timing is the number that would justify deployment, and it does not exist anywhere in the notebook.

REQUIRED REPAIR

Define a perturbation budget over the flow features an attacker controls — time-to-live, packet size distribution, inter-arrival timing, window size — regenerate the held-out attack rows under that budget, and report worst-case per-family recall. That figure is the detector's adversarial value and it is the number that should sit beside 0.999502.

DEFENSIBLE REPLY

You are asking for a perturbation experiment in flow-feature space that no notebook in this series runs and that my section 13 never claims to have run. If your point is that my per-family recalls are ceilings rather than estimates, say that — it is one sentence and I will add it.

ORDERED CORRECTION

Re-aim at what the notebook does claim. The defensible charge is not 'no evasion experiment' — no notebook in the series has one — but that the per-family recalls in section 9 are reported without the qualifier that they were measured against scripted tooling with no evasion objective, so 0.668 is an upper bound and is quoted as a point. That is one sentence in the ledger and it is the honest version of this fault.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 496,687 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Eight out of thirty, and this one lands in REJECT rather than the bottom band, which I want to explain rather than leave as an arithmetic accident. The bottom band is for a reported quantity that is an artifact of the file, and this file is clean — a duplicate rate of 0.002 and a contamination of 0.001, both measured, both reproduced by an independent probe. The headline here is not an artifact of the file. It is an artifact of the corpus: of a laboratory in which one set of hosts ran attack tooling and another ran background traffic, and whose separability the notebook itself correctly identifies as multi-feature and generation-derived. That is the definition of the REJECT band and it is precisely where this belongs. If anyone wants to press me on the single point that separates this from its siblings, it is claim reproducibility, where I have given two rather than one: the audit here does survive re-derivation across eight reseeds and across three refitted ablation settings, which its sibling notebook 14 does not. That is partial execution of the right method, and it earns the point. It does not earn more, because the claim was never re-derived under a defensible protocol — only under the same random split, perturbed. And the reason none of this is a pass is one sentence in section 13: that is a test we did not run here. The test was named, the corpus was chosen for it, the sibling data was one call away, and it was not run.

IoT Attack Detection on CICIoT2023

One million forty-eight thousand five hundred seventy-five flow records pulled from the Kaggle upload subhajournal/iotintrusion, credited to Neto and colleagues 2023, forty-one features, positive rate 0.9767 — attack is the majority class here and benign is the rare one.

HEADLINE

RandomForest, held-out ROC-AUC 0.999394, accuracy 0.996513, printed in the cell-16 ranking table and named the winning learner in the section-13 conclusion.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9

/10 · self D

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 9 critical (FATAL)
- 12 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook is careful in ways worth naming. Its abstract states before anything else that the published CICIoT2023 corpus is about forty-seven million flows and that the Kaggle mirror used here provides a subset, and that every number below describes the subset rather than the corpus. It is the only notebook in my set that carries the cross-validation's scope into both section 12 and the conclusion: it says explicitly that the folds are cut from the first forty thousand rows of the one hundred and twenty thousand-row training subsample, about three point eight percent of what was loaded, and that the figure therefore reports fit stability on a slice and not generalisation. Section 11 states in advance that an ablation which barely moves is not vindication but its own generation artifact, and the conclusion honours that reading rather than reversing it. That is discipline, and it is rare.

EXAMINER RULING

The row count of this corpus is exactly two to the twentieth, minus one — a software boundary, not a sampling design — and every quantity in this notebook, above all the class balance that defines the task, is conditioned on wherever some file-writer stopped.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The row count is a limit, not a sample

EVIDENCE

Cell 11 prints: loaded 1,048,575 rows by 41 features. Two to the twentieth is one million forty-eight thousand five hundred seventy-six. Subtract one for a header line and you have exactly the printed figure. That number is also the maximum row count of a Microsoft Excel worksheet. Now look at the loader in the same cell: NROWS is set to one million five hundred thousand and passed as the nrows argument to read_csv. One million forty-eight thousand five hundred seventy-five is well below one million five hundred thousand, so that cap never bound — it is inert, and it did not produce this number. Section 3 and the abstract both describe the result as a Kaggle subset of the roughly forty-seven million published corpus.

SCIENTIFIC CONSEQUENCE

A data row count that lands exactly on a known software limit minus a header is not the output of a sampling design. It is the output of a boundary. Which boundary — a spreadsheet save, a head command, a chunked writer — this notebook does not know, because it never checked. And the word subset does real work in the abstract: it invites the reader to treat these rows as a draw from CICIOT2023 whose properties approximate the whole. That is a claim about a sampling process, and no sampling process has been identified. The consequence is not confined to the row count. It propagates to the positive rate of 0.9767, the majority-class baseline of 0.9767, the duplicate rate of 0.063, every per-family recall, and the six-thousand-one-hundred-and-nineteen-row benign population on which the operational false-positive rate is estimated. There is a further tell in the split geometry: cell 16 prints a held-out size of two hundred sixty-two thousand one hundred forty-four, which is exactly two to the eighteenth, because a quarter of two to the twentieth minus one rounds to precisely that. The power of two runs all the way through the experiment.

REQUIRED REPAIR

Fetch the authoritative CIC distribution of CICIOT2023 and compare: total record count, benign fraction, and the count for each of the thirty-three attack types, printed side by side with the mirror's. If they agree, you have converted an assumption into a measurement. If they do not, the mirror is a truncation and every number in this notebook must be recomputed on a corpus-wide random sample. At minimum, publish a checksum of the downloaded file against the upstream object.

Sommer and Paxson cited, and their arithmetic never done

EVIDENCE

Cell 18 prints: operational FALSE-POSITIVE RATE at 0.5 equals 0.0699, four hundred twenty-eight benign flagged of six thousand one hundred nineteen. Reference three in cell 26 is Sommer and Paxson, 2010, Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. The literature review in cell 2 lists it as the closed-world critique. No cell anywhere multiplies 0.0699 by a flow volume, states an alerts-per-day figure, or names an analyst budget.

SCIENTIFIC CONSEQUENCE

Do the arithmetic on the page. A false-positive rate of 0.0699 means about seventy false alerts for every one thousand benign flows, and about sixty-nine thousand nine hundred for every million. An IoT segment of any size produces benign flows in the millions per day. This detector, at the operating point the notebook itself reports, would generate tens of thousands of false alarms daily against a population of analysts numbering in single digits. That is the closed-world result Sommer and Paxson published in 2010 and this notebook lists in its own references: a laboratory false-positive rate that looks small becomes an unusable alert volume the moment it meets real traffic. The citation is present. The calculation is not.

REQUIRED REPAIR

State a flow volume for a plausible deployment — flows per hour on the segment — multiply through, and print alerts per analyst per day at the reported threshold. Then set an alert budget, work backwards through the precision-recall curve you are already drawing to the threshold that meets it, and report recall at that threshold. That number, not 0.999394, is the deployable capability.

Thirty-three attack families measured, twenty-one never shown

EVIDENCE

The praxis pack's probe reports thirty-three groups with zero skipped, so the count-five floor in cell 18 excluded nothing — I want to be precise, this is a display truncation and not the floor. Read the code. The plotted series is built as `pd.concat([srt, dot, head, nine, and, srt, dot, tail, three])` whenever there are more than twelve groups, so the figure shows twelve of thirty-three. The printed line is `srt, dot, head, six`, so the text shows six, and those six are a subset of the nine already in the figure. Thirty-three minus the twelve that are plotted leaves twenty-one families whose recalls were computed and then discarded from every output. The six that print are Recon-OSScan 0.73, SqlInjection 0.821, Recon-PortScan 0.875, DictionaryBruteForce 0.886, DNS_Spoofing 0.894 and BrowserHijacking 0.895. And in the same loop, `value_counts` computes the number of positives per family, uses it solely for the count-five gate, and never prints it.

SCIENTIFIC CONSEQUENCE

The notebook's own section 3 says the honest metrics are per-family recall and the shortcut audit, not aggregate accuracy. It then withholds two thirds of the per-family recalls. A reader cannot tell whether the twenty-one unshown families are the easy floods or further weak spots, and cannot evaluate any of the six that are shown, because a recall of 0.73 on twenty positives and a recall of 0.73 on twenty thousand are different claims and the denominator was deleted. The quantity the notebook nominates as its honest lead is the quantity it prints least completely.

REQUIRED REPAIR

Print a table of all thirty-three families with recall, positive count and a Wilson binomial interval on each. Sort worst-first, show every row, and mark as unscorable any family whose interval is wider than half the unit interval rather than reporting a bare number.

The data-trust grade is decided by six ten-thousandths and flips on reseeding

EVIDENCE

Cell 20 prints: best single-feature AUC equals 0.9906, feature `rst_count`, and then single-feat grade D and data trust grade D. Read the grading line in the same cell: the letter is D at or above 0.99 and C below it. The printed value is 0.9906, so the margin is 0.0006. The praxis pack's probe reseeds the sixty-thousand-row audit sample eight times: the values span 0.9889 to 0.9922, and seeds five and six, at 0.9895 and 0.9889, land below the cutoff and grade C. The pack scored this criterion zero out of one and called the published D a sampling accident. Note the direction, because it is counter-intuitive: a lower measured value gives a better letter on this scale.

SCIENTIFIC CONSEQUENCE

The letter is the object a reader carries away — the notebook's own summary line describes it as the data-trust grade — and it is decided here by which sixty thousand of one million forty-eight thousand five hundred seventy-five rows the sampler happened to draw. Two of eight seeds return a different letter. A grade that flips on a resample is not a property of the corpus; it is a property of the draw, and reporting it as a grade attaches a stable-looking label to an unstable quantity.

REQUIRED REPAIR

Report the single-feature AUC as an interval across seeds — 0.9889 to 0.9922 is already computed — and drop the letter, or widen the grade bands until they exceed the resampling spread. A cutoff at 0.99 on a quantity with a 0.0033 seed range cannot discriminate.

The detector saw about two thousand eight hundred examples of normal

EVIDENCE

Cell 11 prints positive rate 0.9767. Cell 16 prints: trained on 120,000 stratified subsample, held-out 262,144. The split is stratified, so the training subsample contains about two thousand eight hundred benign rows — that is one hundred and twenty thousand times one minus 0.9767, which is 0.0233. Cell 18 confirms the held-out side directly: the false-positive line names six thousand one hundred nineteen benign flows in a held-out set of two hundred sixty-two thousand one hundred forty-four, and four hundred twenty-eight of those six thousand one hundred nineteen are flagged.

SCIENTIFIC CONSEQUENCE

A network intrusion detector's job is to recognise normal and object to departures from it. This model has seen fewer than three thousand examples of normal. Worse, the direction of the evaluation is inverted relative to deployment: on a real IoT network benign traffic is the overwhelming majority and attack is rare, so the quantity that governs whether the system is usable is precisely the false-positive rate on benign traffic — and that is the one quantity this corpus can barely estimate, because it contains almost no benign traffic to estimate it from. Six thousand one hundred nineteen negatives is a thin denominator for a rate that will be multiplied by millions of flows a day. The headline of 0.999394 is dominated by the ninety-seven point seven percent of rows the model finds easy.

REQUIRED REPAIR

Report the benign counts explicitly in train and test on the printed line. Then re-weight the held-out evaluation to a realistic operational prior — state a plausible attack prevalence for a production IoT segment, reweight the confusion matrix to it, and report precision at that prior. Also report a binomial confidence interval on the false-positive rate, because on six thousand one hundred nineteen trials it is not a point.

DEFENSIBLE REPLY

My false-positive rate divides false alarms by negatives only, so the 0.9767 prior never touches it; on 6,119 trials the ninety-five percent interval on 0.0699 is plus or minus 0.006, which is narrower than the resolution I print; and I print that denominator on the same line you are quoting — '428 benign flagged of 6,119'. What is left of your fault is the operational-prior reweighting, which is fault three.

ORDERED CORRECTION

Re-mechanise to the training side, which is the part that is real and unanswerable: a stratified 120,000-row subsample at a 0.9767 positive rate carries about 2,796 benign examples, so a detector whose job is to model normal was fitted on fewer than three thousand instances of it, and that number is printed nowhere. Delete the 'estimated on a population that is 97.7 percent attack' framing and delete the binomial-interval demand, which the arithmetic does not support. Move the operational-prior reweighting into fault 3.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

The uploading account is not the crediting author, and nothing was verified against upstream

EVIDENCE

Cell 11 sets REF to subhajournal slash iotintrusion and downloads it. The provenance table in section 3 names the source as that Kaggle upload and the origin as CICIOT2023, and the literature review credits Neto and colleagues 2023. Neto and colleagues did not upload subhajournal slash iotintrusion. The notebook computes no checksum, no row-count comparison, and no schema comparison against the Canadian Institute for Cybersecurity's own distribution.

SCIENTIFIC CONSEQUENCE

Every claim in this notebook is a claim about whatever a third party uploaded and labelled. That party may have subsetted, re-ordered, re-typed, re-labelled, deduplicated or truncated — and given the row count in fault one, at least one of those things demonstrably happened. Attributing the resulting numbers to CICIOT2023 and to Neto and colleagues, as the abstract and provenance table do, attributes them to a corpus this notebook never touched.

REQUIRED REPAIR

Download from the CIC's own distribution, or, if the mirror must be used for size reasons, print a SHA-256 of the mirror file alongside the row count, column list and per-label counts of the authoritative release, and state exactly which of those match.

DEFENSIBLE REPLY

Faults one, six and seven all ask me for the same download and the same comparison table.

ORDERED CORRECTION

Fold into fault 1. Retain only the residual fault 1 does not carry: the abstract and provenance table attribute these numbers to CICIOT2023 and to Neto and colleagues, who did not upload `subhajournal/iotintrusion`, so the attribution is to an object the notebook never verified it was holding.

Dropping the strongest feature raises the score

EVIDENCE

Cell 22 prints the ablation table: headline as-is 0.999394, de-duplicated with six percent of rows removed 0.999367, shortcut feature dropped, meaning rst_count removed, 0.999441. Dropping the strongest single feature moved the score up by 0.000047. The praxis pack's probe reproduces the same shape at four decimals — base 0.9996, drop rst_count 0.9996, drop rst_count and Weight 0.9996 — with the second column worth minus zero. The conclusion reports this as: the ablation refutes a single-feature story.

SCIENTIFIC CONSEQUENCE

An instrument that returns a higher reading after you remove the thing it is measuring is not measuring it. Each ablation row re-splits and refits, and the notebook says so, so 0.000047 is noise. But then the whole table lies inside its own noise band: the three settings differ by at most 0.000074 across the board. The ablation therefore cannot distinguish a shortcut from no shortcut on this corpus at all, and the conclusion's word refutes overstates what an instrument with this resolution can do. To the notebook's credit, section 11 already says that a score which barely moves is not vindication but its own generation artifact, and the conclusion honours that. The defect is that the instrument is then reported as having decided something.

REQUIRED REPAIR

Run each ablation setting over at least ten seeds and report mean and standard deviation per row. If the standard deviation exceeds the between-row differences — and on this evidence it will — state that the ablation is uninformative at this corpus's separability level, and replace it with a test that has room to move, such as scoring on behaviour-only features with the whole block of flow-header counters removed.

The strongest feature counts a flag the tooling emitted

EVIDENCE

Cell 20 prints: best single-feature AUC equals 0.9906, feature `rst_count`. The abstract describes the corpus as thirty-three attack types against a large IoT testbed of one hundred and five devices, and the provenance section notes in the notebook's own words that flood attacks dominate and are trivially separable. No cell reports the score on a feature set with the flow-header counters removed as a block.

SCIENTIFIC CONSEQUENCE

A reset count of a particular magnitude is what you get when one specific flooding tool, run from one specific host, meets one specific device's TCP stack under one specific network condition. It is a fingerprint of the laboratory apparatus. An adversary using a different tool, or throttling, or spoofing, produces a different reset profile at no cost and with no change to their objective. So the near-sufficiency of this column measures how the corpus was produced, not what an attacker does.

REQUIRED REPAIR

Ablate the flow-header and flag-counter columns as a block, not one at a time, refit, and report the resulting AUC as the behaviour-only score. Publish both numbers.

DEFENSIBLE REPLY

In a genuine TCP reset flood or a SYN flood the reset count is the attack — it is not an incidental artifact of tooling but the literal signature of the technique — so a high single-feature AUC on `rst_count` may be exactly the real signal an intrusion detector should be using.

10

CRITICAL MEASURED VERIFIED EXACT DUPLICATES ONLY

Six percent exact duplicates counted, near-duplicates never looked for

EVIDENCE

Cell 20 prints: exact-duplicate row rate, whole corpus, equals 0.063, and train-test exact-row contamination equals 0.017. The contamination is computed by rounding the feature matrix to six decimals and testing tuple membership in a set built from training rows, so a single differing digit in any of forty-one columns makes two rows count as unrelated. There is no nearest-neighbour distance computation anywhere in the notebook. The provenance section states that flood attacks dominate this corpus.

SCIENTIFIC CONSEQUENCE

Six point three percent exact duplicates is already substantial, but exactness is the wrong criterion for flood traffic. A flood emits thousands of flows from the same source, in the same second, to the same destination, differing by a byte count here or a microsecond there. Under an exact-match test those are all distinct rows, and they distribute freely across the split. That is the leakage that produces a 0.999394 and it is the leakage this audit is constructed not to see. The reported contamination of 0.017 is therefore a lower bound of unknown tightness on the quantity that matters.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from each held-out row to its closest training row, in a scaled feature space, and report the fraction below a small threshold. Then refit after removing held-out rows within that threshold and report the drop. That is the near-duplicate contamination and it is the number the headline depends on.

12

CRITICAL MEASURED VERIFIED MODEL RANKING WITHOUT A TEST

A winner named on a gap of thirty-three millionths

EVIDENCE

Cell 16 prints: RandomForest roc_auc 0.999394, XGBoost 0.999361, LightGBM 0.999336, LogisticRegression 0.993944, and then the line best model: RandomForest. The gap between first and second is 0.000033. The gap from first to third is 0.000058. Cell 24 prints the winner's cross-validated standard deviation as plus or minus 0.0001, which is three times the first-to-second gap. There is no McNemar test, no bootstrap and no confidence interval anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every downstream section of this notebook — the confusion matrix, the per-family recalls, the operational false-positive rate, the ablation, the cross-validation — is computed for RandomForest and for no other model, because RandomForest was declared the winner. That declaration rests on thirty-three millionths of ROC-AUC. Had the seed differed, a different model would have carried every subsequent number in the notebook, and nothing in the notebook tells the reader that.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top three models, plus a bootstrap interval on each model's AUC. If the intervals overlap — and at these gaps they certainly do — report a three-way tie and either carry all three through section 9 or state that the choice of downstream model was arbitrary.

The precision-recall curve drawn for the wrong class, and its area never taken

EVIDENCE

Cell 18 calls `precision_recall_curve` on the held-out labels and the winner's positive-class probabilities, where positive means attack, and plots the result as the third panel. The function `average_precision_score` appears nowhere in the notebook, and the strings PR-AUC and average precision appear nowhere in any cell. The headline reported in cell 16, in the ablation table in cell 22, and in the section-13 conclusion is ROC-AUC throughout.

SCIENTIFIC CONSEQUENCE

Two failures compound here. First, you plotted the right kind of diagnostic and then declined to read it — the area is never taken and no number derived from that panel enters any ranking or conclusion. Second, and worse on this corpus, the curve as drawn is for the majority class. A precision-recall curve whose positive class is ninety-seven point seven percent of the data starts at a precision of 0.9767 and has almost nowhere to go; it is uninformative by construction. The curve that carries the operationally decisive information is the one with benign as the positive class, the two point three three percent minority, and that curve is never drawn at all.

REQUIRED REPAIR

Compute average precision on the probability vector already in scope in cell 18 — and compute it for the benign class specifically, by inverting the label, since benign is the minority here. Print both, bootstrap intervals, and rank the four learners on the benign-class average precision rather than on ROC-AUC.

Atlas proof

docs/course/security-ml-notebooks/nb14-ciciot2023.ipynb · cell 18,
`precision_recall_curve` plot; zero-based manifest cell IDs: 18=1a93dc54

The operating point is a library default

EVIDENCE

Cell 18 sets pred equal to pb greater than 0.5, a literal, and every operational number in section 9 — the confusion matrix, the six point nine nine percent false-positive rate, all thirty-three per-family recalls — is computed at that point. There is no cost matrix, no expected-cost minimisation and no threshold sweep anywhere in the notebook. The praxis pack's full-source scan reports an explicit cost matrix in zero of thirty-six notebooks and threshold selection from a precision-recall curve in zero of thirty-six.

SCIENTIFIC CONSEQUENCE

A missed attack on an IoT segment is a compromised device; a false alarm is an analyst-minute, and at seventy per thousand benign flows it is a great many analyst-minutes. Those costs are not comparable and the balance between them is set entirely by the threshold. Reporting the detector at 0.5 reports it at the one operating point nobody chose.

REQUIRED REPAIR

State a cost model — cost per missed attack against cost per false alarm, or equivalently an alert budget per analyst per shift — sweep the threshold, and report recall, precision and alert volume at the point that satisfies it. Report 0.5 only as a reference.

DEFENSIBLE REPLY

Faults three, fourteen and sixteen all end with the same instruction: choose the operating point from the precision-recall curve.

ORDERED CORRECTION

Fold into fault 3. Keep the one clause fault 3 lacks: all thirty-three per-family recalls, including the six the conclusion quotes, are functions of the 0.5 threshold rather than properties of the model.

The rare class is benign, and nothing is weighted toward it

EVIDENCE

Cell 16 constructs LogisticRegression with max_iter three hundred, RandomForestClassifier with sixty trees, XGBClassifier with eighty rounds and depth six, and LGBMClassifier with eighty rounds. Not one carries a class_weight or scale_pos_weight argument. Cell 11 prints positive rate 0.9767, so the negative class is the minority at one in forty-three. The praxis pack's scan confirms no class weighting in any of the thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

An unweighted loss is minimised by getting the majority right, and here the majority is attack. So the fitting procedure is systematically indifferent to exactly the class whose misclassification produces the six point nine nine percent false-positive rate that makes this detector unusable. The reported operational weakness is a direct consequence of a constructor argument that was not supplied, and correcting it costs one keyword.

REQUIRED REPAIR

Refit with class_weight balanced for the forest and the linear model and scale_pos_weight set to the positive-to-negative ratio for the boosted pair, and re-report the false-positive rate on benign traffic. If it falls materially, the operational figure in section 9 was an artifact of the loss weighting.

Uncalibrated forest votes thresholded as probabilities

EVIDENCE

Cell 18 takes best dot predict_proba and thresholds it at 0.5. The winner is RandomForest with sixty trees, so predict_proba returns the fraction of sixty trees voting positive — a quantity with sixty-one possible values, fitted on a class ratio of forty-three to one. There is no CalibratedClassifierCV, no isotonic or Platt fit, no reliability diagram and no Brier score anywhere in the notebook. The praxis pack's scan reports probability calibration present in zero of thirty-six.

SCIENTIFIC CONSEQUENCE

A vote fraction is not a probability, and one produced under a forty-three to one class ratio is systematically pushed toward the majority. So a cut at 0.5 sits at an unknown point on the actual risk scale, and every operational figure in section 9 inherits that. It also means the score cannot be combined with any other signal, ranked for triage, or given to an analyst as a confidence, which are the three things an operational detector's output is for.

REQUIRED REPAIR

Calibrate on a held-out calibration split with isotonic regression, print the reliability diagram and the Brier score before and after, and select the operating point on the calibrated scale.

DEFENSIBLE REPLY

This is the third request for the same threshold selection.

ORDERED CORRECTION

Retain only the calibration-specific residual: a sixty-tree vote fraction has sixty-one possible values and cannot be used for triage ranking or fused with another signal, which is a loss no threshold sweep repairs. Everything else here is fault 3.

One hundred and five devices, no device grouping

EVIDENCE

The abstract states that CICIoT2023 captures thirty-three attack types against a large IoT testbed of one hundred and five devices. Cell 16 calls `train_test_split` with a `stratify` argument and no `groups` argument. The section-13 conclusion states it directly: the split is random, not temporal or entity-grouped, and every number above therefore measures in-distribution separability only. The praxis pack's full-source scan reports `GroupKFold` or any entity-grouped split present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

With one hundred and five devices producing every row, a random split guarantees that for almost every held-out flow there exists a training flow from the same physical device, the same firmware, the same MAC, the same position in the topology, often within the same attack campaign. The model can therefore match a neighbour rather than learn a concept, and 0.999394 is consistent with device recognition rather than attack recognition. The notebook names this exact limitation and leaves the number in place.

REQUIRED REPAIR

Group by device — hold out entire devices, train on the rest — using `GroupShuffleSplit`, and report the drop from 0.999394. Leave-one-device-out is the deployment-relevant protocol for a testbed corpus and it is twenty lines.

DEFENSIBLE REPLY

Name the column. The forty-seven columns in this file are flow statistics, flag counters and protocol indicators; there is no device field, no MAC and no address in the mirror or in the CIC's own feature extractor output, so the grouped split you are prescribing cannot be built from this corpus at any price.

ORDERED CORRECTION

The filing's own rebuttal already anticipates this and is correct; promote it into the body. Replace REQUIRED with the true demand: state in the abstract that this corpus admits no entity-grouped protocol at all, so the 105-device generation structure is an unfixable confound rather than a design choice — and that unfixability, not the author's split, is the finding.

Attack campaigns are episodes in time, split at random

EVIDENCE

Cell 16 calls `train_test_split` with `test_size 0.25`, a `random_state` and `stratify`, and no time ordering. The conclusion states: the split is random, not temporal or entity-grouped. The praxis pack's scan reports a temporal split present in zero of thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

An attack campaign in a testbed is a burst: one tool, one configuration, one window of minutes. A random split cuts through the middle of every burst, so the training set and the held-out set contain flows from the same execution of the same tool against the same device. That is not generalisation to a new attack; it is interpolation within a single event. The relevant deployment question — does a detector trained on the campaigns of last week recognise next week's — is not asked.

REQUIRED REPAIR

Order by capture time, train on the earlier campaigns and test on the later ones, or hold out whole campaigns rather than random flows, and report the drop. If the mirror carries no timestamp, say so explicitly as a limitation of the mirror, which is itself a reason to use the authoritative distribution.

DEFENSIBLE REPLY

There is no clock in this file. IAT is a statistic computed inside a flow, not a time of day, and neither the mirror nor the CIC's own extractor emits a timestamp — so the campaign ordering you want me to split on does not exist in the data.

ORDERED CORRECTION

The filing already hedges correctly in its last sentence ('If the mirror carries no timestamp, say so explicitly'); make that the whole REQUIRED and drop the prescription in front of it. Do not assert that no temporal holdout is constructible — if the file preserves capture order a positional holdout would be, and neither party can verify the row order — but do not prescribe a sort key that is absent.

Four fixed configurations, and a winner separated by noise

EVIDENCE

Cell 16 fixes sixty trees for RandomForest, eighty rounds and depth six for XGBoost, eighty rounds for LightGBM, three hundred iterations for the linear model. No GridSearchCV, no RandomizedSearchCV, no nested cross-validation appears anywhere. The praxis pack's scan reports hyperparameter search present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

The winner leads the runner-up by 0.000033 in ROC-AUC. Adding forty trees to the forest, or ten boosting rounds to XGBoost, would plainly move either model by more than that. So the sentence best model: RandomForest is a statement about which four integers were typed into cell 16, and every number in sections 9 through 12 is computed for whichever model those integers happened to favour.

REQUIRED REPAIR

Nested cross-validation with an inner selection loop and an outer estimation loop, so that each algorithm is reported at its own best configuration and the ranking is a ranking of methods rather than of defaults.

DEFENSIBLE REPLY

Faults twelve and nineteen are one missing pass filed twice.

ORDERED CORRECTION

Merge into 12. Keep the one-line consequence: adding forty trees or ten rounds would move either model by more than the 0.000033 that decided the ranking, so 'best model: RandomForest' is a statement about four integers.

A security detector never tested against an adversary who adapts

EVIDENCE

There is no perturbation experiment, no stated evasion budget and no worst-case recall in any cell of the notebook. The features on which the model depends are flow-level counters and timing statistics — cell 20's top single feature is `rst_count` — every one of which an attacker can alter by changing tool, rate, packet size or timing. The literature review cites Sommer and Paxson 2010, whose argument is precisely that the operational environment is adversarial and non-stationary in a way laboratory evaluation does not capture.

SCIENTIFIC CONSEQUENCE

The reported recalls are recalls against an attacker who is not trying to avoid detection, because the attacks in the corpus were executed by scripts run for the purpose of generating labelled data. The weakest reported family already sits at 0.73 under those maximally favourable conditions. What that figure becomes when the adversary throttles, pads, or randomises inter-arrival times is the number that would justify deploying this, and it does not exist.

REQUIRED REPAIR

Define a perturbation budget in flow-feature space — for instance, allow the attacker to scale packet rate and inter-arrival timing within stated bounds and to suppress the reset counters — regenerate the held-out attack rows under that budget, and report worst-case per-family recall. That number, not 0.999394, is the detector's adversarial value.

DEFENSIBLE REPLY

The evasion study you are describing appears in none of the thirty-six notebooks and is a different piece of work. If the charge is that my per-family recalls are upper bounds reported as points, that is a fair sentence and it costs me one line.

ORDERED CORRECTION

Re-aim at what is actually claimed. The defensible form: the per-family recalls are reported as point estimates without the qualifier that they were measured against scripted tooling running with no evasion objective, so every one of them — including the 0.73 the conclusion nominates — is an upper bound quoted as a point. One sentence in the ledger, and it is a criticism of the reporting rather than a demand for an experiment no notebook in the series runs.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 262,144 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Seven out of thirty, which places this in the bottom band: the reported quantity is an artifact of the file. I want the class to notice that the band definition and the finding coincide literally rather than metaphorically. The row count is a software limit. The held-out size is a power of two because the row count is a power of two minus one. The class balance, the baseline, the benign population that carries the operational estimate — all of it is downstream of wherever a writer stopped writing, and this notebook, which is otherwise the most scrupulous of the four I was given, never once opened the authoritative distribution to check. It disclosed that it held a subset. Disclosure is not verification, and one download would have converted the disclosure into a fact.

Mobile-Money Fraud Detection: the PaySim Simulator

The first one million five hundred thousand rows of PaySim, an agent-based mobile-money simulator of six million three hundred sixty-two thousand six hundred twenty transactions, six features, positive rate 0.0011.

HEADLINE

XGBoost, held-out ROC-AUC 0.994731, printed in the cell-16 ranking table and carried into the cell-22 ablation table as headline, as-is.

DEFENSE

F

Selected-model estimand

DOCTORAL

6

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 9 critical (FATAL)
- 13 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be very clear that this is the most self-aware notebook in the set, and it scored a perfect ten out of ten from the friendly audit for exactly that reason. Section 3 states outright that PaySim's fraud script empties the victim account, that amount, oldbalanceOrg and newbalanceOrig therefore re-encode the rule that generated the label rather than merely correlating with it, and that every score below should be read as how well does a learner recover PaySim's scripted account-emptying rule. Section 10 explains that a shortcut living in the relation between two columns is invisible to both a single-column check and a whole-row check. Section 11 states that dropping one of three coupled columns produces a partial drop whether or not the coupling is driving the score, and so cannot be read as evidence against it. Section 13 opens with the sentence: the honest metric is rare-class recall, not the flattering ROC-AUC. Almost nobody writes that about their own work.

EXAMINER RULING

This notebook wrote down, in three separate sections, the precise experiment that would decide whether its headline is fraud detection or arithmetic, explained in detail why every test it does run is structurally incapable of deciding it, and then ran none of them and published 0.994731.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The fraud script writes the label and the features together

EVIDENCE

Section 3 states it in the notebook's own words: on fraudulent rows the transferred amount is, with few exceptions, the origin account's whole prior balance, and the post-transfer origin balance is driven to zero. The six features that survive cell 11's drop list therefore include amount, oldbalanceOrg and newbalanceOrig, and on a fraudulent row those three satisfy amount equals oldbalanceOrg, and newbalanceOrig equals zero. Cell 16 then reports XGBoost at held-out ROC-AUC 0.994731 and accuracy 0.999619.

SCIENTIFIC CONSEQUENCE

This is a relational leak, not a column leak. No single feature carries it — the single-feature audit in cell 20 correctly returns only 0.7846 — because the signal lives in an equation across three columns. A depth-six boosted tree reconstructs that equation trivially. So 0.994731 is not an estimate of how well fraud is detected. It is an estimate of how well XGBoost approximates a subtraction. The notebook says this in section 3 and reports the number as the headline anyway, and a number that measures a subtraction cannot support any claim about mobile money.

REQUIRED REPAIR

Two experiments, both cheap. First, write the generator's rule down as a hand-coded classifier — flag a row when newbalanceOrig equals zero and amount equals oldbalanceOrg — and print its held-out recall, precision and AUC. If the hand rule matches XGBoost's 0.994731, the model learned the rule and nothing else, and that is the finding. Second, drop all three coupled columns simultaneously, not one, refit, and report the number. That is the notebook's actual detection score.

A one-and-a-half-million-row headline fitted on about one hundred and thirty frauds

EVIDENCE

Cell 16 prints: materialized 1,500,000 rows, trained on 120,000 stratified subsample, held-out 375,000. Cell 11 prints positive rate 0.0011. One hundred and twenty thousand times 0.0011 is about one hundred and thirty-two, and the split is stratified so the ratio is preserved. The held-out count is exact and can be recovered from cell 18: the false-positive line names three hundred seventy-four thousand five hundred ninety-eight negatives, and three hundred seventy-five thousand minus that is four hundred and two positives. The cap is justified in a code comment reading these learners saturate long before then on this data, and no learning curve exists anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every claim in this notebook — the 0.994731, the grade A, the two per-family recalls, the four-learner ranking — rests on a model that saw on the order of one hundred and thirty examples of the thing it is meant to detect. That is a number a reader is entitled to before deciding what the headline means, and it is not printed. The word used in its place is the subtitle's phrase, greater than or equal to one million records, which describes the download.

REQUIRED REPAIR

Print the absolute positive count in train and in test on the same line as the row counts — it is one call to y-train dot sum. Then draw the learning curve the saturation comment asserts: held-out AUC and fraud recall against training size, and against training positive count, out to the full one and a half million rows. If it is flat, the cap is now justified rather than assumed.

Atlas proof

docs/course/security-ml-notebooks/nb15-paysim-fraud.ipynb; docs/course/security-ml-notebooks/assignments/evidence/SEAS8414-notebook-review-key.html · Notebook cells 11 and 16; review-key nb15 facts and criterion 7; zero-based manifest cell IDs: 11=50dcf49c, 16=e2b4684f

Both validity checks declared blind by their own author, grade A printed regardless

EVIDENCE

Section 10 says: a shortcut can live in the relation between two columns — the amount equals the account's prior balance, the balance afterwards is zero — and such a shortcut is invisible to both. Section 11 says: a partial, non-fatal drop is the expected outcome whether or not the coupling is driving the score, and cannot be read as evidence against it. Cell 20 then prints, four cells apart from that first statement: single-feat grade A, contam grade A, and then data trust grade: A.

SCIENTIFIC CONSEQUENCE

The two checks are one-column-at-a-time and whole-row-at-a-time. The artifact is a three-column relation. So the instruments and the defect are orthogonal by construction, and their returning A is not weak evidence of soundness — it is no evidence at all. The measured quantity has zero informational bearing on the question the grade appears to answer. And an A is what a reader skimming the PDF sees, three lines above the ablation.

REQUIRED REPAIR

Replace the grade with a test that can see a relation: fit a small model on the engineered residuals — amount minus oldbalanceOrig, and newbalanceOrig itself — and report its AUC alone. If that two-number model approaches 0.99, the relation is the score. Then either withdraw the letter grade for this corpus or state on the same printed line that the grade is not evaluable here.

DEFENSIBLE REPLY

The letter A is printed by a code cell whose thresholds I did not choose, and my conclusion spends four sentences telling the reader not to read it as a clean bill of health, naming the exact artifact it cannot see. Your remedy — fit on the residuals — is fault one's remedy.

ORDERED CORRECTION

Merge the residual-model experiment into fault 1. Keep only the sharp, separable point: the grading code is a fixed template applied to thirty-six corpora, and on this one its two instruments are orthogonal to the defect by construction, so the letter it emits is not weak evidence of soundness but no evidence at all — and a template that returns its top grade on a corpus its own author says is compromised is a broken instrument, which is a criticism of the series rather than of this notebook's prose.

LightGBM at 0.615463, below baseline on accuracy, unremarked

EVIDENCE

Cell 16's ranking table: XGBoost roc_auc 0.994731, RandomForest 0.955591, LogisticRegression 0.804841, LightGBM 0.615463. The same table prints LightGBM accuracy 0.996632 against the MajorityBaseline row at accuracy 0.998900. The printed guard above the table reads: any model must beat THIS, not 0.5, to be interesting. LightGBM does not. The section-13 conclusion names only the winner and never mentions LightGBM at all.

SCIENTIFIC CONSEQUENCE

XGBoost and LightGBM are both gradient-boosted decision-tree libraries, fitted on identical rows, identical features and identical labels, differing only in implementation. They report ROC-AUCs 0.379268 apart. That spread is not a modelling detail; it is a signal that the decision surface is fragile in a way no number in this notebook explains. And a sub-baseline learner is a diagnostic, not a row in a ranking table. Printing it silently invites the reader to average across four learners as though they were four opinions on the same well-posed problem.

REQUIRED REPAIR

Flag every sub-baseline model explicitly on the printed line, as the guard sentence promises. Then diagnose the 0.379268 spread: it is almost certainly LightGBM's default minimum-samples-per-leaf refusing to split on a class with roughly one hundred and thirty training positives, which is a finding about the corpus and should be reported as one.

DEFENSIBLE REPLY

My baseline row is printed inside the ranking table precisely so a reader can do that comparison, and the learner that fails it is fourth of four and carries nothing — no confusion matrix, no recall, no ablation, no conclusion sentence. The headline model clears the bar by 0.0007. Contrast notebook 16, where the winner does not.

ORDERED CORRECTION

Reframe around the 0.379268 spread, which is the real finding and which the filing already identifies: two implementations of gradient boosting on identical rows, features and labels differ by 0.379 in ROC-AUC, and neither the results section nor the conclusion remarks on it. That is a statement about the corpus — roughly 130 training positives against LightGBM's default `min\_child\_samples` of 20 and `min\_sum\_hessian\_in\_leaf` — and it is diagnostic in a way the sub-baseline accuracy is not. File it under A5 rather than B7.

Two recalls, four hundred and two positives between them, no counts and no intervals

EVIDENCE

Cell 18 prints: worst per-family recalls, CASH_OUT 0.596, TRANSFER 0.756. The code above it calls value_counts on the fraudulent held-out rows to get exactly those counts, uses them only for the gate reading if cnt less than five, continue, and never prints them. The total is recoverable by subtraction and is four hundred and two. The conclusion then quotes both recalls verbatim and says the weakest group sits at 0.596, which is where detection is thinnest.

SCIENTIFIC CONSEQUENCE

Four hundred and two positives split across two groups. Take the most favourable possible case and suppose all four hundred and two sat in one group: a recall of 0.596 on four hundred and two trials carries a ninety-five percent binomial interval of roughly plus or minus five points, because the square root of 0.6 times 0.4 over 402 is about 0.024, and twice that is 0.048. Split across two groups the intervals are wider still. So the difference the conclusion draws between 0.596 and 0.756 may not be a difference at all, and the notebook's own nominated honest number is quoted to three decimals with no denominator and no interval.

REQUIRED REPAIR

Print the positive count beside each recall, attach a Wilson binomial interval to each, and state whether the CASH_OUT and TRANSFER intervals overlap. If they do, withdraw the sentence identifying CASH_OUT as where detection is thinnest.

DEFENSIBLE REPLY

Do the interval arithmetic rather than sketching it. CASH_OUT holds about 203 held-out frauds and TRANSFER about 199; Wilson intervals of [0.527, 0.660] and [0.690, 0.807] do not overlap, so the difference my conclusion draws is a real difference and the sentence you want withdrawn is supported.

ORDERED CORRECTION

Strike 'So the difference the conclusion draws between 0.596 and 0.756 may not be a difference at all' and strike the conditional withdrawal in REQUIRED. The surviving charge is correct and narrower: `cnt` is computed by value_counts, used only for the count-five gate, and never printed, so a reader is given three-decimal recalls with no denominator and no interval — and the intervals happen, on this corpus, to be about plus or minus 0.066 and plus or minus 0.058, which is wider than the precision the conclusion reports them to.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

A fraudster who leaves money in the account is invisible

EVIDENCE

Section 3 establishes that the detectable signature is that the transferred amount equals the origin account's whole prior balance and the post-transfer balance is driven to zero. Cell 22 shows removing one of the three coupled columns still leaves 0.955019, so the signature is redundantly encoded. There is no evasion experiment, no perturbation budget and no worst-case recall anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The signature is not a property of fraud. It is a property of PaySim's fraud, which drains the account because a simulator author wrote it that way. A real attacker who moves ninety percent of a balance rather than one hundred percent breaks the equation amount equals oldbalanceOrg and leaves newbalanceOrig non-zero, and every feature this model depends on stops firing at once. The notebook contains no number describing what happens then, which means it contains no number describing the detector's value against an opponent.

REQUIRED REPAIR

Perturb the held-out fraudulent rows by leaving a residual balance — sweep it from one percent to fifty percent of the prior balance, recomputing amount and newbalanceOrig consistently — and plot recall against residual. Report the residual at which recall falls below one half. That figure is the detector's evasion budget.

DEFENSIBLE REPLY

I did not propose a fraud detector. Section three says in bold that every score below measures how well a learner recovers PaySim's scripted rule and must not be read as evidence about real mobile-money fraud — you are demanding the evasion budget of a capability I explicitly declined to claim.

ORDERED CORRECTION

The residual-balance sweep is a genuinely good experiment and is worth keeping as a constructive suggestion rather than a charge, because it is the cheapest possible demonstration of fault 1: perturb the held-out frauds to leave a one percent residual and the hand rule's recall falls from 0.977 to approximately zero, which shows in one line that the signal is the equation and not the fraud. Reword as 'the notebook could have shown its own central point in five lines and did not', filed under D1 with fault 1, rather than as an adversarial-robustness charge against a withdrawn claim.

A false-positive rate of zero celebrated while forty percent of one fraud type is missed

EVIDENCE

Cell 18 prints: operational FALSE-POSITIVE RATE at 0.5 equals 0.0000, fourteen benign flagged of three hundred seventy-four thousand five hundred ninety-eight. The same line prints CASH_OUT recall 0.596. Section 1 states: extreme class imbalance means accuracy is meaningless — recall on the rare class is the real question. The conclusion carries the false-positive figure and both group recalls, and carries no overall recall or miss count for the model at all.

SCIENTIFIC CONSEQUENCE

Do the arithmetic the notebook did not. There are four hundred and two frauds in the held-out set. At a CASH_OUT recall of 0.596 and a TRANSFER recall of 0.756, the model misses somewhere between roughly ninety-eight and roughly one hundred and sixty-two of them, depending how the four hundred and two divide between the two types. Against that, fourteen false alarms. The notebook leads its operational panel with the fourteen and never states the miss count in any form. In mobile money each miss is one customer's account emptied. The number that matters is absent and the number that does not is bolded.

REQUIRED REPAIR

Print the confusion matrix as text, not only inside a figure: true positives, false negatives, and the absolute count of frauds missed. Then state the alert budget in the other direction — fourteen false alarms per three hundred seventy-five thousand transactions scaled to a real provider's daily volume — and set the threshold from the precision-recall curve you are already drawing, not at 0.5.

DEFENSIBLE REPLY

Open the first panel of the figure you are quoting from. It prints 374,584, 14, 129 and 273 in the cells. The miss count is one hundred and twenty-nine and it is on the page, in the same output, above the line you are reading.

ORDERED CORRECTION

Two fixes. Delete 'never states the miss count in any form' and replace with the true and still-damaging form: the miss count exists only inside a rendered image, never in text, never in the conclusion's validity ledger, and never as a recall — the model's overall recall is $273/402 = 0.679$ and that number appears nowhere in the notebook. Second, the bracket 'somewhere between roughly ninety-eight and roughly one hundred and sixty-two' is correct — the actual figure is 129 — but you can now name it exactly, so name it. The remaining alert-budget demand is fault 14's.

The first one and a half million rows, and the full-corpus fraud rate never printed

EVIDENCE

Cell 11 sets `NROWS` to one million five hundred thousand and passes it as the `nrows` argument to `pandas read_csv`, which takes the leading rows. Section 3 states this in its own words: one million five hundred thousand analysed here, leading rows, per the loader read cap. The full file is stated as six million three hundred sixty-two thousand six hundred twenty rows. The printed positive rate 0.0011 is the prefix's. The full corpus's positive rate appears nowhere in the notebook.

SCIENTIFIC CONSEQUENCE

A prefix is not a sample. Whatever ordering the file carries — and a simulator writes its output in simulation order — the first twenty-three point six percent of the rows is a systematically selected region of it, not a random draw from it. Every quantity in this notebook, the positive rate, the duplicate rate of 0.000, the contamination of 0.000, the two per-family recalls, is conditioned on that region, and the notebook offers the reader no way to check whether the region is representative because it never reads the rest of the file.

REQUIRED REPAIR

Read the `isFraud` column alone across all six million three hundred sixty-two thousand six hundred twenty rows — one column, seconds of work, no memory pressure — and print the full-corpus positive rate beside the prefix's. If they differ, replace the prefix with a random sample of the same size and re-run. If they agree, you have converted an assumption into a measurement at negligible cost.

A random split on a phenomenon that only exists in time order

EVIDENCE

Cell 16 calls `train_test_split` with `test_size 0.25`, a `random_state` and a `stratify` argument, and no time-based ordering. The section-13 scope paragraph states it plainly: the split is random, not temporal or entity-grouped, and every number above therefore measures in-distribution separability only. The praxis pack's full-source scan reports a temporal or time-ordered split present in zero of the thirty-six notebooks in this series.

SCIENTIFIC CONSEQUENCE

A deployed fraud model is trained on the past and scored on the future, and that constraint is the entire source of the difficulty. A random split removes it. The model here is permitted to learn from simulation hours that come after the hours it is evaluated on, which is not a subtle optimism — it is a different problem, and an easier one. The disclosure paragraph names this exactly and then leaves the number in place, which is naming the limitation rather than removing it.

REQUIRED REPAIR

Order the slice by its simulation time index, train on the first eighty percent of the timeline and score on the last twenty percent, and report the drop from 0.994731. Report both numbers side by side; the difference is the price of the random split and it is the number a practitioner needs.

The two account-identifier columns deleted without printing their cardinality

EVIDENCE

Cell 11's drop list explicitly names `nameOrig` and `nameDest` and removes both. Cell 16 calls `train_test_split` with a `stratify` argument and no `groups` argument. The praxis pack's scan reports `GroupKFold` or any entity-grouped split present in zero of thirty-six notebooks. Nowhere does the notebook print `nunique` for either identifier column.

SCIENTIFIC CONSEQUENCE

Deciding not to group is legitimate; deciding without looking is not. If destination accounts repeat — and a simulator that models mule accounts would make them repeat — then the same destination appears on both sides of the split and the model can learn which accounts receive fraud rather than which transactions are fraudulent. One call to `nunique` on each column, printed, settles the question in a line. It was not run, so the notebook cannot say whether grouping was needed or superfluous.

REQUIRED REPAIR

Print `nunique` for `nameOrig` and `nameDest` against the row count. If either column has substantially fewer distinct values than rows, run `GroupShuffleSplit` on it and report the drop. If both are near-unique, print that fact and the grouping question is closed on evidence.

DEFENSIBLE REPLY

`PaySim` generates a fresh origin customer for nearly every transaction, so `nameOrig` is close to one-to-one with rows and grouping on it would be very nearly a no-op — which would make the omitted check a formality rather than a defect.

PR-AUC named in the literature table, drawn on the page, never computed

EVIDENCE

The related-approaches table in cell 3 carries the row imbalanced-learning studies with the caveat AUC flattered; PR-AUC and recall matter. Section 13 opens: the honest metric is rare-class recall, not the flattering ROC-AUC. Cell 18 calls `precision_recall_curve` and plots it in the third panel. The function `average_precision_score` appears nowhere in the notebook, and no PR area is printed, ranked on, or carried into the conclusion. The headline reported everywhere remains the flattering ROC-AUC of 0.994731.

SCIENTIFIC CONSEQUENCE

You plotted the right diagnostic and then declined to read it. At the printed rate of 0.0011 — about one fraud in every nine hundred transactions — ROC-AUC is close to insensitive to the false-positive count, which is why 0.994731 looks like near-perfection while four hundred and two positives sit in the held-out set with roughly a third of them missed. Average precision would have shown that immediately. The curve containing the answer is printed on the page beside the number that hides it.

REQUIRED REPAIR

Call `average_precision_score` on the probability vector already in scope in cell 18, print it, bootstrap its interval over held-out rows, and re-rank all four learners on it. Put that number, not the ROC-AUC, in the conclusion your own section 13 says should carry the honest metric.

Atlas proof

docs/course/security-ml-notebooks/nb15-paysim-fraud.ipynb · Cell 18 (0-based); zero-based manifest cell IDs: 18=9e850519

13

MAJOR

UNTESTED

DOWNGRADED

NO CALIBRATION

The cited calibration paper, and no calibration

EVIDENCE

Reference two in cell 26 is Dal Pozzolo and colleagues, 2015, Calibrating probability with undersampling for unbalanced classification. Cell 18 computes pred as pb greater than 0.5 and reports operational FALSE-POSITIVE RATE at 0.5. There is no CalibratedClassifierCV, no isotonic or Platt fit, no reliability diagram and no Brier score anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A boosted tree fitted on a class ratio of roughly nine hundred to one emits scores that are not probabilities, and the point of the cited paper is precisely that the imbalance warps the scale. The value 0.5 therefore does not correspond to any belief about the transaction. Every operational quantity in section 9 — the fourteen false positives, the recall of 0.596, the confusion matrix — is a consequence of a cut placed at an arbitrary point on an arbitrary scale.

REQUIRED REPAIR

Calibrate on a held-out calibration split, print the reliability diagram and the Brier score before and after, and only then select the operating point — from the precision-recall curve, on the calibrated scale.

DEFENSIBLE REPLY

Faults eight, thirteen and fourteen all finish with the same instruction.

ORDERED CORRECTION

Fold into fault 14. Retain only the calibration-specific residual: an uncalibrated score at a 900-to-1 ratio cannot be ranked for triage or fused with another signal, which no threshold sweep repairs.

14

CRITICAL

UNTESTED

VERIFIED

THRESHOLD FIXED AT 0.5, NO COST MODEL

No cost matrix on a problem whose two errors differ by orders of magnitude

EVIDENCE

Cell 18 sets pred equal to pb greater than 0.5, a literal. There is no cost matrix, no expected-cost minimisation, and no threshold sweep anywhere in the notebook. The praxis pack's full-source scan reports an explicit cost matrix present in zero of thirty-six notebooks and threshold selection from a PR curve present in zero of thirty-six.

SCIENTIFIC CONSEQUENCE

Section 3 tells us what a missed fraud is in PaySim: the victim's entire account balance moved out and cashed out. A false positive is one legitimate transfer held for review. Those costs are not comparable and are not within two orders of magnitude of each other. When costs are that asymmetric the threshold is not a detail of the evaluation; it is the result. Reporting at 0.5 reports the point a library default happened to sit at.

REQUIRED REPAIR

State a cost matrix in currency — average balance lost per missed fraud against handling cost per false review — sweep the threshold across the held-out set, plot expected cost, and report recall and precision at the cost minimum. Report 0.5 only as a reference point, if at all.

One fraud in about nine hundred, and no reweighting

EVIDENCE

Cell 11 prints positive rate 0.0011. Cell 16 constructs LogisticRegression with max_iter 300, RandomForestClassifier with sixty trees, XGBClassifier with eighty rounds and depth six, and LGBMClassifier with eighty rounds. None carries a class_weight or scale_pos_weight argument. The praxis pack's scan confirms class weighting present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

An unweighted loss at a ratio of about nine hundred to one is minimised almost entirely by classifying negatives correctly, and the reported outcomes are exactly what that predicts: a false-positive rate rounding to zero, a CASH_OUT recall of 0.596, and one learner — LightGBM — collapsing to 0.615463 because its default leaf constraints will not carve out a class with roughly one hundred and thirty training positives. Every one of those three results is a consequence of a constructor argument that was not supplied.

REQUIRED REPAIR

Refit with scale_pos_weight set to the negative-to-positive ratio, roughly nine hundred to one, for the boosted learners and class_weight balanced for the linear and forest learners, and re-report both per-family recalls and the four-learner ranking. If LightGBM recovers, the 0.379268 spread in cell 16 was an artifact of the loss weighting and the ranking table must be withdrawn.

The headline sits one cross-validated standard deviation above the cross-validated mean

EVIDENCE

Cell 24 prints: XGBoost three-fold CV ROC-AUC equals 0.9862 plus or minus 0.0089. Cell 16's headline is 0.994731. The difference is 0.008531 — almost exactly one of those standard deviations. Now read the line above the CV call in cell 24: cvX, cvy equals Xtr dot iloc up to forty thousand. The folds come from the first forty thousand rows of the one hundred and twenty thousand-row training subsample, about two point seven percent of the corpus. Neither the print line, nor section 12, nor the conclusion states that bound. Notebook 14 in this same series states it explicitly in both places, so the disclosure exists in the template and is absent here.

SCIENTIFIC CONSEQUENCE

A reader given 0.994731 and 0.9862 plus or minus 0.0089 with no scope statement reads the second as corroborating the first. It does not. It is fold-to-fold stability of a fit on a forty-thousand-row slice, and it happens to sit a full standard deviation below the number it appears to support — which is itself the signal that the single held-out split drew favourably. No bootstrap interval on the headline exists to check that, because the praxis pack's scan finds none in any of the thirty-six notebooks.

REQUIRED REPAIR

Bootstrap the held-out AUC over resampled test rows and report the interval. Carry the forty-thousand-row bound into the printed CV line as notebook 14 does. And state explicitly whether the headline falls inside the bootstrap interval of the cross-validated estimate.

No search, and a 0.379 spread that proves it mattered

EVIDENCE

Cell 16 hard-codes sixty trees for the forest, eighty rounds and depth six for XGBoost, eighty rounds for LightGBM, and three hundred iterations for the linear model. There is no GridSearchCV, no RandomizedSearchCV and no nested cross-validation in the notebook. The praxis pack's scan reports hyperparameter search present in zero of thirty-six.

SCIENTIFIC CONSEQUENCE

Ordinarily the absence of a search is a matter of degree. Here the notebook prints its own proof that it was decisive: two implementations of the same algorithm class, on identical data, differ by 0.379268 in ROC-AUC. That gap is not a property of gradient boosting. It is a property of two sets of default leaf and split constraints meeting a class with roughly one hundred and thirty training positives. So the ranking table does not rank methods. It ranks defaults.

REQUIRED REPAIR

Nested cross-validation, inner loop selecting each learner's hyperparameters — at minimum `min_child_samples`, `min_child_weight` and the learning rate for the boosted pair — and outer loop estimating generalisation, so each algorithm is reported at its own best rather than at whatever its maintainers shipped.

DEFENSIBLE REPLY

Faults sixteen, seventeen and eighteen are one absent pass filed three times, and the spread you cite as proof is already the evidence in faults three and fifteen.

ORDERED CORRECTION

Merge into 16. Keep one sentence: on this corpus the absence of a search is not a matter of degree, because the notebook prints its own proof that it was decisive — two implementations of one algorithm class, identical data, 0.379268 apart.

Six decimals on a quantity whose spread is in the third

EVIDENCE

Cell 16 rounds roc_auc to six decimals, and cell 22's ablation table reports 0.994731, 0.989154 and 0.955019 at the same resolution. Cell 24 reports the cross-validated standard deviation as plus or minus 0.0089. The conclusion then quotes the difference 0.005577 between two ablation rows and characterises it as re-split noise, which concedes that a quantity in the third decimal is noise while continuing to print the sixth.

SCIENTIFIC CONSEQUENCE

Reported precision is a claim about resolution. Six digits assert that the seventh would be meaningful, when the notebook's own text says the third is not. A reader comparing 0.994731 against 0.989154 is being invited to distinguish quantities the notebook has already told them cannot be distinguished.

REQUIRED REPAIR

Report every AUC to the resolution the bootstrap interval supports — on this evidence, three decimals at most — and print the interval beside it rather than trailing digits.

DEFENSIBLE REPLY

The six decimals are deliberate and the code says so: the comment in cell 16 reads six dp, a 1.000000 is a red flag, not a win, so the extra digits exist to expose a saturated score rather than to assert resolution.

No published PaySim result reproduced or compared against

EVIDENCE

The literature review in cell 3 lists Lopez-Rojas, Elmir and Axelsson 2016 and carries a related-approaches row reading rule plus ML with the caveat simulated data; fraud confined to two transaction types. The notebook explicitly labels that table as drawn from the wider literature and not measurements reproduced on this exact corpus. No cell reproduces the originators' protocol, their split, or their metric.

SCIENTIFIC CONSEQUENCE

PaySim was published with its own detection results by its own authors, and this notebook analyses a prefix of that corpus with a different split, a different subsample size and a different metric. There is consequently no quantity in the notebook that any reader of the 2016 paper could line up against anything in it, and the reported score floats free of every prior number in the field.

REQUIRED REPAIR

Reproduce the originators' evaluation protocol on the same slice — their split and their reported metric — and print your figure next to theirs, or state in the abstract that no comparison to prior work is offered and why.

DEFENSIBLE REPLY

The notebook never claims comparability; it explicitly marks the related-approaches table as not reproduced here, which is a more honest disclaimer than most papers offer, and reproducing a 2016 simulation-conference protocol on a differently-versioned Kaggle export may not be well defined.

Model-internal importances offered as the explanation panel

EVIDENCE

Cell 18 reads best dot `feature_importances_` and plots the top twelve as the importance panel. There is no permutation importance and no SHAP computation anywhere in the notebook. With only six features, that panel is the entire stated account of what the model uses.

SCIENTIFIC CONSEQUENCE

The central question of this notebook is whether the score is carried by the three coupled balance columns. A built-in tree importance measured on the training set is exactly the wrong instrument for answering it, because correlated features split their apparent importance arbitrarily between themselves depending on which one the tree reached first. So the panel that appears to answer the coupling question systematically understates each coupled column by dividing the credit among them.

REQUIRED REPAIR

Permutation importance on the held-out set, computed both for individual columns and for the coupled block of three permuted together. The block permutation is the number that speaks to the central claim and it is one grouped call.

DEFENSIBLE REPLY

With only six features and an explicit three-column coupling already named in section 3, the importance panel is decorative rather than load-bearing, and the ablation in cell 22 — not the importance chart — is what the notebook actually reasons from.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 375,000 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45\text{e-}06$ and a range of $1.18\text{e-}05$, against a reported de-duplication gap of $7.07\text{e-}06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result. This notebook is where the charge bites hardest: its printed duplicate rate is essentially zero, so de-duplication changed almost nothing about the corpus — and the score moved anyway. That movement can only be the split.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

Closing. Six out of thirty, and I want to explain the axis that most needs explaining. Construct validity scores one, not two, even though an ablation was executed and a number was printed — because section 11 of this notebook states in writing that the executed ablation cannot be read as evidence about the coupling in either direction. A test its own author declares uninformative about the construct scores as disclosure, not as method. That is the whole verdict in one line. This notebook understood its problem better than its own auditors did, wrote the understanding down four separate times, and then did not spend the twenty minutes that would have converted the understanding into a measurement. The friendly pack gave it ten out of ten for the quality of that writing. I am giving it six out of thirty for the quality of the evidence, and both of us are right about what we measured.

Credit-Card Fraud Detection: the Sparkov Simulated Dataset

One million eight hundred fifty-two thousand three hundred ninety-four simulated card transactions from the Sparkov generator, pulled off the Kaggle upload kartik2112/fraud-detection, twelve features, positive rate 0.0052.

HEADLINE

XGBoost, held-out ROC-AUC 0.948632, printed in the cell-16 ranking table and named the winning learner in the section-13 conclusion.

DEFENSE

F

Selected-model estimand

DOCTORAL

5

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

5.5

/10 · self A

D1 1 External validity	D2 0 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 0 Honesty / scope

- 21 live faults
- 10 critical (FATAL)
- 11 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Credit where it is owed, and it is owed generously. This notebook ran the drop-the-feature ablation, the ablation came back against it, and it printed the collapse rather than burying it: 0.948632 down to 0.574302 in cell 22. Its own section-13 conclusion then says, in plain words, that this is a genuine single-feature shortcut and that the headline must not be read as broad detection capability. It goes further and says the letter grade understates the problem and that you should trust the ablation rather than the grade. It also tells you, in the literature review, that the Sparkov generator is software and not a peer-reviewed study. That is more candour than most of what gets published in this area.

EXAMINER RULING

This notebook fitted a threshold on the dollar amount, called it a fraud detector, and printed three cells later the number that refutes it — and then, in the same conclusion, declared a winner whose accuracy sits below the very baseline that same conclusion calls decisive.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The detector is one column: transaction amount

EVIDENCE

Cell 22 prints the ablation table. Headline, as-is: 0.948632. Shortcut feature dropped, meaning amt removed and the model refitted: 0.574302. That is a fall of 0.374330 onto a floor where 0.5 is a coin toss. The independent probe in the praxis pack reproduces it and goes one step further: base 0.9457, drop amt 0.5817, drop amt and gender 0.5800. The second column is worth 0.0017. There is no second column.

SCIENTIFIC CONSEQUENCE

Sparkov is a generator. It draws a transaction amount for a legitimate purchase from one distribution and for a fraudulent purchase from another, because a human wrote those two distributions into the software. What has been measured here is the distance between two numbers a programmer chose. Every inference from 0.948632 to the phrase fraud detection passes through the assumption that a real fraudster's spend distribution matches Sparkov's, and nothing in this notebook supports that assumption.

REQUIRED REPAIR

Refit with amt removed and report that number as the headline, because 0.574302 is the actual measurement of everything this feature set knows about fraud other than price. Then, separately, take the amt distribution for fraudulent and for legitimate rows out of the corpus and compare it against a published distribution from real card data — Dal Pozzolo's own European card corpus is the obvious comparator and it is already in your reference list. If the two do not match, say so, and the number does not leave the simulator.

The declared winner fails the notebook's own decisive bar

EVIDENCE

Cell 16 prints the majority-class baseline accuracy as 0.9948, with the sentence: any model must beat THIS, not 0.5, to be interesting. In the same printed table, XGBoost accuracy is 0.994593. That is below 0.9948. So is LightGBM at 0.994055, and so is LogisticRegression at 0.994373. Three of the four learners fail. The single learner that clears the bar is RandomForest at 0.995046, and RandomForest was not selected, because selection is by ROC-AUC, where RandomForest is second worst at 0.884538. The section-13 conclusion then repeats the rule — the accuracy column must clear that bar to mean anything — and names XGBoost the winning learner two sentences later.

SCIENTIFIC CONSEQUENCE

Do the arithmetic on the held-out set, which cell 16 prints as four hundred sixty-three thousand and ninety-nine rows. Always predicting legitimate produces about two thousand four hundred and eight errors, that is 0.0052 times 463,099. The selected winner produces about two thousand five hundred and four, that is one minus 0.994593, times 463,099. The model this notebook chose makes roughly ninety-six more mistakes than a constant. It is not that the comparison was omitted. The comparison was performed, printed, declared decisive, and then contradicted on the same page.

REQUIRED REPAIR

Either withdraw the accuracy bar as a selection criterion and say explicitly that selection is on ROC-AUC alone and that the winner does not beat a constant on accuracy, or select RandomForest and report its ROC-AUC of 0.884538 as the headline. What you may not do is state a rule, fail it, and print the failing model as the result.

Atlas proof

docs/course/security-ml-notebooks/nb16-sparkov-cc-fraud.ipynb · cells 15–16 and 25; zero-based manifest cell IDs: 15=2ee16bbf, 16=7df8ae74, 25=751eecfc

Six merchant categories at zero recall, on unknown counts

EVIDENCE

Cell 18 prints: worst per-family recalls, gas_transport 0.0, home 0.0, personal_care 0.0, food_dining 0.0, kids_pets 0.0, grocery_net 0.0. The praxis pack's probe reports fourteen groups with zero skipped, so those six are six of fourteen. Now read the code in cell 18. It iterates over fam_te where y equals one, calling value_counts, which produces the number of frauds per category. That count is used for exactly one thing — the gate that reads, if cnt less than five, continue — and is then thrown away. It is never printed, never plotted, and never reaches the conclusion.

SCIENTIFIC CONSEQUENCE

A recall of 0.0 on five positives and a recall of 0.0 on five hundred positives are different findings. The first is a sampling accident. The second is a categorical operational failure. The reader cannot tell which of these the notebook is reporting, and neither can the author, because the denominator was computed and deleted. The held-out set contains two thousand four hundred and thirteen frauds, that is 463,099 minus the 460,686 legitimate transactions the false-positive line names, and how those two thousand four hundred and thirteen distribute across fourteen categories is the single most operationally important quantity in this notebook. It is not in it.

REQUIRED REPAIR

Print the per-category positive count beside every recall — it is one variable already sitting in memory in cell 18. Then attach a binomial confidence interval to each recall, because a recall estimated on a two-digit denominator has an interval wide enough to swallow the estimate. Report every category with its n, and label any category whose interval spans more than half the unit interval as unscorable rather than as zero.

A data-trust grade of A decided by six ten-thousandths

EVIDENCE

Cell 20 prints: best single-feature AUC = 0.8494, feature amt, and then data trust grade A. Read the grading line in the same cell. The letter is B if the single-feature AUC is at or above 0.85, and A below it. The printed value is 0.8494. The margin is 0.0006. The praxis pack's probe then reseeds the sixty-thousand-row audit sample eight times and gets a range from 0.8057 to 0.8589; seed seven grades it B. The full-corpus value is 0.8343 against the published 0.8494, and the pack notes the published draw is a high one, some twenty-five times the deciding margin.

SCIENTIFIC CONSEQUENCE

A letter grade produced by a hard threshold on a quantity whose resampling spread is 0.053 wide carries no information about the corpus. It carries information about which sixty thousand rows were drawn. And the letter that this coin-flip produced is A — the top grade — on a corpus where the very next cell shows the model collapsing to 0.574302 when one column is removed. The notebook itself says the grade understates the problem, which is the correct reading and also a confession: a diagnostic that returns its best grade on this corpus is not a diagnostic.

REQUIRED REPAIR

Report the single-feature AUC as an interval, not a letter — the eight-seed spread 0.8057 to 0.8589 is already computed and is the honest object. Delete the letter grade, or make it a function of the ablation rather than of the single-feature threshold, since the ablation is the test that carries information here and the notebook says so itself.

The conclusion misreads its own printed zeros

EVIDENCE

The exact sentence in the section-13 conclusion is: the weakest group sits at 0.000, so the model misses most of it. A recall of 0.000 is not most. It is all. And the printed line in cell 18 does not show one weakest group. It shows six groups tied at exactly 0.0 — gas_transport, home, personal_care, food_dining, kids_pets and grocery_net.

SCIENTIFIC CONSEQUENCE

The honesty apparatus of this entire series rests on the claim that the conclusion reads the printed numbers back accurately. Here it does not. The word most converts a total operational blindness into a partial shortfall, and the singular the weakest group converts six failures into one. A reader who reads only the conclusion — which, for a conclusion, is the design intent — comes away believing the detector underperforms somewhere. It does not underperform. In six of fourteen merchant categories it is inert.

REQUIRED REPAIR

Replace the sentence with the printed fact: in six of fourteen merchant categories the detector recovered zero frauds at threshold 0.5. Name the six. Then state the consequence, which is that fraud in those categories is not merely under-detected but wholly undetected by this model at this operating point.

DEFENSIBLE REPLY

Read the sentence before the one you are quoting. My conclusion lists all six zeros by name, verbatim, and then says 'the weakest group sits at 0.000'. The reader has the six in front of them. What you have found is one wrong word — 'most' where it should read 'all' — not a conclusion that converts six failures into one.

ORDERED CORRECTION

Narrow to the surviving error, which is real and worth a sentence: a recall of 0.000 means the model recovered none of that category, and the word 'most' understates a total blindness as a partial shortfall. That is the friendly probe's finding too — it calls it a wording slip and still awards the criterion. Drop 'describes a single group where six were printed'; the six are printed in the same paragraph. If you want the stronger version of this fault, use it against the absent overall recall instead: 360 of 2,413 frauds recovered, 0.149, is stated in no text in the notebook.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Train and test files merged, then re-split at random

EVIDENCE

Section 3 of the notebook names the row count as approximately 1.85 million and identifies its composition in its own words as fraudTrain plus fraudTest. Cell 11 then globs every CSV under the download directory and calls pandas concat on all of them, producing one million eight hundred fifty-two thousand three hundred ninety-four rows. Cell 16 then draws train_test_split with test_size 0.25 and a stratify argument. Nothing anywhere in the notebook acknowledges that a partition existed before that call destroyed it.

SCIENTIFIC CONSEQUENCE

Fraud is a time-ordered phenomenon. A file named test that ships alongside a file named train is, at minimum, a claim by whoever built the corpus about which rows come after which. Merging them and re-splitting at random means training rows and test rows are now interleaved in whatever ordering the source used, and the model is permitted to learn from transactions that postdate the ones it is scored on. That is not a subtle statistical concern. It is the model being shown the future.

REQUIRED REPAIR

Load fraudTrain as train and fraudTest as test, fit on the first and score on the second, and report that number next to the random-split number. The difference between the two is the quantity of interest and it costs one line of loader code — you already know both filenames, because your own section 3 prints them.

Atlas proof

docs/course/security-ml-notebooks/nb16-sparkov-cc-fraud.ipynb · cell 11, file loading and concatenation; zero-based manifest cell IDs: 11=f9f8380d

PR-AUC named as the honest signal and never computed

EVIDENCE

The first line of section 13 reads: fraud is a small minority, so the honest signal is per-merchant-category recall and PR-AUC, not accuracy. Cell 18 calls precision_recall_curve and plots the result in the third panel. Search the notebook for average_precision_score and it is not there. Search it for the string PR-AUC and you find exactly two occurrences, both in prose — that conclusion sentence, and nothing else. The number is never taken.

SCIENTIFIC CONSEQUENCE

At a positive rate of 0.0052, ROC-AUC is close to blind to the false-positive count, because the negative class is a hundred and ninety times larger than the positive one and swamping it costs the false-positive rate almost nothing. Average precision is the statistic that is not blind to it. The notebook identified that, drew the curve that contains the answer, and then did not integrate it. That is not an oversight of ignorance. It is one function call away from an object already in scope.

REQUIRED REPAIR

Call average_precision_score on the same probability vector already held in the variable pb in cell 18, print it, bootstrap a confidence interval over the held-out rows, and rank the four learners on it. If the ranking changes, the ROC-AUC ranking in cell 16 is withdrawn.

Atlas proof

docs/course/security-ml-notebooks/nb16-sparkov-cc-fraud.ipynb · cells 18 and 25; zero-based manifest cell IDs: 18=1dfa40a0, 25=751eecfc

The cited paper is about calibration; no calibration is performed

EVIDENCE

Reference two in cell 26 is Dal Pozzolo and colleagues, 2015, Calibrating probability with undersampling for unbalanced classification. Cell 18 computes `pred as pb greater than 0.5` and reports the operational false-positive rate at 0.5 as 0.0010, four hundred fifty-one legitimate transactions flagged of four hundred sixty thousand six hundred eighty-six. Nowhere in the notebook is there a `CalibratedClassifierCV`, a Platt scaling, an isotonic fit, a reliability diagram, or a Brier score.

SCIENTIFIC CONSEQUENCE

Boosted trees trained on a class ratio of one to a hundred and ninety produce scores that are not probabilities. Dal Pozzolo's 2015 result, which this notebook lists in its own references, is precisely that undersampled and imbalanced training warps the probability scale and must be corrected before a threshold means anything. The number 0.5 has no meaning on an uncalibrated score. So the operating point reported here is an arbitrary cut on an arbitrary scale, and every downstream figure — the false-positive rate, the six zero recalls, the confusion matrix — inherits that arbitrariness.

REQUIRED REPAIR

Fit the calibration on a held-out calibration split, either Platt or isotonic, print the reliability diagram and the Brier score before and after, then choose the operating point on the calibrated scale from the precision-recall curve you are already drawing.

DEFENSIBLE REPLY

Faults nine and ten end with the same instruction.

ORDERED CORRECTION

Fold into fault 10. Keep only the residual a threshold sweep does not repair: at a 192-to-1 ratio the score is not on a probability scale, so it cannot be ranked for triage or fused with another signal, and Dal Pozzolo 2015 — reference two — is the paper that says so.

Cost-sensitive modelling named in the notebook's own table, and absent

EVIDENCE

The related-approaches table in cell 3 has a row reading cost-sensitive fraud models, with the caveat simulated distribution. Cell 18 sets pred equal to pb greater than 0.5, a literal. There is no cost matrix anywhere in the notebook, no expected-cost minimisation, and no sweep of the threshold.

SCIENTIFIC CONSEQUENCE

In card fraud a false negative costs the issuer the transaction value plus chargeback handling; a false positive costs a declined card and some fraction of a customer relationship. Those are not the same number and they are not within a factor of ten of each other. When the costs are asymmetric the threshold is not a detail of the evaluation, it is the result. Reporting a detector at 0.5 with no cost model reports the one operating point chosen by a library default rather than by the problem.

REQUIRED REPAIR

State a cost matrix in currency — a plausible average fraud loss and a plausible false-decline cost — sweep the threshold, and report the expected cost per thousand transactions as a function of threshold with its minimum marked. Report recall and precision at that minimum, not at 0.5.

A positive rate of 0.0052 and no reweighting anywhere

EVIDENCE

Cell 11 prints positive rate 0.0052. Cell 16 constructs LogisticRegression with max_iter 300 and nothing else, RandomForestClassifier with sixty trees, XGBClassifier with eighty rounds and max_depth 6, and LGBMClassifier with eighty rounds. Not one of the four carries a class_weight or a scale_pos_weight argument. The praxis pack's full-source scan confirms this holds across all thirty-six notebooks in the series.

SCIENTIFIC CONSEQUENCE

The six zero recalls in cell 18 are the direct, predictable consequence. An unweighted loss on a 0.0052 positive rate is minimised almost entirely by getting the negatives right, and in a merchant category with a small number of frauds the cheapest thing the model can do is predict legitimate for every row in it. That is exactly what it did in six categories. The failure the notebook reports as its most operationally important result is a failure it engineered by declining a single constructor argument.

REQUIRED REPAIR

Refit with scale_pos_weight set to the negative-to-positive ratio, roughly one hundred and ninety-two to one, for the boosted models and class_weight balanced for the linear and forest models, and report per-category recall again. If the six zeros do not move, that is a finding about the feature set. If they do move, the reported result was an artifact of the loss weighting.

DEFENSIBLE REPLY

My cell 22 shows the entire feature set other than `amt` reaching 0.574 — a coin toss plus a nudge. scale_pos_weight reweights a loss; it does not conjure signal that is not in the twelve columns, and eleven of my twelve columns describe who the cardholder is rather than what they did.

ORDERED CORRECTION

Strike 'The six zero recalls in cell 18 are the direct, predictable consequence' and 'a failure it engineered by declining a single constructor argument'. The surviving charge is that no weighting was tried and no evidence is offered either way, so the reader cannot tell whether the zeros are a feature-set failure or an objective failure — and the filing's own REQUIRED already states the correct disjunction ('If the six zeros do not move, that is a finding about the feature set'). Lead with the disjunction rather than with the conclusion.

The cited feature-engineering paper is the missing method

EVIDENCE

Reference three in cell 26 is Bahnsen and colleagues, 2016, Feature engineering strategies for credit card fraud detection. The twelve features this notebook actually fits on are enumerated in its own abstract: merchant, amt, gender, city, state, zip, lat, long, city_pop, job, merch_lat, merch_long. Cell 11's drop list removes trans_date_trans_time, unix_time, cc_num and trans_num. So the timestamp and the card identity — the two columns from which every Bahnsen-style aggregate is built — were both available and both discarded.

SCIENTIFIC CONSEQUENCE

Of the twelve surviving columns, one is a price and the other eleven describe who the customer is and where they live. Six of them — city, state, zip, lat, long, and city_pop — are five encodings of a single quantity, the cardholder's home location, plus its population. Nothing in the vector says anything about how this cardholder behaves, because the columns that would have supported that were dropped as identifiers. The model therefore cannot detect anomalous behaviour, only anomalous price, and the ablation in cell 22 confirms that is exactly what it does.

REQUIRED REPAIR

Rebuild the feature vector using cc_num as a grouping key rather than as a feature: per-card transaction count and amount statistics over trailing windows of one hour, one day and seven days, time since previous transaction, and distance from the previous merchant location. That is the Bahnsen construction, it is in your reference list, and both source columns are in the file you already downloaded.

Nominal categories given an arbitrary integer order, before the split

EVIDENCE

Cell 11 loops over every object-dtype column and applies `LabelEncoder fit_transform` on the whole frame, before cell 16 draws the split. Cell 2 of the notebook itself flags the problem in advance: it says several of these — merchant, city, job — are high-cardinality categoricals a model can simply memorise. The conclusion discloses the transductive step, and then adds the sentence: on an all-numeric corpus that step is a no-op. This corpus is not all-numeric.

SCIENTIFIC CONSEQUENCE

`LabelEncoder` assigns integers in sorted order of the string, so merchant number four hundred and one is placed between four hundred and four hundred and two for no reason connected to anything. A tree can only recover the true grouping by spending splits to carve out individual integers, so the encoding wastes model capacity and produces splits that are meaningless out of sample. And because the encoder was fitted before the split, the integer assigned to a category depends on which categories exist in the held-out rows.

REQUIRED REPAIR

Fit the encoder inside the fold with an explicit unseen-category bucket, and use an encoding appropriate to nominal data — one-hot for the low-cardinality columns, out-of-fold target encoding for merchant and job — rather than an arbitrary integer order.

DEFENSIBLE REPLY

Gradient-boosted trees are widely used with integer-coded nominal features and recover the groupings adequately given enough depth and rounds; and the mapping never consults the label, so the transductive step leaks no label information, which is exactly what the conclusion says.

Atlas proof

`docs/course/security-ml-notebooks/nb16-sparkov-cc-fraud.ipynb · cell 11 LabelEncoder loop; cell 25 limitation disclosure; zero-based manifest cell IDs: 11=f9f8380d, 25=751eecfc`

No grouping by cardholder, though the card number was in hand

EVIDENCE

Cell 11's drop list explicitly names `cc_num`, the card number, and removes it. Cell 16 calls `train_test_split` with a `stratify` argument and no `groups` argument. The praxis pack's full-source scan reports `GroupKFold` or any entity-grouped split present in zero of the thirty-six notebooks in this series. And the customer is not actually gone from the feature vector: `lat`, `long`, `zip`, `city`, `city_pop`, `job` and `gender` jointly identify a Sparkov customer with high specificity.

SCIENTIFIC CONSEQUENCE

Dropping `cc_num` removed the ability to group by customer without removing the customer. A cardholder who is defrauded in the training rows reappears in the held-out rows carrying the identical latitude, longitude, job and city population, so the model can learn which simulated people get defrauded rather than which transactions are fraudulent. Grouping was available — you held the key and deleted it — and was not done.

REQUIRED REPAIR

Retain `cc_num` as a grouping key, not as a feature, and use `GroupShuffleSplit` or `GroupKFold` so that every transaction from one card falls entirely on one side. Report the drop from 0.948632. That is twenty lines.

DEFENSIBLE REPLY

The cell-22 ablation shows that with `amt` removed the entire remaining feature set — including all the demographic columns that would carry any customer memorisation — reaches only 0.574302, which bounds how much customer-level leakage can be contributing to the headline at somewhere near nothing.

Four hand-set configurations, no search, and a winner declared

EVIDENCE

Cell 16 hard-codes RandomForestClassifier with n_estimators sixty, XGBClassifier with n_estimators eighty and max_depth six, LGBMClassifier with n_estimators eighty, and LogisticRegression with max_iter three hundred. There is no GridSearchCV, no RandomizedSearchCV, no Optuna, and no nested cross-validation anywhere in the notebook. The praxis pack's scan reports hyperparameter search present in zero of thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

The declared ranking — XGBoost 0.948632 over LightGBM 0.937603 over RandomForest 0.884538 — is a ranking of four arbitrary configurations, not of four algorithms. Sixty trees in a forest is a choice; eighty boosting rounds is a choice; and the gap between the top two is 0.011029, which is comfortably inside the range a single depth or learning-rate change would move either of them. The sentence best model: XGBoost is therefore a statement about a line of code, not about a method.

REQUIRED REPAIR

Nested cross-validation: an inner loop that selects each learner's hyperparameters and an outer loop that estimates its generalisation, so that the number reported for each algorithm is that algorithm at its own best rather than at a default someone typed.

DEFENSIBLE REPLY

Faults fifteen, sixteen and seventeen are one absent pass filed three times.

ORDERED CORRECTION

Merge into 17. Keep the one line the others lack: the gap between XGBoost and LightGBM is 0.011029, comfortably inside what a single depth or learning-rate change moves either model, so the ranking ranks configurations.

A winner declared with no significance test on paired predictions

EVIDENCE

Cell 16 prints the ranking and the line best model: XGBoost. The gap to second place is 0.948632 minus 0.937603, that is 0.011029. There is no McNemar test, no bootstrap over the held-out rows, and no confidence interval anywhere in the notebook — the praxis pack's scan confirms bootstrap or confidence interval present in zero of thirty-six. For scale, the only dispersion figure the notebook does print, the cross-validated standard deviation in cell 24, is plus or minus 0.0221, which is twice the gap it is ranking on.

SCIENTIFIC CONSEQUENCE

All four models are scored on the same four hundred sixty-three thousand and ninety-nine rows, so their errors are paired and correlated, and a paired test is both available and cheap. Without it, the claim that XGBoost beats LightGBM on this corpus has no support. And the consequence is not academic: the winner is the model that fails the accuracy baseline, and the runner-up fails it too, while the only model that passes it is ranked third.

REQUIRED REPAIR

McNemar's test on the paired predictions of the top two models at the chosen threshold, plus a bootstrap confidence interval on each model's held-out AUC over resampled test rows. If the intervals overlap, report a tie and say so.

DEFENSIBLE REPLY

Same missing pass, third filing.

ORDERED CORRECTION

Merge into 17. Preserve the consequence sentence, which is the sharpest in the trio and belongs beside fault 2: the winner is the model that fails the accuracy baseline, the runner-up fails it too, and the only model that passes is ranked third.

Six-decimal point estimates, and one dispersion figure whose scope is undisclosed

EVIDENCE

Cell 16 prints `roc_auc` to six decimals, with the code comment 6 dp: a 1.000000 is a red flag, not a win. Cell 24 prints the three-fold cross-validated ROC-AUC as 0.9261 plus or minus 0.0221. Now read the line above it in the same cell: `cvX, cvy` equals `Xtr` dot `iloc` up to forty thousand. Those folds are cut from the first forty thousand rows of the one hundred and twenty thousand-row training subsample — about 2.2 percent of the corpus, that is forty thousand of one million eight hundred fifty-two thousand three hundred ninety-four. Neither the print line nor section 12 nor the section-13 conclusion states that bound anywhere. Notebook 14 in this same series does state it, in the same words, in its own section 12 and conclusion, so the disclosure was written for this template and is missing here.

SCIENTIFIC CONSEQUENCE

The gap between the cross-validated 0.9261 and the headline 0.948632 is 0.0225, almost exactly one of those standard deviations. A reader given both numbers with no scope statement will read the CV as corroboration of the headline. It is not. It measures fold-to-fold stability of the fit on a forty-thousand-row training slice, and it says nothing about the held-out estimate. The sibling notebook proves the author knew that and wrote it down there.

REQUIRED REPAIR

Bootstrap the held-out AUC over resampled test rows and report the interval, report every headline to the resolution that interval supports rather than to six decimals, and carry the forty-thousand-row bound into the print line itself as notebook 14 does.

A saturation claim asserted in a code comment and never measured

EVIDENCE

Cell 16 caps training at one hundred and twenty thousand rows and carries this comment: we fit on a STRATIFIED subsample because these learners saturate long before then on this data. Cell 16's printed line confirms it — materialized 1,852,394 rows, trained on 120,000, held-out 463,099. There is no learning curve anywhere in the notebook. The word saturate appears once, in that comment, and no cell measures it.

SCIENTIFIC CONSEQUENCE

Six point five percent of the corpus was used for fitting, which is defensible if and only if the score has flattened, and whether it has flattened is an empirical question this notebook does not ask. It matters here more than usual: fraud in the held-out set numbers about two thousand four hundred and thirteen rows, so the training subsample contains on the order of six hundred fraud examples, and six merchant categories return zero recall. Whether those zeros are a feature-set failure or simply too few positives in training is exactly what a learning curve would separate, and it was not drawn.

REQUIRED REPAIR

Plot held-out AUC and per-category recall against training size at ten thousand, thirty thousand, one hundred and twenty thousand, four hundred thousand and one million rows. If the curve is flat, the cap is justified and the claim is now measured. If it is still rising at one hundred and twenty thousand, the headline is a statement about a budget.

Model-internal importances plotted on a feature set the notebook calls memorisable

EVIDENCE

Cell 18 reads best dot feature_importances_ and plots the top twelve as the explanation panel. Cell 2 of the same notebook says: several of these — merchant, city, job — are high-cardinality categoricals a model can simply memorise. There is no permutation importance and no SHAP computation anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A built-in tree importance is computed on the training data, from the splits the model happened to make, and a column with thousands of distinct integer values offers far more candidate split points than gender does. So the panel that is presented as the explanation of the model is measured on the training set and is structurally biased toward exactly the columns the author flagged as memorisation risks.

REQUIRED REPAIR

Permutation importance computed on the held-out set, or SHAP values, reported instead of or alongside the built-in importances, so the explanation is measured where the claim is measured.

DEFENSIBLE REPLY

XGBoost's scikit-learn wrapper reports gain-based importance rather than raw split counts, which is materially less cardinality-biased than classic mean-decrease-impurity, and in any case the cell-22 ablation already establishes the ranking that matters — amt and then nothing.

The one column the model uses is the one the attacker chooses

EVIDENCE

Cell 22 establishes that removing amt takes the model from 0.948632 to 0.574302. Cell 2 states that card fraud is rare and adversarial. There is no perturbation experiment, no evasion budget, and no worst-case recall reported anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every other feature — city, zip, latitude, job, gender — is a property of the victim and cannot be chosen by the fraudster. The transaction amount is the one column the fraudster picks, freely, at no cost, in the moment. So the entire measured capability of this detector rests on the single variable an adversary has complete control over. A fraudster who charges within the ordinary distribution for that merchant category defeats it, and the notebook contains no number describing how much the score falls when they do.

REQUIRED REPAIR

Define a perturbation budget in currency — say, cap each fraudulent transaction at the ninetieth percentile of legitimate amounts for its merchant category — re-score the held-out set under that constraint, and report the worst-case recall. That figure, not 0.948632, is the detector's operational value against an opponent.

DEFENSIBLE REPLY

You are asking me for the score when the attacker neutralises `amt`. I printed it: 0.574302, in cell twenty-two, and my conclusion calls it a genuine single-feature shortcut and tells the reader not to treat the headline as broad detection capability.

ORDERED CORRECTION

Retain only what fault 1 does not carry: the asymmetry of control. Every other column — city, zip, latitude, job, gender — is a property of the victim, and `amt` is the one variable the fraudster sets freely at zero cost, so the collapse in cell 22 is not merely a shortcut finding but an evasion budget of zero. State it as one sentence appended to fault 1 rather than as a separate D2 charge demanding an experiment whose answer is already in the table.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 463,099 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Closing. The rubric total is five out of thirty, which puts this in the bottom band: the reported quantity is an artifact of the file. And I want to be precise about why that verdict is harsher than the friendly audit's five and a half out of ten rather than merely lower. That audit graded whether the notebook told the truth about itself, and by that standard it did — it ran the ablation, it printed the collapse, and it said in plain words that the headline is a single-feature shortcut. This rubric grades whether the scientific claim survives, and here the notebook refutes its own headline in cell 22 and then prints that headline as the result in cell 16 and names its author the winning learner in section 13. Disclosure is not mitigation. The one comparison this notebook actually performed — the winner against its own majority-class baseline — it got backwards, and every axis that would have caught that was left at zero.

Large-Scale Financial Transaction Fraud Detection

Kaggle aryan208 financial-transactions-dataset-for-fraud-detection — a synthetic, community-generated five-million-row transaction file, of which the leading one million five hundred thousand rows are read, at a fraud rate of two point two six percent.

HEADLINE

LightGBM, held-out ROC-AUC 0.748990, on a corpus the notebook's own audit grades A for data trust.

DEFENSE
F
Selected-model estimand

DOCTORAL
7
/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC
7
/10 · self A

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 0 Honesty / scope
22 live faults	3 critical (FATAL)	19 major (CONTESTED)	0 withdrawn after refutation		

WHAT THIS NOTEBOOK GETS RIGHT

I want to begin by saying that this is the most intellectually honest notebook in the set of four I was given, and it is not close. It diagnoses its own failure correctly, from its own printed numbers, and it puts the verdict INCONCLUSIVE in the abstract rather than burying it. It even explains why its own audit is structurally incapable of catching the defect it contains. That is real scientific work, and it is why the rejection that follows is a rejection of a decision, not of an understanding.

EXAMINER RULING

The quantity you reported is an artifact of a fill value in your loader, and you wrote down the one line of code that would have proved it and did not run it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

1

CRITICAL

UNTESTED

VERIFIED

SEPARABILITY MISREAD — THE SHAPE OF THE EVIDENCE

ORAL TOP 5

A coin-flip linear model beside trees at three quarters

EVIDENCE

Cell 16 prints four learners on one split. LogisticRegression scores ROC-AUC 0.501583 — zero point five zero, a coin flip. The three split-based learners reach 0.725762, 0.746834 and 0.748990 — call that three quarters. Cell 20 then prints the strongest single feature at AUC 0.5171, and says every one of the nine features scores below zero point five two on its own. And the independent re-derivation run for this review takes the held-out AUC from 0.7503 to 0.5030 by dropping one column: time since last transaction.

SCIENTIFIC CONSEQUENCE

A logistic regression on scaled inputs can use each column only monotonically — more of this, more fraud. At 0.501583 it finds nothing. Trees find three quarters. Structure that only a split-based learner can reach, in a column that no rank statistic can see, is not a behavioural relationship. It is a cut at one particular value. And your loader manufactures exactly one such value: cell 11 ends with fill-not-a-number with zero point zero, which turns every absent reading of time since last transaction into the sentinel zero. The model is answering the question was this field ever recorded. That is bookkeeping, not fraud detection.

REQUIRED REPAIR

Run the line you yourself wrote in section 13 — group by the label and take the missingness rate of time since last transaction on each side. If those two rates differ, drop the column, add an explicit was-missing indicator, and re-run every number in the notebook against it. That is one line to diagnose and three to fix.

2

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

ORAL TOP 5

You wrote the check and did not run it

EVIDENCE

Section 13, under the heading the most likely explanation, states the hypothesis and then prints the check verbatim: group the dataframe by y, take time since last transaction, and apply the mean of is-not-a-number. The section 3 provenance note flags the same defect a hundred lines earlier. The conclusion then says, in its own words, this notebook never measures that, so it is not reported as a result.

SCIENTIFIC CONSEQUENCE

At doctoral level, naming a fatal limitation is not a defence. It is a confession. You identified the experiment that decides whether your headline is a fraud detector or a missingness ledger, you costed it at one line, you handed the code to the reader, and you published the number anyway. Every sentence of disclosure in this notebook is correct and not one of them changes what the reported quantity is. The disclosure paragraph costs nothing and buys nothing.

REQUIRED REPAIR

Run it. Then either the hypothesis is refuted and the headline stands with an explanation, or it is confirmed and the headline is withdrawn. Publishing a notebook that contains the question, the method, and the admission that the method was not applied is not a defensible state for a doctoral artefact.

The winner is one transaction below always-guess-legitimate

EVIDENCE

Cell 18 prints that the held-out set contains 366,543 benign transactions — three hundred and sixty-six thousand — out of 375,000. Divide those: the true majority-class accuracy is 0.977448. That is exactly the figure RandomForest and LogisticRegression print in cell 16, because they predict legitimate for every row. LightGBM prints 0.977445 — three hundred and sixty-six thousand five hundred and forty-two correct, one transaction fewer than the trivial rule. XGBoost prints 0.977419, which is eleven fewer. The MajorityBaseline row prints 0.977400, and it prints that only because the code in cell 16 applies round to four decimal places to the baseline row while every model row gets round to six.

SCIENTIFIC CONSEQUENCE

A reader scanning that accuracy column sees the winner at nine seven seven four four five sitting above a baseline at nine seven seven four zero zero and concludes the model clears the bar. It does not. It is below it. The gap is a rendering artifact of two different rounding calls three lines apart in the same cell. You built the baseline row precisely so a reader could make that comparison, and then you formatted it so the comparison comes out wrong.

REQUIRED REPAIR

Round every row in the ranking table to the same precision, and print the raw correct-classification counts beside the rates so the comparison is on integers. Then flag any learner at or below the baseline as a diagnostic, not as a row in a ranking.

DEFENSIBLE REPLY

The table is ranked by ROC-AUC, not accuracy; section 13 prints all four accuracies at six decimals and states 'The accuracy column carries no information whatsoever' and that the model's deployed behaviour 'is exactly the 0.9774 baseline'. A one-transaction gap in 375,000 rows is not an ordering, and the notebook's stated reading -- the model equals the trivial rule -- is the substantively correct one.

ORDERED CORRECTION

Drop 'the winner is one transaction below always-guess-legitimate' as the headline of the charge, since one transaction in 375,000 is noise. Keep only the rendering defect: two different rounding calls three lines apart in the same cell make the winner appear to clear a baseline it is indistinguishable from.

Atlas proof

docs/course/security-ml-notebooks/assignments/evidence/SEAS8414-praxis-findings-pack.html · nb17, criterion 3

Sender account existed and was never used as a group

EVIDENCE

The source file carries eighteen columns, and among them are sender account and receiver account. Your cell 11 drop list names both, so you removed them as features. Cell 16 then calls train-test-split with stratify on y and nothing else. There is no GroupKFold and no GroupShuffleSplit anywhere in the notebook — nor, I am told, anywhere in the thirty-six notebooks of this series.

SCIENTIFIC CONSEQUENCE

Fraud is committed by accounts, not by rows. When one account's transactions are scattered across the split, the model does not need to learn what fraud looks like; it needs to learn what that account looks like, and the behavioural columns — spending deviation, velocity, geo anomaly — are precisely the columns that hold an account nearly constant. Your removal of sender account from the feature matrix proves you understood the memorisation risk. Removing an identifier stops the model keying on the identifier. It does not stop one account straddling the split.

REQUIRED REPAIR

GroupShuffleSplit on sender account. Report the held-out AUC under the grouped split beside the random one, and report the drop. It is twenty lines, and the grouping key was in the file you already read.

DEFENSIBLE REPLY

sender_account exists and is dropped, so the grouped split was constructible -- and I built it. It returns 0.749502 against the random 0.748990. There is no drop to report. The corpus averages 2.06 transactions per sender (median 2, max 11) across 729,834 accounts, and the nine retained features carry no account-identifying information, so there is no account to memorise. The entire held-out score is one column's missingness bit: drop it and the model falls to 0.503850, with or without grouping.

ORDERED CORRECTION

Delete the causal claim that 'the behavioural columns hold an account nearly constant' and that the model 'needs to learn what that account looks like' -- both are refuted by the measured grouped-split AUC of 0.749502. The surviving charge is narrower and methodological: a Tier-1 group key was in hand and the split's independence was never validated, even though validating it costs the twenty lines the fault correctly estimates.

The winner's margin is inside your own fold spread

EVIDENCE

Cell 16 puts LightGBM at 0.748990 and XGBoost at 0.746834. The margin is 0.002156 — about two thousandths. Cell 24 prints the three-fold cross-validated AUC of that same winner as 0.7459 plus or minus 0.0038 — nearly four thousandths. The separating margin is smaller than the fold-to-fold spread you printed yourself, and the whole notebook is then written about LightGBM.

SCIENTIFIC CONSEQUENCE

You have no evidence that LightGBM is the better model on this corpus. You have evidence that it drew the higher number once. Every downstream result — the importance chart, the ablation, the operational false-positive rate, the per-family recalls — is computed on a model selected by a difference you cannot distinguish from noise. The comparison is the notebook's stated purpose and it does not support the selection it makes.

REQUIRED REPAIR

A McNemar test on the paired held-out predictions of the two candidates, or bootstrap confidence intervals on the two AUCs with the overlap reported. If the intervals overlap, say so and carry both models forward.

DEFENSIBLE REPLY

0.748990 and 0.746834 are scored on the same 375,000 rows, so their difference is paired and its standard error is far smaller than either model's marginal spread. Hanley-McNeil puts the marginal SE of an AUC of 0.749 on 8,457 positives and 366,543 negatives at about 0.0031; at a between-model correlation of 0.9 the paired SE of the difference is about 0.0014, so a 0.002156 margin is roughly 1.5 sigma -- borderline, not obviously noise. The honest statement is that no test was run, not that the margin is inside a spread that measures a different quantity.

ORDERED CORRECTION

Replace 'the winner's margin is inside your own fold spread' with the correct mechanism: the ranking is declared without a paired test, and the appropriate instruments are DeLong on the two correlated AUCs or McNemar on the paired predictions -- both of which the 'required' section already names.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

The clock was in the file and you deleted it

EVIDENCE

The source CSV's second column is timestamp. Your cell 11 drop list names it explicitly, alongside transaction id, IP address and device hash. So the field existed, you read the file, and you removed it. The split in cell 16 is random. There is no time-ordered holdout in the notebook.

SCIENTIFIC CONSEQUENCE

A random split lets the model train on next month and test on last month. Fraud typologies turn over; a detector that scores well only when it has seen the future is not a detector. The availability of the temporal split is proven by the very line that discards the column — you cannot argue you did not have the clock, because you had to hold it in order to throw it away.

REQUIRED REPAIR

Sort by timestamp, train on the earlier portion, test on the later portion, and report the drop against the random split. If the drop is large, that number, not 0.748990, is your result.

DEFENSIBLE REPLY

timestamp exists and is dropped, so I built the temporal holdout. It returns 0.749645 against the random 0.748990. There is no drop. The premise fails empirically: the monthly fraud rate over the thirteen months present runs 0.02215 to 0.02325 -- a 5% relative band that is pure sampling noise -- and the file is not even time-ordered, since consecutive rows carry August, August, May and October timestamps. Fraud typologies do not turn over in a corpus drawn i.i.d. from one stationary generator.

ORDERED CORRECTION

Delete 'a random split lets the model train on next month and test on last month' and 'fraud typologies turn over' -- neither holds on this generator, and the measured temporal-split AUC is 0.749645. The surviving charge is that a Tier-1 clock was discarded without the one-line check that would have shown the distribution to be stationary.

You drew the right curve and refused to read it

EVIDENCE

The positive rate printed in cell 11 is 0.0226 — two point three percent. Cell 18's code calls precision-recall-curve and plots it as the third panel. Nowhere in the notebook is average-precision-score called, no area under that curve is printed, and no model is ranked by it. That holds across all thirty-six notebooks in this series: the curve is drawn thirty-six times and its area is computed zero times.

SCIENTIFIC CONSEQUENCE

At two percent positives, ROC-AUC is dominated by the ranking of the overwhelming negative mass and moves very little when false positives multiply. Precision-recall is the diagnostic that prices them. You computed the curve, you rendered it, and then you selected your winner, wrote your conclusion and assigned your trust grade using a number that cannot see the quantity the curve was drawn to show. I am not saying you omitted the diagnostic. I am saying you performed it and declined to read it.

REQUIRED REPAIR

Average precision with a bootstrap confidence interval as the headline, and precision reported at a fixed recall — say, at recall zero point five — so a reader can see what catching half the fraud costs.

Every honest number depends on an unexamined constant

EVIDENCE

Cell 18 sets pred as probability greater than zero point five. Everything downstream comes from that line: the confusion matrix, the operational false-positive rate of 0.0000 with one benign transaction flagged out of 366,543, and the four per-transaction-type recalls of zero point zero. There is no cost matrix in the notebook and no threshold selected from a curve.

SCIENTIFIC CONSEQUENCE

For a task with asymmetric costs, the operating point is the result. A missed fraud and a frozen legitimate card are not the same event and do not carry the same price. One half is not a decision; it is the default argument of a comparison operator. Your section 13 concludes from these figures that the model is not a detector at all, and that conclusion is correct — but it is a conclusion about one arbitrary threshold, not about the model.

REQUIRED REPAIR

Write down the cost of a missed fraud and the cost of a false alarm, in the same units, and choose the threshold that minimises expected cost. Then report recall and alert volume at that threshold.

DEFENSIBLE REPLY

A cost-optimal threshold is a statement about a detector, and the notebook establishes there is no detector. I ran the curve: catching half the fraud costs 96,368 alerts at precision 0.0439; catching 95% costs 182,942 at precision 0.0439. Every operating point on this model prices the same single bookkeeping bit, so selecting one by expected cost would dress an artifact in operational clothing rather than expose it.

9

MAJOR

UNTESTED

DOWNGRADED

NO UNCERTAINTY ON THE HEADLINE

A point estimate carrying an entire notebook

EVIDENCE

The headline 0.748990 is a single number from a single split at a single seed. Cell 24's 0.7459 plus or minus 0.0038 is a fold-to-fold spread computed inside the training subsample, on the first forty thousand of its one hundred and twenty thousand rows. No bootstrap appears anywhere in the notebook, and none appears in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

A fold-to-fold standard deviation on a slice of the training data is not a standard error on the held-out estimate. They answer different questions and they are not interchangeable. So the notebook has no statement at all about how much 0.748990 would move under resampling — which means it cannot support any claim of the form this is higher than that, and it makes several.

REQUIRED REPAIR

Bootstrap the held-out set, one thousand resamples, and report the AUC with a ninety-five percent interval. Report every comparison as an interval on the difference.

DEFENSIBLE REPLY

Hanley-McNeil gives the standard error of an AUC of 0.749 on 8,457 positives and 366,543 negatives as about 0.0031; the notebook's printed 3-fold spread of ± 0.0038 is within a factor of 1.2 of that. The bootstrap the fault requires returns approximately what is already printed, so the defect is presentational, not evidential -- and the notebook's disclosed-limitations paragraph already labels the figure as scale for re-split noise rather than as a standard error.

10

MAJOR

MEASURED

DOWNGRADED

FALSE PRECISION

Six decimals on a quantity you never resampled

EVIDENCE

Cell 16 rounds every model's AUC to six decimals, with the comment that a one point zero zero zero zero zero zero is a red flag, not a win. So 0.748990 is reported to the millionth. The notebook's own nearest measure of variability, the fold spread in cell 24, is 0.0038 — thirty-eight ten-thousandths, roughly four thousand times the resolution being printed.

SCIENTIFIC CONSEQUENCE

Six decimals is a claim about resolution. It tells the reader that the fourth, fifth and sixth digits carry information. They do not; they are below the noise floor of the estimator by three orders of magnitude. The cost of that is not cosmetic — it is precisely what makes the two-thousandth margin over XGBoost in fault five look like a result.

REQUIRED REPAIR

Report to the resolution the estimator supports. On this evidence that is two decimal places, written as zero point seven five with an interval.

DEFENSIBLE REPLY

The six decimals are a deliberate leakage tripwire, not a resolution claim: a score that rounds to 1.000000 must be visible as such rather than absorbed into 1.00.

Saturation asserted in a comment, never measured

EVIDENCE

Cell 16 caps the training set at 120,000 and the comment on that line reads: because these learners saturate long before then on this data. There is no learning curve in the notebook. Cell 11 also carries an assert that the row count is at least one million — an honesty gate on the size of the download, applied to a corpus from which eight percent of the training half is actually used.

SCIENTIFIC CONSEQUENCE

You have justified a design decision with an empirical assertion and provided no measurement of it. Saturation is not a property you may assume; it is a curve you plot. And on a fraud corpus with a two percent positive rate, the training subsample contains only a few thousand positives, so the claim that more data would not help is exactly the claim most likely to be wrong. The one-million-row assert then certifies the quantity you did not use.

REQUIRED REPAIR

A learning curve — held-out AUC against training size at ten thousand, thirty thousand, one hundred thousand, three hundred thousand, one million — with the plateau shown. If it has not plateaued, train on more. Otherwise state the effective sample size in the abstract instead of the download size.

DEFENSIBLE REPLY

Cell 16 prints 'materialized 1,500,000 rows | trained on 120,000 (stratified subsample) | held-out 375,000' and the code comment states in capitals that every headline is a subsample number. And the saturation question is moot on this corpus: the headline is one column's missingness bit, whose AUC is 0.7489 at 120,000 training rows under random, grouped and temporal splits alike -- more rows cannot move a quantity that is already a single bit.

Grade A on a corpus whose headline is an artifact

EVIDENCE

Cell 20 grades the single-feature check with cutoffs at zero point eight five, zero point nine five, zero point nine nine and zero point nine nine nine. The observed best single-feature AUC is 0.5171. The nearest cutoff is zero point eight five, a third of the scale away. The grade prints A. Meanwhile the re-derivation for this review shows one column carrying the held-out AUC from 0.7503 down to 0.5030 — an artifact worth roughly a quarter of an AUC point, passing through an instrument whose first trip-wire sits at zero point eight five.

SCIENTIFIC CONSEQUENCE

The letter is not a measurement of anything a reader cares about. Your own section 13 says so: the grade would still return A even if the hypothesis were confirmed. An instrument that returns its best grade on a corpus whose headline is a fill-value artifact is not conservative or stable — it is measuring a different quantity from the one it is named after, and the name is doing the work.

REQUIRED REPAIR

Report the quantity with an interval and delete the letter. If a triage grade is wanted, the question it must ask is not whether any one column is near-perfect but whether any one artifact accounts for the whole headline — which is a drop-and-refit on the model's own top feature, not a per-column rank statistic.

DEFENSIBLE REPLY

The abstract states 'the audit passes, the headline does not' and 'the reported grade A is therefore a statement about the audit's coverage rather than about the data'; section 13 prints 'Corrected data-trust grade: not established' and 'The audit found nothing and there is nothing to find are different claims, and only the first is supported'. The letter is not doing the work the charge says it is doing, because the notebook removes it.

ORDERED CORRECTION

Replace the re-derived figure '0.7503 down to 0.5030' with the notebook's printed 0.748990 falling to 0.503850. Retarget the charge from 'the name is doing the work' -- which section 13 refutes -- to the narrower true point: the printed cell output still emits '= > data trust grade: A' with no in-cell qualification, so a reader who consumes the outputs without the prose is misled.

Zero alerts, zero catches, no price on either

EVIDENCE

Cell 18 prints one benign transaction flagged out of 366,543, and per-transaction-type recalls of zero point zero for payment, transfer, withdrawal and deposit. Subtract: 375,000 held-out rows minus 366,543 benign leaves 8,457 fraudulent transactions in that held-out set — eight thousand four hundred and fifty-seven — and every family recall prints as zero. Sommer and Paxson is in your reference list. There is no alerts-per-day figure anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The operational question for a fraud system is how many analyst-hours an operating point costs and how much fraud it catches for them. At one half you have priced neither, because at one half the system does nothing: it raises one alert and misses eight and a half thousand frauds. Any threshold that recovers those frauds raises a false-alarm volume you have never computed, and the precision-recall curve that would give it to you is in panel three of cell 18, unread.

REQUIRED REPAIR

Take the precision-recall curve you already computed, choose the operating point at a stated recall, and multiply its false-positive rate by a stated daily transaction volume. Print alerts per analyst per day beside the recall.

DEFENSIBLE REPLY

Cell 18 prints the operational false-positive rate, the flagged count, the benign denominator and the per-family recalls. What is missing is an alert volume at a USEFUL recall, not an alert volume at any threshold, and the charge as written says the latter.

ORDERED CORRECTION

Reword the defect to 'no alert volume is computed at any threshold that catches fraud', and supply the number the charge asks for, which I measured: catching half the fraud costs 96,368 alerts at precision 0.0439, and catching 95% costs 182,942 -- against 8,457 frauds in the held-out set.

Adversarial in section one, untested in section thirteen**EVIDENCE**

Section 1 states the task in these words: fraud is rare and adversarial. The features are amount, velocity score, spending deviation score, geo anomaly score, merchant category, device used and time since last transaction. No perturbation, no evasion, no worst-case recall under a budget appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every one of those columns is under the attacker's control at low cost. A fraudster who splits a transfer, waits, or moves device changes velocity, time since last transaction and device used directly. If your model's discriminative power lives in a missingness sentinel, as fault one argues, an attacker who simply ensures a prior transaction exists defeats it entirely and for free. You called the domain adversarial and then measured only the non-adversarial case.

REQUIRED REPAIR

State a perturbation budget in the units of the features — so many minutes of delay, so many transactions of splitting — search for the cheapest perturbation that flips the prediction, and report worst-case recall under that budget beside the average-case number.

DEFENSIBLE REPLY

spending_deviation_score, velocity_score and geo_anomaly_score are sampled by a generator, not measured from behaviour, so 'the cheapest perturbation that flips the prediction' has no cost in the world. The professor's own evasion argument -- ensure a prior transaction exists and the missingness bit vanishes -- is correct and is an argument, not a missing measurement; and it is already implied by section 13's proposed fix of adding an explicit was_missing indicator.

15

MAJOR

UNTESTED

DOWNGRADED

IMPURITY IMPORTANCE REPORTED AS EXPLANATION

The central argument rests on an unprinted bar chart

EVIDENCE

Cell 18 plots best-dot-feature-importances-underscore — the split-based impurity importance — as a horizontal bar chart. Section 13 then asserts that time since last transaction ranks first in that chart and that device used ranks eighth of nine. Neither ranking appears as text anywhere in the notebook; both are visual readings of a figure.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward continuous, high-cardinality columns and against low-cardinality categorical ones. Time since last transaction is continuous; device used is a small categorical. That is exactly the direction of the known bias, so the very ranking that carries your argument is produced by the estimator least entitled to produce it — and it is not written down, so a reader cannot check it without re-running you.

REQUIRED REPAIR

Permutation importance on the held-out set, or SHAP values, printed as a table with numbers. Any claim of the form this column ranks first must appear as text, not as a bar someone has to measure with their eye.

DEFENSIBLE REPLY

The two rankings section 13 reads off the chart are both correct -- I refit the model and printed them. The known continuous-versus-categorical bias of impurity importance pushes in the direction that would have made the argument HARDER to find, not easier, and it did not mislead.

16

MAJOR

UNTESTED

DOWNGRADED

NO MODEL SELECTION

Defaults throughout and cross-validation on a slice

EVIDENCE

Cell 16 instantiates all four learners at hand-set constants, with fixed tree counts — sixty for RandomForest, eighty for XGBoost and LightGBM. There is no grid, no random search and no nested cross-validation. Cell 24 then runs three-fold cross-validation on cvX equals Xtr dot iloc up to forty thousand — the first forty thousand rows of the training subsample, a third of what the headline model saw.

SCIENTIFIC CONSEQUENCE

The comparison between four learners at four sets of arbitrary defaults is a comparison of defaults, not of learners, and the reported 0.7459 plus or minus 0.0038 describes a model fitted on a third of the training data, so it is not a cross-validated estimate of the headline pipeline. The notebook presents it beside the headline as if it corroborated it.

REQUIRED REPAIR

Nested cross-validation — an inner loop that selects hyperparameters, an outer loop that estimates performance — and cross-validation run on the same data the headline model is fitted on.

DEFENSIBLE REPLY

Untuned defaults applied uniformly are a fair comparison of learners on a fixed budget, and no hyperparameter search reaches a different answer when the entire signal is one column's missingness -- every tuned tree finds the same bit.

One seed, used everywhere, by the code's own admission

EVIDENCE

Cell 10 sets RANDOM-STATE equals zero, and the comment beside it reads, in the notebook's own words, single seed used everywhere. That seed drives the train-test split, the stratified subsample, all four model fits and the ablation's re-splits. The audit sample in cell 20 uses a second fixed seed, one. Nothing is repeated.

SCIENTIFIC CONSEQUENCE

Seed stability was probed for the single-feature audit and never for the reported model, so there is no evidence about how much 0.748990 moves under a different draw. The re-derivation for this review found that the audit's own selected column changes five times across eight seeds, which shows that in this notebook the seed is not a formality.

REQUIRED REPAIR

At least ten seeds for the headline split and fit, with mean and standard deviation reported in place of the single value.

DEFENSIBLE REPLY

The seed drives the reported number only through a 375,000-row held-out split, on which the AUC is stable to the third decimal across random, grouped and temporal partitions alike (0.748990, 0.749502, 0.749645).

ORDERED CORRECTION

The stronger form of this charge is not seed instability but argmax fragility, and it should be promoted: the ablation's target column is chosen by argmax over nine single-feature AUCs spanning 0.5010 to 0.5171 (device_used 0.5171, geo_anomaly_score 0.5160, spending_deviation_score 0.5133), all within 0.016 of each other and all within noise of chance. The instrument that decides what gets ablated is selecting essentially at random. The claim that 'the audit's own selected column changes five times across eight seeds' is a re-derivation figure I could not verify and should be replaced by the printed spread, which makes the point without needing it.

The encoder sees the test set before the split exists

EVIDENCE

Cell 11 loops over the object columns and calls LabelEncoder dot fit-transform on the full frame. The categorical columns in this file include transaction type, merchant category, location, device used and payment channel, so the loop is active here — this is not a no-op. The split is not drawn until cell 16. The conclusion discloses this and correctly says no label information leaks.

SCIENTIFIC CONSEQUENCE

The encoder's category vocabulary is fitted on rows that will become the test set. No label leaks, and I accept that. But the protocol is not the one a deployed system can run, because a deployed system meets categories it has never seen and this pipeline has no bucket for them. The reported number therefore describes a setting that cannot exist in production.

REQUIRED REPAIR

Fit the encoder inside the training fold, with an explicit unseen-category bucket, and report the held-out score under that protocol.

DEFENSIBLE REPLY

LabelEncoder assigns integers by sorted category order and never consults the label, so fitting it on the training fold instead would produce a different but equally arbitrary mapping. On a shuffled split over 1.5 million rows every test category is present in train with certainty, so the measured effect is nil. The notebook discloses the transductive step precisely and correctly identifies the only real consequence, which is the absence of an unseen-category bucket in deployment.

Zero point zero zero zero from fifty thousand rows

EVIDENCE

Cell 20 measures contamination against the first fifty thousand held-out rows — one seventh of the 375,000 in the held-out set — and prints 0.000. The duplicate rate over the whole corpus also prints 0.000. There is no confidence interval on either.

SCIENTIFIC CONSEQUENCE

A zero from a sample is not a zero. It is an observation consistent with a true rate anywhere below roughly one in seventeen thousand, and the grade cutoffs the notebook applies — five percent for a B, fifteen percent for a C — are read off that unqualified figure. On this corpus the true rate does appear to be zero, so no harm is done here; the defect is that the notebook's method could not have told the difference.

REQUIRED REPAIR

A random sample with a binomial confidence interval on the contamination rate, or an exact computation over the full held-out set, which on 375,000 rows is affordable.

DEFENSIBLE REPLY

The disclosed-limitations paragraph already says 'Train/test overlap is measured on the first 50,000 held-out rows, so read it as a sampled estimate', and the whole-corpus duplicate rate of 0.000 makes a non-zero contamination rate arithmetically near-impossible, since contamination requires duplicate feature vectors.

An assert that certifies quantity on a generated file

EVIDENCE

Cell 11 contains an assert that the frame has at least one million rows, described as an honesty gate. Section 3 states plainly that the corpus is synthetic, community-generated, with no peer-reviewed origin, and that engineered columns like geo anomaly score may encode the label. The abstract calls it a large-scale methods sandbox rather than a validated benchmark.

SCIENTIFIC CONSEQUENCE

Scale on a generated file is not evidence. The assert certifies a property — row count — that is free to the generator and uninformative about the phenomenon, while the property that matters, whether the engineered scores encode the label, is flagged in prose and never tested. A gate that guards the cheap quantity and waves the expensive one through gives a false impression of rigour.

REQUIRED REPAIR

Ablate the engineered anomaly columns as a group — geo anomaly score, spending deviation score, velocity score — and report the held-out AUC on the raw behavioural columns alone. That is the measurement the section 3 caveat calls for.

DEFENSIBLE REPLY

The fault asks me to ablate geo_anomaly_score, spending_deviation_score and velocity_score to see whether the engineered scores encode the label. I ran the complement, which is stronger: with time_since_last_transaction removed the model retaining all three engineered scores reaches 0.503850, i.e. chance. The engineered columns encode nothing at all, so the section 3 caveat the fault says was never tested is answered in the negative.

ORDERED CORRECTION

Replace the required ablation with its measured outcome: the engineered anomaly scores carry no signal (the model without time_since_last_transaction scores 0.503850 despite retaining all of them), so the live artifact is the loader's sentinel, not the generator's scores.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 375,000 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

This notebook is the concrete case. Its `LogisticRegression` returns ROC-AUC 0.501583 at accuracy 0.977448, against a majority-class baseline of 0.9774 — a ranking at chance and an accuracy equal to the majority rule. Whether that is a converged fit on features carrying no linear signal, or a fit that failed, cannot be determined from the page, because cell 10 guaranteed a `ConvergenceWarning` could not print. That is the cost of the filter, stated concretely.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

Closing. So the ruling is this. Everything you understood, you understood correctly, and you wrote it down with a clarity I would be pleased to see in a published paper. But a doctoral artefact is judged on what it establishes, not on what it suspects, and this one establishes nothing about fraud. It reports a number that its own author believes is a ledger of which rows had a field recorded, and the line of code that would decide the question is printed in section 13 and was never run. Go and run it, and bring me back either a headline or a retraction.

Malicious URL Detection (Phishing / Defacement / Malware)

Kaggle moutasmtamimi malicious-url-detection-dataset-enhanced-2026 — an aggregated community dump of 632,844 URLs described by 84 lexical, host and live-site features, one third of them malicious.

HEADLINE

XGBoost, held-out ROC-AUC 0.983724, with a data-trust grade of B and an ablation the conclusion says refutes a single-feature story.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 2 critical (FATAL)
- 21 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook has the cleanest bookkeeping of the four I was given. Every figure in its conclusion traces to a printed cell, it refuses to concatenate the three older versions of the corpus that sit in the same archive, and it explicitly declines the vindication reading of its own flat ablation. It also says, without being asked, that its result does not show the signal would transfer to URLs from a different feed. All of that is right, and none of it is what I am about to reject.

EXAMINER RULING

Your headline measures how well one scrape of one aggregation separates itself, and the grade you assigned that scrape turns on the third decimal place of a fifty-thousand-row sample.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The grade turns on the third decimal of a sample

EVIDENCE

Cell 20 prints train-test contamination of 0.149 and a contamination sub-grade of B. The grading line in that same cell reads: C if contamination is greater than or equal to 0.15. So 0.149 gives a B and 0.150 gives a C, and the distance from your printed value to the boundary is 0.001 — one thousandth. The independent re-derivation run for this review measured the same quantity on two different fifty-thousand-row windows and got 0.1489 and 0.1476. Those two windows differ from each other by 0.0013, which is larger than the 0.001 that separates your published letter from the next one down. And the conclusion states plainly that this sub-check, not the single-feature check, drives the overall grade.

SCIENTIFIC CONSEQUENCE

The letter B on the front of this notebook is not a property of the corpus. It is a property of which fifty thousand rows happened to be looked at. A quantity whose resampling spread straddles its own decision boundary carries no information at that boundary, and yet this letter is the summary a reader takes away and the notebook's own conclusion nominates it as the headline verdict on data trust. You have published a categorical judgement produced by a coin whose bias you never measured.

REQUIRED REPAIR

Report the contamination rate with a binomial confidence interval — on fifty thousand draws that interval is roughly plus or minus three thousandths, which already crosses the cutoff — and then delete the letter. If a categorical summary is wanted, it must be assigned from the interval, not from the point estimate, and it must say straddles the B and C boundary when it does.

Host was column fourteen and you never grouped on it

EVIDENCE

The canonical CSV this notebook reads carries a hundred columns, and the fourteenth of them is host. Your cell 11 drop list names it, so you removed it from the feature matrix. Cell 16 then calls train-test-split with stratify on y and nothing else. There is no GroupKFold and no GroupShuffleSplit in this notebook, nor in any of the thirty-six in this series.

SCIENTIFIC CONSEQUENCE

A phishing campaign registers dozens of URLs on a handful of hosts with one naming convention. Under a random split those URLs land on both sides, and the model does not have to learn what maliciousness looks like — it has to learn what that campaign's naming convention looks like, which it can do from the length, the entropy, the hyphen count and the subdomain depth. That is memorisation of a campaign presented as detection of a class. Your own literature review names this exact failure: the caveat you wrote beside blacklist-plus-lexical features reads poor generalization to new campaigns. You wrote the failure mode down and then built the split that hides it.

REQUIRED REPAIR

GroupShuffleSplit on host. Report the held-out ROC-AUC under the grouped split beside the random one, and report the drop. The grouping key was column fourteen of the file you already loaded.

Fifteen columns record a crawler's visit, not a URL

EVIDENCE

The corpus carries fifteen columns whose names begin with web-underscore: web http status, web is live, web external ratio, web unique domains, web favicon, web content security policy, web x-frame, web strict transport security, web x-content, web security score, web forms count, web password fields, web hidden inputs, web has login and web SSL valid. It also carries a reachability class column. None of these are in your cell 11 drop list, so all of them are features. Your own conclusion names them: the rest are host and live-site probes of that same site.

SCIENTIFIC CONSEQUENCE

A phishing site that has been reported and taken down answers a crawler differently from a benign site that has not. Web is live and web http status therefore encode takedown latency — a property of when the crawler ran relative to when the abuse was reported — and takedown latency is correlated with the label through the labelling pipeline, not through anything the attacker did. A model using those columns is reading the corpus's collection history. At deployment there is no such history: the URL arrives in an email before anyone has reported it, and it is live. So the operational value of those columns is not merely lower than measured, it is inverted.

REQUIRED REPAIR

Ablate the fifteen web-underscore columns and the reachability class as a group, retrain, and report the held-out ROC-AUC on the URL-string features alone. That number, not 0.983724, is the one that describes classifying a URL you have not yet visited.

DEFENSIBLE REPLY

The URL-string-only number you are asking for is 0.981104 against the headline's 0.983724 -- a difference smaller than my own re-split noise. Those sixteen columns alone reach only 0.678500, so they are real but wholly redundant with the lexical features. Your claim that their operational value 'is inverted' is a claim about a quantity worth seven ten-thousandths of the score.

ORDERED CORRECTION

Delete 'That number, not 0.983724, is the one that describes classifying a URL you have not yet visited' and 'the operational value of those columns is not merely lower than measured, it is inverted' -- the measured URL-only score is 0.981104. The surviving charge is narrower: sixteen deployment-unavailable columns were kept as features and the one-cell ablation that would have retired the concern was never run.

Thirty-three millionths, against a noise floor never measured

EVIDENCE

Cell 22 prints the headline at 0.983724 and the drop-the-top-feature row at 0.983691. The difference is 0.000033 — thirty-three millionths. The conclusion reads that difference as: the ablation refutes a single-feature story, so the separability is multi-feature. But the ablation function re-splits and refits, and the notebook never repeats it to see how much it moves on its own. Notebook 20 in this same series runs the identical ablation code and prints a discrepancy of 0.000192 between two rows computed on data with zero rows removed — that is, on identical data. Its own conclusion calls that figure re-split noise.

SCIENTIFIC CONSEQUENCE

The claim under test is that a difference is small. You cannot establish that a difference is small using an instrument whose noise you have not bounded, and the sister notebook running your exact code shows that instrument's noise is roughly six times the effect you are reporting. So 0.000033 is not evidence that dropping abnormal-url costs nothing. It is a number indistinguishable from the machinery's own jitter, and the conclusion's central inference is drawn from it.

REQUIRED REPAIR

Fix one held-out split across every ablation arm, refit only the model, and repeat each arm at least ten times. Report each arm as a mean with a spread, and read the difference against that spread rather than against zero.

DEFENSIBLE REPLY

The sentence 'the ablation refutes a single-feature story' does not rest on 0.000033. It rests on the printed best single-feature AUC of 0.7937 against a headline of 0.9837, which already forecloses any single-feature account. The ablation row is corroborative decoration, and the same conclusion paragraph explicitly states that 'each ablation re-splits and refits, so tiny differences are re-split noise'.

ORDERED CORRECTION

Strike 'the load-bearing sentence of the conclusion' -- the load is carried by the 0.7937 single-feature AUC printed three sentences earlier. Replace the borrowed nb20 noise figure of 0.000192 with a direct measurement on this notebook: five re-splits of the same pipeline give std 0.000252 and range 0.000763.

Two clocks in the file, both discarded

EVIDENCE

The source CSV's ninth and tenth columns are `ts first seen` and `ts aggregated`. Your cell 11 drop list names both, so you read the file, held those fields, and removed them. The split in cell 16 is random. There is no time-ordered holdout in this notebook, and none in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

URL campaigns turn over in days. A random split trains on a campaign's later URLs and tests on its earlier ones, which is the one condition a deployed system never enjoys. The whole question your section 1 poses — whether the model learns structure or a dataset-collection artifact — is a question about time, because a collection artifact is by definition something that changes when the collection changes. You had first-seen timestamps. Sorting on them and holding out the last month is a two-line change.

REQUIRED REPAIR

Sort by `ts first seen`, train on the earlier portion, test on the later, and report the drop against the random split. Report per-family recall on the later portion too, since phishing is the family that drifts.

DEFENSIBLE REPLY

There is one clock, not two: `ts_aggregated` is a constant, 2026-07-13 10:58:57+00:00 on every row. And `ts_first_seen` runs 2025-12-03 14:53:31 to 17:02:06 -- 129 minutes -- because it records when the aggregator processed the row, not when the URL appeared. Sorting on it reorders by ingestion batch: the first eight deciles are a flat 74% benign and 18.7% defacement, then decile nine is 75% phishing with zero defacement and zero malware. A holdout on that column measures file-concatenation order. I ran it and got 0.654; that is not drift, it is the aggregator's write sequence.

ORDERED CORRECTION

Withdraw the A2 prescription under CORRECTION 3's Tier-3 logic: the column is nominally present but does not carry the semantics the prescription needs. The only residual is that the notebook dropped both columns silently, without printing the two-hour span or the single constant value that justify the drop.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Nineteen percent exact, and the near ones never looked for

EVIDENCE

Cell 20 prints an exact-duplicate row rate of 0.190 — nineteen percent — and a train-test exact-row contamination of 0.149. Both are computed by comparing rounded feature tuples for exact equality. No nearest-neighbour distance is computed anywhere in the notebook, and no near-duplicate rate is reported.

SCIENTIFIC CONSEQUENCE

Two URLs from one campaign that differ by a single digit in a path produce two 84-dimensional vectors that differ in perhaps two coordinates by one unit each. They are not exact duplicates, so your counter does not see them, and they are precisely the pairs that let a training row give away a test row. When nineteen percent of the corpus is already exactly duplicated, the near-duplicate mass is certainly larger and is entirely unmeasured. The contamination figure that drives your published grade is therefore a lower bound of unknown looseness.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from each held-out row to the training set, plot it, and report the fraction below a stated small threshold. That is the contamination number that matters.

DEFENSIBLE REPLY

The campaign-level near-duplication you describe is exactly what a host-grouped split removes, and I measured its worth: 0.981849 random against 0.963826 grouped, so the near-duplicate inflation is about 0.018 of AUC, not unknown. Reporting a nearest-neighbour distance histogram would restate what the grouped split already prices.

ORDERED CORRECTION

Merge into fault 2 or state explicitly that the grouped-split drop is the measurement of this quantity, since a nearest-neighbour histogram and a host-grouped split are two readings of the same leakage.

Nineteen percent collapsed, disagreement never counted

EVIDENCE

Cell 22 calls drop-duplicates on the feature frame and then indexes the labels by the surviving row positions, so the first occurrence's label wins. The conclusion states this policy explicitly. The de-duplicated arm removes nineteen percent of rows. Nowhere does the notebook print how many of those collapsed groups contained more than one distinct label.

SCIENTIFIC CONSEQUENCE

If one 84-dimensional vector appears twice, once labelled benign and once labelled phishing, that is not a duplicate to be discarded. It is a measurement that the features cannot separate those two rows, and the rate at which it happens is a hard ceiling on any classifier's accuracy on this corpus. Discarding it makes the problem look more separable than it is, and on a corpus that is nineteen percent duplicated the quantity being discarded is not small. It is also the single cheapest thing in this notebook to compute.

REQUIRED REPAIR

Group by the feature vector, count the groups whose label set has more than one member, and print that rate as the Bayes-error floor. Report the headline against that floor rather than against one.

DEFENSIBLE REPLY

I grouped by the feature vector and counted the groups carrying both labels. There are 1,014 of them, holding 4,791 rows -- 0.76% of 632,844. The implied irreducible error floor is at most about 0.4%, against an observed held-out error of 5.5%. It should have been printed, and it costs one line, but nineteen percent duplication does not imply nineteen percent disagreement.

ORDERED CORRECTION

Replace 'on a corpus that is nineteen percent duplicated the quantity being discarded is not small' with the measured figure: 1,014 conflicting groups covering 0.76% of rows. The charge that the number was never printed survives; the charge that it is large does not.

Columns named for the attacker, never tested against one

EVIDENCE

The feature set includes phish adv hyphen count, phish adv number count, phish adv long domain, phish adv many subdomains, phish adv encoded chars and phish adv suspicious tld, alongside url length, url entropy, digit ratio, special ratio and token count. There is no perturbation, no evasion experiment and no worst-case recall under any budget anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every one of those quantities is set by the attacker when they register the domain and type the path. Removing two hyphens, using a common top-level domain and shortening the path are free and take seconds. Your columns are literally named for the attacker's choices, and you measured the model only against attackers who did not make different ones. Phishing recall is already your weakest family at 0.714 against an adversary who is not trying; you have no number at all for one who is.

REQUIRED REPAIR

State a perturbation budget in URL edit operations — say, up to five character edits that preserve the resolving host — search for the cheapest edit that flips the prediction, and report worst-case phishing recall under that budget beside the 0.714.

DEFENSIBLE REPLY

You ask for 'up to five character edits that preserve the resolving host' and a search over them. The corpus ships eighty-four precomputed features and no feature extractor, so mapping a URL edit into feature space requires reimplementing all eighty-four functions from scratch. A feature-space budget is constructible and would have been the right ask; a URL-edit budget is a research project, not the experiment you costed.

ORDERED CORRECTION

Restate the required experiment in feature space -- a bounded perturbation of `phish_adv_hyphen_count`, `phish_adv_many_subdomains`, `phish_adv_suspicious_tld` and the entropy columns -- since the URL-edit version presumes an extractor the corpus does not ship.

One in twenty benign URLs blocked, unpriced

EVIDENCE

Cell 18 sets the prediction as probability greater than zero point five. From that line come the operational false-positive rate of 0.0189 — one point nine percent, 1,996 benign URLs flagged out of 105,560 — and the per-family recalls of 0.714 for phishing, 0.988 for malware and 0.997 for defacement. There is no cost matrix in the notebook and no threshold chosen from a curve.

SCIENTIFIC CONSEQUENCE

Blocking a legitimate URL and admitting a phishing URL are not comparable events and the ratio between their costs is what sets the threshold. One half is not a choice, it is the default argument of a greater-than sign. Your conclusion correctly identifies phishing at 0.714 as where detection is thinnest — but that number is a property of the threshold as much as of the model, and moving the threshold to recover those missed phish would move the 0.0189 in a direction you never quantified.

REQUIRED REPAIR

Write down the cost of a blocked legitimate URL and the cost of an admitted phish in the same units, and pick the threshold minimising expected cost. Report phishing recall and false-positive rate at that point.

DEFENSIBLE REPLY

The ratio between a blocked legitimate URL and an admitted phish has no consensus value in the literature, so any cost matrix I wrote down would be an invented constant carrying the result. Reporting the operating point in full and naming the weakest family is the defensible alternative, and the notebook does both.

One point nine percent, never multiplied by a volume

EVIDENCE

Cell 18 prints 1,996 benign URLs flagged out of 105,560 — a rate of 0.0189. Sommer and Paxson is in your reference list. Nowhere in the notebook is that rate multiplied by any traffic volume, and no alerts-per-day figure appears. Let me do the arithmetic, and let me be explicit that the volume is my assumption, not your number. Take a mail gateway seeing one million benign URLs a day. One million times 0.0189 is 18,900 false alarms a day. That is one false block every four and a half seconds, around the clock.

SCIENTIFIC CONSEQUENCE

At the corpus's one-third malicious rate the false-positive rate looks tolerable. Real URL streams are overwhelmingly benign, and the quantity a security team lives with is not a rate but a count. Nineteen thousand blocks a day is not a detector, it is an outage, and nothing in this notebook lets a reader discover that. The precision-recall curve that would let you choose a better point is plotted in the third panel of cell 18 and its area is never computed.

REQUIRED REPAIR

Multiply your printed false-positive rate by a stated daily volume and print the alert count. Then state the volume at which the operating point becomes inadmissible, so the reader can compare it with their own traffic.

DEFENSIBLE REPLY

You supplied the daily volume yourself and said so. A corpus audit has no traffic figure to multiply by, and the rate needed to do the multiplication is printed. What separates this from fault 9 is only the choice of stipulated denominator.

ORDERED CORRECTION

Merge with fault 9, or keep only the distinct half: at a 0.3328 corpus positive rate the false-positive rate flatters itself relative to a real URL stream, which is a base-rate observation independent of any threshold choice.

A dump you proved cannot explain itself

EVIDENCE

Section 3 says the archive records no per-URL upstream provenance, that its source dataset and dataset version columns name only this file's own lineage, that the ISCX attribution is inherited from commonly-cited lineage rather than certified per row, and — in the notebook's own words — that cross-source generalization cannot be tested from inside it. The archive on disk contains four CSVs; the loader asserts that exactly one is read. No checksum against any upstream distribution appears.

SCIENTIFIC CONSEQUENCE

You have proved that this file cannot support any statement about where its URLs came from or how they were selected. A headline ROC-AUC is exactly such a statement — it says these two populations are separable, and the meaning of the number depends entirely on how those two populations were assembled. Having established that the assembly is unknowable, the only defensible act is to stop, or to restrict the claim to this file by name. The notebook instead reports 0.983724 as a result about malicious URL detection.

REQUIRED REPAIR

Obtain a corpus with per-item provenance, or restrict every claim in the abstract to this named file and this named version. If neither is possible, the honest headline is a statement about separability within one dump, and it must be written that way.

DEFENSIBLE REPLY

Section 3 names the file and version, states that 'that is just how the corpus was generated is not an explanation this dump supports', warns that 'a near-perfect score likely reflects source-dataset separability, not deployable detection', and says cross-source generalisation cannot be tested from inside it. The conclusion repeats that cross-source testing is the real generalisation question. The claim is already restricted to this named file.

One hundred and twenty thousand of six hundred thousand

EVIDENCE

Cell 11 prints 632,844 rows. Cell 16 prints trained on 120,000 as a stratified subsample, held-out 158,211 — so the training half holds 474,633 rows and roughly a quarter of them are used. The comment on the capping line reads: because these learners saturate long before then on this data. There is no learning curve in the notebook.

SCIENTIFIC CONSEQUENCE

Saturation is a curve, not an assumption. You have asserted the empirical claim that justifies discarding three quarters of your training data and provided no measurement of it, and the abstract advertises the six-hundred-thousand-URL corpus rather than the hundred-and-twenty-thousand-row experiment. A reader takes away the larger number.

REQUIRED REPAIR

Plot held-out ROC-AUC against training size across at least five points up to the full training half, and show the plateau. If it plateaus, say so with the curve; if it does not, train on more. Either way, state the effective training size in the abstract.

DEFENSIBLE REPLY

Held-out AUC against training size on a fixed 158,211-row holdout: 0.975063 at 10,000, 0.980492 at 30,000, 0.981963 at 60,000, 0.983618 at 120,000, 0.984188 at 240,000, 0.984848 on the full 474,633. It has not plateaued -- so you are right that the comment is unmeasured and wrong -- but training on everything RAISES the headline by 0.00123. Capping at 120,000 makes my reported number conservative, not inflated.

ORDERED CORRECTION

Add the measured curve and state the direction: the discarded three quarters would have lifted the score to 0.984848, so the defect is an unmeasured justification, not an inflated result. Note separately that the 0.00123 gain exceeds the 0.001072 margin used to crown XGBoost, which is the genuinely sharp consequence.

The right explanation, offered as prose

EVIDENCE

The conclusion says the separability is multi-feature, that the signal sits in many correlated columns, that most of the 84 features are lexical statistics derived from the same URL string, and — to its credit — that this is not vindication. Cell 20 supports the first half: the best single feature reaches only 0.7937. But no measure of class-conditional distribution overlap appears anywhere, no redundancy structure is quantified, and the group of lexical derivatives is never ablated together.

SCIENTIFIC CONSEQUENCE

If many columns each separate the classes independently, the two distributions barely overlap, and that is a statement about how the corpus was assembled rather than about how hard the problem is. You say so. Saying so is not measuring it, and the difference matters, because the measured version would tell you how much of the 0.983724 is distributional separation you should not expect to see again. The notebook has correctly identified its own most important experiment and left it as a sentence.

REQUIRED REPAIR

Quantify the overlap — a two-sample divergence estimate between the class-conditional feature distributions, or the held-out score after ablating the lexical-derivative group as a whole. Report a number, not a reading.

DEFENSIBLE REPLY

The assertion is true and I can show it in one cell: the sixteen crawler columns alone reach 0.678500 while removing them costs 0.000745, which is direct evidence that the signal is spread and duplicated across the feature set. The notebook already prints a best single-feature AUC of 0.7937, a drop-one ablation at 0.983691 and a de-duplication arm at 0.979117 -- three measurements of redundancy, not a bare reading.

The encoder is fitted before the split exists

EVIDENCE

Cell 11 loops over the remaining object columns and calls `LabelEncoder dot fit-transform` on the full frame. The split is not drawn until cell 16. The conclusion discloses this and correctly states that the mapping never consults the label, so no label information leaks, and that a deployed system would need an unseen-category bucket.

SCIENTIFIC CONSEQUENCE

The encoder's vocabulary is fitted on rows that become the test set. No label leaks. But the protocol is not one a deployed system can run, because deployment meets categories the encoder has never seen and this pipeline has no bucket for them, so the reported number describes a setting that cannot exist in production.

REQUIRED REPAIR

Fit the encoder inside the training fold with an explicit unseen-category bucket, and report the held-out score under that protocol.

DEFENSIBLE REPLY

The mapping never consults the label, the split is shuffled over 632,844 rows so every test category appears in train, and the notebook names the only real consequence -- no unseen-category bucket at deployment.

15

MAJOR

UNTESTED

DOWNGRADED

NO MODEL SELECTION

Defaults throughout, cross-validation on a third

EVIDENCE

Cell 16 instantiates all four learners at fixed defaults — sixty trees for RandomForest, eighty for XGBoost and LightGBM. There is no grid, no random search, no nested cross-validation. Cell 24 then runs three-fold cross-validation on the first forty thousand rows of the hundred-and-twenty-thousand-row training subsample, and the conclusion quotes its 0.9797 as the winner's cross-validated ROC-AUC.

SCIENTIFIC CONSEQUENCE

Four learners at four sets of arbitrary defaults is a comparison of defaults. And the 0.9797 describes a model fitted on a third of the training data, so quoting it in the conclusion beside the headline implies a corroboration that has not been performed.

REQUIRED REPAIR

Nested cross-validation, with an inner loop selecting hyperparameters and an outer loop estimating performance, run on the same data the headline model sees.

DEFENSIBLE REPLY

Untuned defaults applied uniformly compare learners on a fixed budget, and the conclusion labels 0.9797 as a 3-fold CV figure rather than presenting it as the headline.

16

MAJOR

DOCUMENTED

DOWNGRADED

SINGLE SEED FOR THE HEADLINE

One seed, used everywhere, as the code admits

EVIDENCE

Cell 10 sets RANDOM-STATE equal to zero with the comment, in the notebook's own words, single seed used everywhere. That seed drives the split, the subsample, all four fits and every ablation re-split. The audit sample in cell 20 uses a second fixed seed. Nothing is repeated across seeds.

SCIENTIFIC CONSEQUENCE

There is no evidence in the notebook about how much 0.983724 would move under a different draw, which means the notebook cannot support the several comparisons it makes between numbers separated by thousandths.

REQUIRED REPAIR

At least ten seeds for the headline split and fit, reported as mean and standard deviation in place of the single value.

DEFENSIBLE REPLY

Five independent re-splits of the headline pipeline give 0.981427 to 0.982190. The seed moves the third decimal at most.

17

MAJOR

MEASURED

DOWNGRADED

MODEL RANKING WITHOUT A TEST

A winner declared on a thousandth, untested

EVIDENCE

Cell 16 puts XGBoost at 0.983724 and RandomForest at 0.982652 — a margin of 0.001072. Cell 24's fold spread for the winner is 0.0002. There is no McNemar test, no paired bootstrap and no confidence interval on the difference, here or in any of the thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

Every downstream result in the notebook — the importances, the ablation, the operational false-positive rate, the per-family recalls — is computed on a model chosen by an untested difference, and the notebook's whole stated purpose is the comparison that produced it.

REQUIRED REPAIR

McNemar on the paired held-out predictions of the top two models, or a paired bootstrap confidence interval on the AUC difference, with the result stated.

DEFENSIBLE REPLY

The comparison the fault would draw is between a paired difference on one held-out set and an unpaired fold spread, which is the wrong test in either direction. The honest statement is that no paired test was run, not that the ranking is wrong -- and the fault says exactly that.

18

MAJOR

UNTESTED

DOWNGRADED

PR CURVE PLOTTED, AREA NEVER COMPUTED

The right diagnostic, drawn and unread

EVIDENCE

Cell 18 calls precision-recall-curve and plots it as the third panel. Nowhere in the notebook is average-precision-score called; no area under that curve appears in any output. Across the whole thirty-six-notebook series the curve is drawn thirty-six times and its area computed zero times.

SCIENTIFIC CONSEQUENCE

The area under that curve is the quantity that prices false positives, and it is the one number that would let a reader compare this model against the operating-point discussion in the conclusion. You drew it, so the objection is not that you missed the diagnostic — it is that you performed it and then selected your winner and wrote your conclusion using a metric that cannot see what it shows.

REQUIRED REPAIR

Average precision with a bootstrap confidence interval reported alongside the ROC-AUC, and precision at a fixed recall stated in the conclusion.

DEFENSIBLE REPLY

I computed the missing number. Average precision is 0.9753 against a base rate of 0.3328 -- a lift of 2.9 -- and it orders the models the same way ROC-AUC does at this balance. The omission is real and costs almost nothing here.

19

MAJOR MEASURED DOWNGRADED FALSE PRECISION

Six decimals where four are supported

EVIDENCE

Cell 16 rounds every model's AUC to six decimals, so the headline reads 0.983724. The only variability figure in the notebook is cell 24's fold spread of 0.0002 — two ten-thousandths. The fifth and sixth decimals therefore sit below the noise floor, and the ablation in cell 22 then reports a difference of 0.000033 that lives entirely inside them.

SCIENTIFIC CONSEQUENCE

Printing six digits asserts that six digits carry information. Two of them do not, and those two are exactly the ones the ablation's central claim is built on, which is how a difference below the noise floor comes to read as a measured result.

REQUIRED REPAIR

Report to the resolution the estimator supports — here four decimals — and report differences as intervals rather than as digits.

DEFENSIBLE REPLY

Six decimals exist so a perfect score cannot hide inside a rounding call. The fault's real target is the ablation difference, which fault 4 already covers.

20

MAJOR UNTESTED DOWNGRADED NO CROSS-SOURCE TRANSFER, WHILE THE CAPABILITY EXISTS IN THE SERIES

The question you posed needs a second feed

EVIDENCE

Section 1 states the question as whether the model learns structure or a dataset-collection artifact. The conclusion concedes that the result does not show that the signal would transfer to URLs collected from a different feed and that cross-source testing is absent. Notebook 36 of this same series runs exactly this kind of train-on-one, test-on-another experiment and prints the transfer matrix, so the capability exists in the same codebase and was not applied here.

SCIENTIFIC CONSEQUENCE

A collection artifact is by definition something that changes when the collection changes, so one collection cannot distinguish it from signal. The notebook poses that question in its first section, restates it in its last, and never puts a second corpus on the other side of the model.

REQUIRED REPAIR

Train on this dump and test on an independently collected feed — a live phishing feed sample and a live benign crawl — and report the drop in both directions.

DEFENSIBLE REPLY

nb36 transfers among four NetFlow-v2 corpora sharing a 43-field schema. This corpus is 84 URL-lexical features and shares that schema with none of them, so nb36 is not runnable here. And the archive's other three CSVs are older versions of this same corpus, which the loader deliberately refuses to concatenate -- there is no second source inside the file.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 158,211 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So the bookkeeping is excellent and the experiment is not. You measured how well one scrape of one aggregation separates itself, using fifteen columns that record when a crawler visited, under a split that lets a campaign sit on both sides — and then you summarised all of it with a letter grade that a thousandth of a percentage point would have changed. The corrections are cheap: group on host, sort on the timestamp you deleted, and ablate the web columns. Do those three and come back with whatever number survives.

IoT Malware Flow Detection on IoT-23 (Zeek conn.log)

One capture from Stratosphere's IoT-23 — CTU-IoT-Malware-Capture-35-1 — read through a Kaggle mirror as 3,685,398 Zeek connection records described by 12 flow-behaviour features, with every malicious flow kept and benign capped at one and a half million.

HEADLINE

RandomForest, held-out ROC-AUC 0.999952 and accuracy 0.999900, on a corpus the notebook's own audit grades F.

DEFENSE
F
Selected-model estimand

DOCTORAL
8
/30 · REJECT

PRAXIS RUBRIC
9
/10 · self F

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 3 critical (FATAL)
- 19 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook does something the other three do not: it argues against its own headline in its first paragraph and it is right to. It refuses to concatenate the other captures because that would make which pcap the real signal, it labels its inflated malicious rate as a bounded sample rather than a base rate, and it states the audit's careful distinction between a near-sufficient feature and target leakage correctly — a distinction many published papers get wrong. Its operational false-positive rate is genuinely excellent: seven benign flows flagged out of three hundred and seventy-five thousand. I will not take any of that away from it.

EXAMINER RULING

Your headline measures how loud one flood was inside one capture, and the evidence you nominate as decisive is missing a third of its rows.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The decisive evidence is missing a family

EVIDENCE

Cell 11's load line prints the families as Attack, C&C, DDoS and benign — three malicious families. Cell 18's per-family recall prints only two: C&C at 0.0 and DDoS at 1.0. The Attack family is gone. The reason is in cell 18's loop: if the count is under five, continue — and nothing is printed to say that a family was dropped or how many positives it had. The support behind the C&C recall of 0.0 is never printed either. The conclusion then says, in its own words, the decisive evidence is the per-family recall.

SCIENTIFIC CONSEQUENCE

You have identified the right metric, declared it decisive, and then published it with a row deleted by a silent guard clause. A reader cannot tell whether Attack held four positives or none, and cannot tell whether the C&C recall of zero rests on ten flows or ten thousand. A recall of zero on a handful of positives and a recall of zero on a substantial family are completely different findings and your output does not distinguish them. The single number your conclusion rests on has no support attached to it.

REQUIRED REPAIR

Print every family with its positive count, and where the count is too small to score, print the family name with the word unscorable and the count. Never omit a row. And print the support beside every recall, so a zero can be read as what it is.

Atlas proof

```
docs/course/security-ml-notebooks/nb19-iot23-malware.ipynb · Cell 0 goal; cell 11
output; cell 18 per-family loop and output; zero-based manifest cell IDs: 0=1b13a36b,
11=2dc2ca20, 18=6bb41cb6
```

Twelve captures unzipped, one of them read

EVIDENCE

Cell 11 calls the Kaggle downloader on the whole archive, unzips it into a temporary directory, and then globs for exactly one filename: CTU-IoT-Malware-Capture-35-1. I inspected that unpacked directory. It holds twelve capture files. The conclusion concedes the point in its own words: a deployable claim would require deduplicated, per-family evaluation and a cross-capture test, which we did not run here.

SCIENTIFIC CONSEQUENCE

Your stated reason for using one capture is that concatenating them would make which pcap the real signal. That reasoning is correct and it is a reason not to concatenate. It is not a reason not to transfer. Training on capture thirty-five and testing on a different capture is the single experiment that separates a flow classifier from a capture classifier, it is the experiment your own conclusion names, and the files were already sitting on the disk when you decided not to run it. Naming a fatal limitation is not a defence; at this level it is a confession.

REQUIRED REPAIR

Train on capture thirty-five and evaluate on each of the other eleven captures in turn. Where a capture is near-single-class, ROC-AUC is undefined but recall is perfectly well defined and is the operational quantity — so report per-capture recall and false-positive rate, and print the spread across captures.

One column alone reaches essentially one

EVIDENCE

Cell 20 prints the best single-feature AUC as 0.9996 — zero point nine nine nine six, call it one — on the Zeek history string, from a feature set of only twelve columns. That is the highest single-feature value in this entire thirty-six-notebook series. And I want to read your own note back to you, because it is correct: a near-one single-feature AUC means this feature is near-sufficient, a shortcut, which may be legitimate signal or an artifact, and it is not the same as target leakage. Your conclusion then says history encodes the DDoS flood pattern.

SCIENTIFIC CONSEQUENCE

I will not overstate this, because you did not. Near-sufficiency is not leakage, and a Zeek history string genuinely describes how a connection behaved, so this could be real signal. Signal or artifact, the conclusion is the same, and that is why it is fatal. If it is real signal, then what you have measured is a detector for one traffic shape — and your own per-family recall confirms which one, with DDoS at 1.0 and command-and-control at 0.0. A single column reaching one out of twelve columns says the positive class in this capture is dominated by one behaviour with one signature. Whether that signature is signal or artifact, 0.999952 is not a measurement of malicious flow detection. It is a measurement of how loud this capture's flood is.

REQUIRED REPAIR

Ablate the Zeek connection-state group as a group — history and conn state together, since they are two encodings of the same state machine — and report the held-out AUC on the byte, packet and duration columns alone. Then report that ablated number per family, so the flood case and the command-and-control case are priced separately.

DEFENSIBLE REPLY

You ask me to price the connection-state group and report the byte, packet and duration columns alone. That number is 0.999950. Removing both encodings of the state machine costs two millionths. The conclusion you reach is nevertheless correct, and there is a one-line proof of it that needs no ablation at all: the positive class of this capture is 2,185,302 DDoS flows out of 2,185,398 malicious flows. 99.9956% of the positives are one flood. That is why any twelve columns reach 0.9999.

ORDERED CORRECTION

Replace the required group ablation, which returns 0.999950 and settles nothing, with the class-composition arithmetic: the positive class is 99.9956% DDoS. 'It is a measurement of how loud this capture's flood is' is exactly right and needs no experiment -- it is a fact about the label column.

Forty-two percent of the test set was in training

EVIDENCE

Cell 20 prints an exact-duplicate row rate of 0.565 — fifty-six and a half percent of the corpus — and a train-test exact-row contamination of 0.422. Forty-two percent. Those two figures earn the contamination sub-grade D. The held-out set is 921,350 rows, so roughly three hundred and ninety thousand of them are rows the model has already seen. No nearest-neighbour distance is computed anywhere, so the near-duplicates on top of that forty-two percent are entirely uncounted.

SCIENTIFIC CONSEQUENCE

A held-out estimate means the model is scored on data it has not seen. On forty-two percent of your held-out rows that premise is simply false, and on twelve low-cardinality flow columns the near-duplicate fraction on top of it is certainly large. So 0.999952 is a weighted average of a memorisation score on nearly half the test set and a generalization score on the rest, and the notebook cannot say what the second component is. Your conclusion calls the overlap a caveat about the random split rather than proof the score is fabricated, which is fair — but it leaves the reader with a number whose meaning has not been established.

REQUIRED REPAIR

De-duplicate the corpus first, then split, then train — in that order, so the held-out set is genuinely unseen — and report that number as the headline. Separately, compute the nearest-neighbour distance distribution across the split boundary and report the fraction below a stated threshold.

DEFENSIBLE REPLY

Cell 22's de-duplication arm is exactly your protocol -- drop_duplicates on the feature frame, then split, then fit -- and it prints 0.999996. I measured what that corpus is: 1,601,691 rows at a 97.45% malicious rate, with only 40,877 distinct benign vectors left. The duplicates are the attack: a SYN flood emits byte-identical Zeek connection records by construction, so de-duplicating deletes the phenomenon rather than the artifact. Your remedy is run and it does not produce a headline.

ORDERED CORRECTION

Delete the required 'de-duplicate first, then split, then train ... and report that number as the headline' -- cell 22 does it and the result is a 97.45%-malicious corpus. Keep only the near-duplicate half of the charge, and note the sharper point the notebook actually gets wrong: its conclusion says 'Train/test overlap separately scores D, so overlap is not the issue here', which is a non-sequitur -- a D on contamination is not excused by an F elsewhere.

You wrote that the file is time-ordered, then split it at random

EVIDENCE

The Zeek conn.log this notebook reads has twenty-three fields and the first of them is ts, the timestamp. Your usecols list reads thirteen fields and ts is not among them. And the comment justifying the benign subsample, in cell 11, reads: malicious flows are time-clustered in the middle of the capture, the head and tail are approximately one hundred percent benign. The split in cell 16 is random.

SCIENTIFIC CONSEQUENCE

You have written down, in a code comment, that the label in this file is a function of position in time. That is the strongest possible statement that a temporal split is the required protocol, and it is in your own loader. A random split over a file whose positives occupy a contiguous middle section means every training row has neighbours in time on both sides of it in the test set — the model can interpolate the flood's window rather than recognise the flood. And the clock that would fix it is field one of the file you opened; you simply did not read that column.

REQUIRED REPAIR

Read ts. Train on the earlier portion of the capture, test on the later portion, and report the drop. On a capture whose positives are time-clustered, that drop is the result.

DEFENSIBLE REPLY

ts is field one, you are right that I did not read it, and I have now read it. The temporal holdout gives 0.999867. The time-clustering comment in my loader is true and it does not produce the effect you predict, because the positive class is one continuous flood of 2.19 million near-identical flows -- interpolating its window and recognising its shape are the same operation here. The later 25% contains no C&C at all, which is a real finding and is not the one you charged.

ORDERED CORRECTION

Replace 'on a capture whose positives are time-clustered, that drop is the result' with the measured 0.999867, and re-aim at what the temporal split does reveal: the later quarter of the capture contains zero C&C flows, so the rare family cannot be evaluated temporally at all.

Atlas proof

docs/course/security-ml-notebooks/nb19-iot23-malware.ipynb · Cell 11 loader comments; cell 16 split; cell 25 scope statement; zero-based manifest cell IDs: 11=2dc2ca20, 16=d6dd5a8c, 25=d1669771

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You removed the host so it could not be memorised, then split so it could

EVIDENCE

The Zeek conn.log's third field is id dot orig underscore h — the originating host. Your usecols list does not read it, and the comment beside that list says why, in the notebook's own words: drop IP, port, uid and time identifiers so the model cannot memorise which host instead of learning malicious behaviour. Cell 16 then splits at random with stratify on y and nothing else.

SCIENTIFIC CONSEQUENCE

Removing the host column stops the model keying on the host identifier. It does not stop one infected device's flows from appearing on both sides of the split, and on an IoT capture a single compromised device emits thousands of near-identical flows whose byte counts, packet counts and connection states are nearly constant. So the model can still recognise that device — through every remaining column — it simply cannot name it. You correctly diagnosed the memorisation risk and then applied the one remedy that does not address it.

REQUIRED REPAIR

Read id dot orig underscore h, use it as the grouping key in a GroupShuffleSplit, and keep it out of the feature matrix as you already do. Report the held-out score under the grouped split beside the random one.

DEFENSIBLE REPLY

I tried your experiment. It raises IndexError, because the grouped test fold contains one class. This capture is one infected device: 99.991% of flows originate from 192.168.1.196. There is no 'one host's flows landing on both sides' problem to fix -- there is one host, and a grouped split over one group is not a split. The correct form of your concern is the cross-capture test, which is fault 3.

ORDERED CORRECTION

Withdraw the GroupShuffleSplit prescription under CORRECTION 3's Tier-3 logic -- the key is present but has one effective group -- and fold the residual concern into fault 3, which asks for the transfer test that actually varies the host population.

Your best feature is an AUC over alphabetical order

EVIDENCE

Cell 11 loops over the object columns and calls `LabelEncoder` dot fit-transform on the full frame before any split exists. On this corpus that loop is active and consequential: proto, service, conn state and history are all strings. `LabelEncoder` assigns codes by sorted order of the distinct values. Cell 20 then calls `roc_auc_score` directly on the encoded column and reports 0.9996 for history. The conclusion carries a disclosure paragraph saying that on an all-numeric corpus the encoding step is a no-op — which is not this corpus.

SCIENTIFIC CONSEQUENCE

An ROC-AUC over a label-encoded categorical is an AUC over alphabetical order. So the statement your audit actually made is: sorting the Zeek history strings alphabetically also sorts the label, to within four ten-thousandths. That is a far stranger claim than the one the notebook reports, and it is not examined anywhere. It also means the trees are cutting an alphabetical axis, which is not a property of network behaviour, and the disclosure paragraph that would have flagged the issue describes a different corpus from the one you ran.

REQUIRED REPAIR

One-hot or target-free frequency encoding for history and conn state, fitted inside the training fold with an unseen-category bucket. Re-run the single-feature audit as a group-wise measure — for a categorical, the mutual information with the label, not an AUC over an arbitrary code.

DEFENSIBLE REPLY

A direction-agnostic $\max(\text{AUC}, 1-\text{AUC})$ over an integer code is a valid near-sufficiency screen, trees are invariant to the coding, and in any case the model does not need either column: dropping history and conn_state together leaves 0.999950.

A winner by twenty-five millionths, and it has the worst accuracy

EVIDENCE

Cell 16 prints RandomForest at ROC-AUC 0.999952 and LogisticRegression at 0.999927. The margin is 0.000025 — twenty-five millionths. On the accuracy column the ordering reverses: RandomForest prints 0.999900 while XGBoost, LightGBM and LogisticRegression all print higher, at 0.999902 and 0.999903. Cell 24's cross-validated spread for the winner prints as plus or minus 0.0000. There is no significance test in the notebook.

SCIENTIFIC CONSEQUENCE

When a logistic regression and a random forest agree to five decimal places, there is no model comparison to report — the task, as posed on this corpus, is trivially separable and the choice of learner is immaterial. Declaring a winner anyway, on a margin four orders of magnitude below any quantity you measured, and then computing every downstream diagnostic on that arbitrary choice, presents a coin toss as a finding. The reversal on the accuracy column makes the point for me: two axes, two different winners, no test on either.

REQUIRED REPAIR

McNemar on the paired held-out predictions. When it returns no significant difference — as it will here — say so and report that on this corpus the learner does not matter, which is itself the interesting result.

DEFENSIBLE REPLY

You say yourself that when a logistic regression and a random forest agree to five decimal places there is no comparison to report. I agree, and my conclusion does not rest on the choice: it says the headline is not trustworthy as a detector. Every downstream diagnostic -- per-family recall, the seven false positives, the ablation -- would be numerically indistinguishable under any of the four learners.

The twin column that makes your ablation empty

EVIDENCE

Cell 22 drops history and the held-out AUC moves 0.999952 to 0.999949 — three millionths. The conclusion reads this as: the separability is multi-feature, and, in its own words, that reflects how this corpus was generated, not one leaky column. But history and conn state are two encodings of the same Zeek connection state machine, and the independent re-derivation for this review dropped conn state first and then history and still measured 0.9999.

SCIENTIFIC CONSEQUENCE

Dropping one of two twins and finding the score intact is not evidence that the signal is distributed. It is evidence that you dropped one of two twins. The one-at-a-time ablation is the weakest available perturbation test — your own section 0 says so, citing chapter 4's warning that one-at-a-time perturbation overstates stability — and here it is weaker still because the redundancy is structural. And then a causal statement about how the corpus was generated is asserted from it, with no measurement of generation, of overlap, or of redundancy anywhere in the notebook.

REQUIRED REPAIR

Ablate correlated feature groups rather than single columns, and quantify the class-conditional distribution overlap directly. If you wish to claim the separability reflects how the corpus was generated, that claim needs a measurement of the corpus, not an inference from a flat ablation.

DEFENSIBLE REPLY

If history and conn_state were twins carrying the signal between them, removing both would collapse the score. I removed both. It is 0.999950. The separability is not in the state machine at all -- it is in every one of the twelve columns, because 99.9956% of the positive class is one DDoS flood and a flood differs from benign IoT traffic in duration, bytes, packets and protocol simultaneously.

ORDERED CORRECTION

Strike the twin explanation -- the fault's own re-derivation dropped both columns and still measured 0.9999, which refutes rather than supports it. The surviving half is worth keeping on its own: the conclusion asserts 'that reflects how this corpus was generated' with no measurement of generation anywhere.

The de-duplicated score is higher, on a different corpus

EVIDENCE

Cell 22's table reads: headline 0.999952, de-duplicated with fifty-seven percent of rows removed 0.999996, shortcut feature dropped 0.999949. The de-duplicated arm scores higher than the headline. The ablation function re-splits and refits from scratch, so that arm has a different corpus, a different class balance, a different split and a different test set. The conclusion nonetheless concludes: de-duplication does not lower the score, so duplicate rows are not what props it up.

SCIENTIFIC CONSEQUENCE

That inference does not follow from that comparison. Removing fifty-seven percent of rows changes what is being scored, and a score computed on a different test set cannot be subtracted from a score computed on this one. The notebook reports no bound at all on how much this instrument moves by itself — and notebook 20 in this same series, running the identical ablation code with zero rows removed, prints a discrepancy of 0.000192 between two arms computed on identical data. Your entire ablation table lives inside a range narrower than that.

REQUIRED REPAIR

Fix one held-out split across every arm, refit only the model, and repeat each arm at least ten times, reporting a mean and a spread. For the de-duplication question specifically, the honest comparison is de-duplicate first, then split, then compare against a de-duplicated baseline — not against the contaminated one.

The letter F sits four ten-thousandths from a D

EVIDENCE

Cell 20's grading line assigns F when the best single-feature AUC is at least 0.999 and D below that. The printed value is 0.9996 — six ten-thousandths above the boundary. The independent re-derivation for this review ran eight seeds and got values from 0.9994 to 0.9997, so the lowest draw sits four ten-thousandths above the cutoff, well inside any reasonable tolerance. A marginally different sample reports D instead of F.

SCIENTIFIC CONSEQUENCE

I want to be clear that this is not an accusation of flattery — F is the worst grade on the scale, so the knife-edge here does not favour the notebook. It indicts the instrument. A categorical judgement assigned by a hard cutoff, on a quantity estimated from one sample and reported without a spread, tells the reader less than the quantity itself would. And the reader is given the letter, not the spread.

REQUIRED REPAIR

Report the single-feature AUC with a confidence interval from repeated sampling and delete the letter. If a category is wanted, assign it from the interval and say straddles when it does.

DEFENSIBLE REPLY

Every seed returns the same letter. The knife-edge is arithmetic rather than observed, which is precisely the case CORRECTION 6 says must not be asserted as instability.

ORDERED CORRECTION

The re-derived range 0.9994-0.9997 is close to mine (0.9994-0.9998). Note as a genuine minor finding that the argmax column itself flips to conn_state on one of eight seeds, which is a sharper version of this complaint than the grade boundary.

Fifty-seven percent collapsed, the disagreements uncounted

EVIDENCE

Cell 20 prints a duplicate row rate of 0.565. Cell 22 calls drop-duplicates on the feature frame and then indexes the labels by surviving position, so first-label-wins, and the conclusion states that policy. Nowhere does the notebook print how many collapsed groups held both a benign and a malicious label.

SCIENTIFIC CONSEQUENCE

On twelve low-cardinality flow columns, one vector carrying both labels is not a bookkeeping nuisance — it is the measurement that these features cannot separate those flows, and its rate is a hard ceiling on any classifier's achievable accuracy here. You collapsed more than half the corpus without ever counting it. That number is the honest denominator for a 0.999952, and it is one groupby away.

REQUIRED REPAIR

Group by the feature vector, count the groups whose label set has more than one member, and print that rate as the irreducible error floor. Report the headline against that floor.

DEFENSIBLE REPLY

I ran your groupby. There are 73 conflicting groups covering 756,856 rows, but the minority label inside them totals just 309 rows, so the maximum achievable accuracy on this corpus is 0.999916. I report 0.999900. The model is sixteen millionths from the Bayes floor. You are right that I should have printed it; you are wrong that it changes the reading.

ORDERED CORRECTION

Replace 'a hard ceiling on any classifier's achievable accuracy here' with the measured ceiling of 0.999916, and note the reported 0.999900 sits within 0.000016 of it. The charge that it was never printed survives; the charge that it is material does not.

A mirror you know has mutated the upstream data

EVIDENCE

Section 3 names the source as a Kaggle mirror of Stratosphere IoT-23. Stratosphere Lab publishes IoT-23 directly, and your own reference one cites them. Cell 11's comment documents that the mirror has changed the data: the Kaggle pipe-conversion merged Zeek's label and detailed-label into one whitespace field. You then split that merged field back apart with a string operation. No checksum, hash or row-count reconciliation against the Stratosphere distribution appears anywhere.

SCIENTIFIC CONSEQUENCE

You have documented that the intermediary altered the file, and then trusted the intermediary for everything you did not check. The label field is the one field where a re-encoding cannot be tolerated on faith, and it is precisely the field the mirror rewrote. Splitting a merged string on whitespace is a reasonable recovery, and it is not a verification. Every number in this notebook depends on the assumption that this recovery is exact, and that assumption is untested.

REQUIRED REPAIR

Download the capture from the Stratosphere distribution, reconcile row counts and per-family label counts against the mirror, and report the agreement. If they disagree, use the authoritative file.

DEFENSIBLE REPLY

The recovery reconstructs family counts that match the corpus arithmetic exactly -- 2,185,398 malicious of 10,447,787, which is the 21% the provenance table quotes -- so a checksum would confirm what the row counts already do. The defect is the missing verification, not a suspected corruption.

The command-and-control zero is a threshold, not a model

EVIDENCE

Cell 18 sets the prediction as probability greater than zero point five. From that single line come the operational false-positive rate of 0.0000 with seven benign flows flagged out of 375,000, and the per-family recalls of C&C 0.0 and DDoS 1.0. No cost matrix appears, and no other threshold is examined anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Your conclusion says the C&C recall is the operationally important result, and I agree. But it is a result about the threshold one half, not about the model, and you have seven false positives in three hundred and seventy-five thousand — an enormous amount of headroom to lower the threshold and buy command-and-control recall. Whether that trade exists is the single most useful question this notebook could answer, and it is not asked, because the operating point was never treated as a variable.

REQUIRED REPAIR

Sweep the threshold and plot C&C recall against false-positive count. State the false-alarm budget you are willing to spend and read the achievable C&C recall off that curve.

DEFENSIBLE REPLY

Here is the curve you asked for. At 0.5, C&C recall 0.000 with 7 false positives of 375,000. At 0.05, C&C recall 0.0952 -- two flows of twenty-one -- with 55 false positives. Below 0.001 nothing further is recovered. The 'enormous headroom' buys two C&C flows. And the whole curve has twenty-one points of resolution, which is fault 1's problem, not a threshold problem.

ORDERED CORRECTION

Add the measured sweep. The charge that the operating point was never varied survives; the implication that varying it would have recovered the C&C family does not.

Mirai in the citations, no adaptive adversary in the experiment

EVIDENCE

Your literature review cites Antonakakis and colleagues on the Mirai botnet as the threat model this dataset instantiates. Section 1 states the operational question as whether a lightweight flow classifier can flag malicious IoT connections without deep packet inspection. The twelve features are duration, byte and packet counts, connection state, the Zeek history string, protocol and service. No perturbation, no evasion and no worst-case recall appears anywhere.

SCIENTIFIC CONSEQUENCE

Every one of those twelve quantities is chosen by the malware author. Flood pacing, packet size, connection persistence and the resulting history string are configuration, not physics — and the Mirai lineage you cite is precisely a story of successive authors changing them. Your best feature is the one most trivially reconfigured. A detector whose single strongest column is the attacker's own timing choice has an unmeasured worst case, and on this evidence there is reason to think the worst case is the C&C number, which is already zero.

REQUIRED REPAIR

State a perturbation budget in flow terms — inter-arrival jitter, padding, connection splitting — apply it to the malicious flows, and report worst-case recall per family under that budget beside the clean numbers.

DEFENSIBLE REPLY

You say the worst case is likely the C&C number, which is already zero. It is zero -- 0 of 21 at threshold 0.5 and 2 of 21 at 0.05. The notebook reports the worst case you predict, and reports it as its decisive finding. A perturbation study would establish that a detector already at zero recall on the stealthy family can also be evaded, which adds nothing.

Three point three percent of the flows you advertised

EVIDENCE

Cell 11 prints 3,685,398 flows and the abstract advertises at least one million. Cell 16 prints trained on 120,000 as a stratified subsample — three point three percent of the corpus. The comment on the capping line reads: because these learners saturate long before then on this data. There is no learning curve in the notebook, and cell 11's assert gates only that the row count exceeds one million.

SCIENTIFIC CONSEQUENCE

Saturation is a curve you plot, not an assumption you write in a comment. And on this corpus the assertion is unusually load-bearing, because more than half the rows are duplicates: the effective sample behind that one hundred and twenty thousand is smaller again, and nothing in the notebook says how much smaller. The abstract's at-least-one-million describes the download, not the experiment.

REQUIRED REPAIR

Plot held-out ROC-AUC against training size across at least five points on de-duplicated data and show the plateau. Report the effective distinct-row count as the sample size in the abstract.

DEFENSIBLE REPLY

On a corpus where an unregularised logistic regression scores 0.999927, a learning curve would show a plateau at every sample size I could plot. The comment is unmeasured, which is fair; the claim it makes is not in doubt here.

A manufactured class balance, never restored

EVIDENCE

Section 3's provenance table says the capture holds ten point four million flows at seventy-nine percent benign. Cell 11 keeps every malicious flow — two million one hundred and eighty-five thousand of them — and caps benign at one and a half million, giving the printed malicious rate of 0.5930, correctly labelled a bounded sample. So the majority-class baseline of 0.5930 that every accuracy figure is judged against is a number you created. Nowhere is precision reported, and nowhere is any rate multiplied back out to the capture's real benign population.

SCIENTIFIC CONSEQUENCE

Let me be fair about what does and does not move. Uniform subsampling of the benign class leaves false-positive rate and recall unbiased, so your seven-in-three-hundred-and-seventy-five-thousand is honest, and scaling it to the capture's real benign population — seventy-nine percent of ten point four million, about eight point two million flows — gives roughly one hundred and fifty false alarms across the entire capture. That is genuinely good and I will say so. What does move is precision and the baseline: at the real balance there are five and a half times as many benign flows per malicious flow as in your sample, and neither precision nor any alert-count figure appears in the notebook at all.

REQUIRED REPAIR

Report precision and alerts-per-hour at the capture's own seventy-nine to twenty-one balance, alongside the bounded-sample figures, and label which is which.

DEFENSIBLE REPLY

The rate is labelled a bounded sample everywhere it appears, and both operational quantities I report -- false-positive rate and per-family recall -- are unbiased under uniform benign subsampling. Only quantities I do not report would move.

Defaults throughout, cross-validation on a slice

EVIDENCE

Cell 16 instantiates four learners at fixed defaults — sixty trees for RandomForest, eighty for the boosters. There is no grid, no random search and no nested cross-validation anywhere. Cell 24 runs three folds on the first forty thousand rows of the hundred-and-twenty-thousand-row training subsample, and the conclusion quotes its 1.0000 as the winning learner's cross-validated ROC-AUC.

SCIENTIFIC CONSEQUENCE

A comparison of four learners at four sets of arbitrary defaults is a comparison of defaults, and quoting a figure computed on a third of the training data beside the headline implies a corroboration that was not performed.

REQUIRED REPAIR

Nested cross-validation with an inner selection loop and an outer estimation loop, run on the same data the headline model sees.

DEFENSIBLE REPLY

Logistic regression at defaults scores 0.999927. There is nothing for a grid to find.

One draw decides which benign flows exist

EVIDENCE

Cell 10 sets RANDOM-STATE to zero with the comment, in the notebook's own words, single seed used everywhere. Cell 11 then draws the benign subsample with a separate fixed generator seeded at zero, selecting one and a half million benign flows from a population the provenance table puts at about eight point two million. That single draw fixes the corpus, the baseline, and therefore every rate downstream. Nothing is repeated.

SCIENTIFIC CONSEQUENCE

The notebook reports no distribution for anything. There is no evidence of how much the headline, the baseline or the per-family recalls move when a different six-sevenths of the benign class is discarded, and the discarded fraction here is very large.

REQUIRED REPAIR

Repeat the benign draw and the headline fit across at least ten seeds and report means and standard deviations for the headline, the operational false-positive rate and each per-family recall.

DEFENSIBLE REPLY

The benign draw is uniform, so false-positive rate and per-family recall are unbiased under any seed, and the audit grade is stable across eight seeds. What the draw fixes is the manufactured baseline, which is labelled a bounded sample wherever it appears.

ORDERED CORRECTION

Correct 'six-sevenths' to the measured figure: 1,500,000 benign flows are kept from 8,262,389, so 81.8% of the benign class is discarded.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 921,350 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So the ruling is this. You built an honest notebook around a number that measures a flood, and then you nominated per-family recall as the decisive evidence and printed it with a family missing and no supports attached. The corrections are not expensive: print every family with its count, read field one and field three of the file you already opened, and test on the eleven captures that were already unzipped on your disk. Until then, this is a measurement of capture thirty-five, and capture thirty-five is not the phenomenon.

NetFlow-Standardized Intrusion Detection: NF-CSE-CIC-IDS2018-V2

Kaggle dhoogla's re-upload of NF-CSE-CIC-IDS2018-V2 — CSE-CIC-IDS2018 re-featured by Sarhan and colleagues into the standard NetFlow v2 schema — read as 17,129,715 flows described by 40 features, 11.84 percent of them attacks.

HEADLINE

LightGBM, held-out ROC-AUC 0.988914, with a data-trust grade of B and an ablation the conclusion says refutes a single-feature story.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 3 critical (FATAL)
- 20 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook's abstract understates its corpus rather than overstating it, which is rarer than it should be. Its per-family recall table is the most informative in the four I was given — fourteen families, none silently dropped — and it correctly leads with the worst of them, Infiltration at 0.276, as the operationally important result rather than the aggregate. Its contamination is genuinely zero and its exact-duplicate rate is genuinely zero. And it names the transfer experiment it owes and points the reader at the notebook that runs it. That is a good scientific instinct.

EXAMINER RULING

You advertised seventeen million flows, trained on seven tenths of one percent of them, and then reported an ablation effect three and a half times smaller than the noise floor printed in the row above it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Fifty-six millionths against a hundred and ninety-two

EVIDENCE

Cell 22's table has three rows. The headline reads 0.988914. The de-duplicated row reads 0.988722 — and that row removed zero percent of the rows, as the table itself says. Identical data, same seed, and the two numbers differ by 0.000192. Your conclusion identifies exactly what that is, in its own words: the difference is re-split noise, not a de-duplication effect. So you have printed your instrument's noise floor: one hundred and ninety-two millionths. The third row, dropping the strongest feature, reads 0.988970 — a movement of 0.000056 from the headline. Fifty-six millionths. Three and a half times smaller than the noise you just measured. And the conclusion says: the ablation refutes a single-feature story, so the separability is multi-feature.

SCIENTIFIC CONSEQUENCE

You measured your own noise floor and then reported an effect underneath it as evidence. That is not a small methodological slip — it is the load-bearing inference of the notebook, and the arithmetic that destroys it is two rows of the same printed table. The drop-the-feature row also lands above the headline, which is consistent with reading jitter rather than an effect.

REQUIRED REPAIR

Fix one held-out split across all three arms, refit only the model, and repeat each arm at least ten times. Report each as a mean with a spread and read the differences against that spread. On this evidence you will need roughly a hundred repeats before a fifty-six-millionth effect is measurable at all — which is itself the finding.

The whole point of the schema, not exercised

EVIDENCE

Section 1 states it outright: the standard feature set was proposed to enable cross-dataset comparison, so it is the right lens for asking whether high scores are feature shortcuts or transferable signal. The conclusion repeats it and concedes, in its own words, that train-on-one, test-on-another is a test we did not run here, and points the reader at notebook 36 of this series, which runs exactly that experiment on four NetFlow-standardized corpora and prints the transfer matrix. Sarhan's paper names three sibling datasets in the same schema — NF-UNSW-NB15-v2, NF-BoT-IoT-v2 and NF-ToN-IoT-v2 — and your own reference list cites it.

SCIENTIFIC CONSEQUENCE

The notebook selects this corpus for one stated reason, poses the question that reason enables, and answers a different question. And the capability is not hypothetical or expensive: it exists in the same codebase, in the same series, three notebooks along, and the sibling corpora are in the same schema by construction. You identified the experiment that would settle your central claim, wrote down that it was the honest next step, and did not run it. Pointing at notebook 36 is a citation, not a result — it does not transfer notebook 36's evidence onto this number.

REQUIRED REPAIR

Train on NF-CSE-CIC-IDS2018-v2 and test on NF-UNSW-NB15-v2 and NF-ToN-IoT-v2, both directions, and print the transfer matrix in this notebook. Report the drop from in-distribution to transfer as the headline, because that drop is the quantity the schema was built to expose.

Seven tenths of one percent of the advertised corpus

EVIDENCE

Cell 11 prints 17,129,715 rows — seventeen million one hundred and twenty-nine thousand flows. Cell 16 prints: trained on 120,000 as a stratified subsample, held-out 4,282,429. Do the division: one hundred and twenty thousand over seventeen million one hundred and twenty-nine thousand is zero point seven percent. Seven tenths of one percent. The held-out set is thirty-five times the size of the training set. The comment on the capping line reads: because these learners saturate long before then on this data. There is no learning curve anywhere in the notebook. Cell 11's assert gates only that the row count exceeds one million.

SCIENTIFIC CONSEQUENCE

You have justified throwing away ninety-nine point three percent of your data with an empirical assertion written in a code comment and never measured. Saturation is a curve, not an assumption. And the shape of the design is upside down: you evaluate on four point three million rows a model fitted on a hundred and twenty thousand, which spends all your statistical power on precision in the estimate and none on the estimate being of anything worth measuring. Fourteen attack families share those hundred and twenty thousand rows, and Infiltration — the family at 0.276 recall — gets whatever fraction of them its base rate allows. The obvious first hypothesis for why Infiltration fails is that the model saw too few of them, and your design makes that hypothesis untestable.

REQUIRED REPAIR

Plot held-out ROC-AUC and per-family recall against training size across at least five points up to the full training half, and show where each plateaus. Infiltration recall is the curve that matters. If it has not plateaued at one hundred and twenty thousand, train on more.

DEFENSIBLE REPLY

Infiltration recall against training size: 0.226 at 30,000, 0.276 at 120,000, 0.303 at 480,000, 0.314 at 2 million, 0.326 on the full training half. A hundred and seven times the data buys five points of recall on a family that stays under a third. So the hypothesis is testable, I have tested it, and Infiltration is a hard-family problem, not a sample-size problem. And the headline moves UP by 0.00127 -- capping at 120,000 understates my result.

ORDERED CORRECTION

Delete 'your design makes that hypothesis untestable' -- it is one cell, and the answer is +0.050 Infiltration recall. Keep and strengthen the saturation charge with the measured curve, and add the genuinely sharp consequence: the 0.00127 the cap discards is 6.6 times the notebook's own printed 0.000192 noise floor and roughly four times the 0.000322 margin used to crown LightGBM.

The forty-three-field schema you ran has forty-one fields

EVIDENCE

Your literature review cites Sarhan and colleagues as defining the forty-three-field NetFlow v2 schema and the NF-star-v2 datasets used here. The Kaggle archive ships two files. One is a document called NetFlow v2 Features dot csv which lists exactly forty-three features, and the first two entries in that list are IPV4 source address and IPV4 destination address. The other is the parquet your loader reads. I inspected its schema: it has forty-three columns, and neither IP field is among them — the two additional columns are Label and Attack. So the payload carries forty-one of the forty-three documented features. Cell 11 prints forty features after dropping the two label columns and one constant. Your notebook never prints a column list anywhere.

SCIENTIFIC CONSEQUENCE

You have asserted, in a literature review, a property of the data you ran — that it is the forty-three-field schema of a named paper — and the file on disk does not have that property. The mirror's own documentation contradicts the mirror's own payload, and the only reason you did not catch it is that no cell in the notebook ever prints what columns it loaded. This is the minimum a doctoral artefact owes: say what you actually ran. I will add, in fairness, that the accident works in your favour on one axis — because those two fields are missing, your model cannot memorise which host, which is a shortcut that would have been open to it and is not. But you did not know that, and you cannot take credit for a defence you were unaware of.

REQUIRED REPAIR

Print the loaded column list. Then reconcile it against the authoritative NF-v2 distribution published with the Sarhan paper, report the row-count and column-set agreement, and either use the authoritative file or state precisely which schema fields your run is missing.

DEFENSIBLE REPLY

You concede the accident works in my favour: with no addressing fields, the model cannot memorise which host. And the discrepancy is not wholly invisible from inside the notebook -- cell 11 prints 'loaded 17,129,715 rows x 40 features', which a reader can set against 43. What I owe is the column list and the reconciliation, which is a documentation fix, not a result-invalidating one.

ORDERED CORRECTION

Soften 'the discrepancy is invisible from inside the notebook' -- the printed 40-feature count is visible, though which fields are missing is not. FATAL is not sustainable for a defect the charge itself says improved the science.

The mirror stripped the only grouping key there was

EVIDENCE

The conclusion states: the split is random, not temporal or entity-grouped, so every number above measures in-distribution separability only. The NetFlow v2 schema carries no absolute timestamp at all — only durations — so a temporal split is unavailable from any distribution of this corpus, and I will not fault you for that. But the schema does carry the two IP address fields, and those are exactly the grouping keys a GroupShuffleSplit would use. They are absent from the parquet you read. Cell 11's drop list is worth reading aloud: it is the set of Label, Attack, y and family, unioned with an empty set. The slot for identifier columns to remove is present in your template and left empty, so no identifier was ever considered.

SCIENTIFIC CONSEQUENCE

Three verified facts, and their consequence. No entity-grouped evaluation appears in this notebook. None is constructible from the file it read, because the grouping keys the schema defines are not in the payload. And the notebook cannot know either of those things, because no cell ever prints the column list. So the scope sentence in your conclusion is not a statement about a methodological choice you weighed — it is a statement about a constraint you never discovered. The sister notebooks show the contrast: notebook 17's drop list names sender account, receiver account, IP address and timestamp, and notebook 19's read list deliberately omits the host field with a comment saying it does so so the model cannot memorise which host. Yours removes the two label columns and nothing else, because there was nothing else there and nobody looked.

REQUIRED REPAIR

Obtain a distribution that retains the addressing fields, group on source address, and report the held-out score under the grouped split. Failing that, state explicitly in the abstract that no entity-grouped evaluation is constructible from this file, so the reader knows the in-distribution number is the only number available rather than the one you preferred.

DEFENSIBLE REPLY

I confirm the constraint: IPV4_SRC_ADDR and IPV4_DST_ADDR are absent from the parquet, so no grouped split is constructible from this file, and no timestamp exists in the NetFlow v2 schema at all, which you rightly decline to fault me for. My scope sentence -- 'the split is random, not temporal or entity-grouped' -- is true as written. What you have caught is that I did not print the column list and therefore could not say the choice was forced rather than made.

ORDERED CORRECTION

Re-file at CONTESTED per CORRECTION 3's Tier-2 rule, which this fault's own evidence establishes. The A2 restraint elsewhere in the fault -- 'the NetFlow v2 schema carries no absolute timestamp at all, and I will not fault you for that' -- is correct and I verified it independently; that judgement should be preserved verbatim.

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

6

MAJOR

UNTESTED

REMECHANISED

FEATURES ENCODE THE GENERATOR'S CONFIGURATION

Source and destination port are features

EVIDENCE I inspected the parquet schema. Layer four source port and layer four destination port are both present as integer columns, alongside protocol and layer seven protocol. Your cell 11 drop list removes nothing but the two label columns, so all four are features. Your own section 3 concedes the risk in its own words: NetFlow header fields, ports and protocol, can act as shortcuts, which the audit probes. Cell 20's audit scores each column separately, one at a time.	SCIENTIFIC CONSEQUENCE In a lab capture, an attack tool is launched against a chosen service on a chosen port from a chosen host, and the destination port therefore records the experimenter's script rather than the adversary's behaviour. SSH brute force at 0.999 recall on a corpus where port twenty-two is a feature is close to a tautology. And your audit cannot detect this, because a shortcut carried by a pair of columns — port plus protocol — is invisible to a diagnostic that scores columns individually. You wrote that the audit probes the header fields. It probes them one at a time, which is not the same thing.	REQUIRED REPAIR Ablate the header group as a group — source port, destination port, protocol, layer seven protocol — retrain, and report the held-out ROC-AUC and the full per-family recall table on behaviour-only features. If SSH brute force falls from 0.999, that is the finding.
DEFENSIBLE REPLY You said 'if SSH brute force falls from 0.999, that is the finding'. It does not fall. It is 0.999 on behaviour-only features. The aggregate moves 0.988914 to 0.988833, a change of 0.000081 -- less than half the 0.000192 noise floor you correctly identified in fault 2. What actually falls when the header group is removed is DoS attacks-Slowloris, from 1.000 to 0.741, and Bot from 1.000 to 0.975. Your experiment was worth running and it returns a different answer from the one you predicted.		
ORDERED CORRECTION Strike the SSH-brute-force-is-a-tautology claim, which the ablation refutes. Re-mechanise onto the result that does support the concern and that the one-at-a-time audit genuinely cannot see: the four header fields ALONE reach a held-out AUC of 0.979138 against the full model's 0.988914. That is near-sufficiency in a column group, measured, and it is exactly the failure mode a per-column rank statistic is blind to.		

Zero duplicates on forty continuous columns

EVIDENCE

Cell 20 prints an exact-duplicate row rate of 0.000 and a train-test exact-row contamination of 0.000, and the contamination sub-grade is A. The features include source-to-destination bytes per second and destination-to-source bytes per second as double-precision floats, and average throughputs as sixty-four-bit integers. The contamination test compares feature tuples rounded to six decimal places for exact equality. No nearest-neighbour distance is computed anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Two continuous throughput values agreeing to six decimal places is not something that happens. So the zero is not a measurement that this corpus is clean — it is a measurement that this test cannot fire on this corpus, and the A grade it earns is unearned in the sense that no evidence was gathered. The leakage that actually matters here is two flows from the same attack burst, seconds apart, differing by a few bytes. Those are near-neighbours, not duplicates, and your split scatters them across the boundary by construction. The diagnostic that would find them was not run.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from held-out rows to the training set in a scaled feature space, plot it, and report the fraction below a stated threshold. That fraction, not the exact-duplicate rate, is the contamination number for a continuous-valued corpus.

DEFENSIBLE REPLY

Section 10 states exactly what the diagnostic is: the fraction of held-out rows that are exact duplicates of training rows. It reports that quantity truthfully as 0.000. A nearest-neighbour distance distribution from 4,282,429 held-out rows to 12,847,286 training rows in 40 dimensions is not a cell in a teaching notebook, and an approximate version would need its own validation.

ORDERED CORRECTION

Replace 'the A grade it earns is unearned' with the accurate form: the grade is earned against a check that cannot fire on continuous data, so it is uninformative rather than false.

The margin is a fifth of the fold spread, and the axes disagree

EVIDENCE

Cell 16 puts LightGBM at ROC-AUC 0.988914 and XGBoost at 0.988592 — a margin of 0.000322, about three ten-thousandths. Cell 24 prints the winner's three-fold cross-validated spread as plus or minus 0.0017, seventeen ten-thousandths. The margin is roughly a fifth of the spread. And on the accuracy column the ordering changes: RandomForest prints the highest accuracy at 0.995050, above LightGBM's 0.994956, while ranking third on AUC. No significance test appears anywhere.

SCIENTIFIC CONSEQUENCE

You have no evidence that LightGBM is better than XGBoost on this corpus, only that it drew the higher number once, and every downstream diagnostic — the importance chart, the ablation, the operational false-positive rate, the fourteen per-family recalls — is computed on that arbitrary selection. The disagreement between the accuracy and AUC orderings is the notebook telling you this itself: two axes, two winners, no test on either. Your own conclusion warns the reader to read the comparison table rather than generalise from the best row, and then the notebook generalises from the best row.

REQUIRED REPAIR

McNemar on the paired held-out predictions of the top two, or a paired bootstrap interval on the AUC difference. If the interval covers zero, say so and carry both models through the diagnostics.

DEFENSIBLE REPLY

0.988914 and 0.988592 are scored on the same 4,282,429 rows, so their difference is paired and its standard error is far smaller than either model's marginal spread. The ± 0.0017 fold spread describes how one model's score moves across different data; it does not bound how two models differ on the same data. The correct instruments are the ones you name -- DeLong or McNemar -- and the honest statement is that neither was run. Note also that the margin of 0.000322 exceeds the notebook's own printed 0.000192 re-split noise floor.

ORDERED CORRECTION

Replace 'the margin is a fifth of the fold spread' with 'no paired test was run'. The accuracy/AUC disagreement -- RandomForest highest on accuracy at 0.995050, third on AUC -- is the durable half and should lead.

The right diagnostic, drawn and never read

EVIDENCE

Cell 11 prints a positive rate of 0.1184. Cell 18 calls precision-recall-curve and plots it as the third panel of the first figure. Nowhere in this notebook is average-precision-score called, no area under that curve is printed, and no model is ranked by it. That holds across the whole thirty-six-notebook series: the curve is drawn thirty-six times, and its area is computed zero times.

SCIENTIFIC CONSEQUENCE

At roughly one attack in eight, ROC-AUC is dominated by the ranking of the benign mass and moves very little as false positives multiply. Average precision is the number that prices them and that would let a reader judge whether Infiltration at 0.276 recall can be improved at acceptable cost. You computed the curve. You rendered it. And then you selected your winner, wrote your conclusion and assigned your trust grade using a metric that cannot see what the curve shows. The objection is not omission — it is that the diagnostic was performed and then declined.

REQUIRED REPAIR

Average precision with a bootstrap confidence interval reported beside the ROC-AUC, and precision at fixed recall stated per family — particularly for Infiltration.

Seventy-two percent of infiltration missed at an unexamined default

EVIDENCE

Cell 18 sets the prediction as probability greater than zero point five. From that line come the operational false-positive rate of 0.0002 with 665 benign flows flagged out of 3,775,421, and the fourteen per-family recalls, the worst of which is Infiltration at 0.276 — so seventy-two percent of infiltration flows are missed. No cost matrix appears anywhere and no other threshold is examined.

SCIENTIFIC CONSEQUENCE

Your conclusion is right that the Infiltration gap, not the aggregate, is the operationally important result. But that gap is measured at one threshold, and that threshold is the default argument of a comparison operator, not a decision anyone made. With 665 false positives in nearly four million benign flows you have an extraordinary amount of headroom to lower the threshold and buy Infiltration recall. Whether that trade is available is the single most useful question this notebook could answer about this corpus, and it is not asked.

REQUIRED REPAIR

Sweep the threshold and plot Infiltration recall against false-positive count. State a false-alarm budget in alerts per analyst per day and read the achievable Infiltration recall off that curve.

DEFENSIBLE REPLY

Here is your curve, on 28,778 held-out Infiltration flows. At 0.5: recall 0.2762, 665 false positives. At 0.1: 0.3115, 4,682. At 0.02: 0.3474, 77,526. At 0.01: 0.3916, 204,465. To recover eleven points of Infiltration recall I must raise false alarms from 665 to over two hundred thousand. You are right that I never asked; you are wrong that the trade was available.

ORDERED CORRECTION

Add the measured sweep. 'Whether that trade is available is the single most useful question this notebook could answer' is correct and the answer is no, which is a stronger charge against the corpus than against the notebook.

A rate printed, never turned into a count

EVIDENCE

Cell 18 prints 665 benign flows flagged out of 3,775,421. Sommer and Paxson is in your reference list. Let me do the arithmetic from your integers rather than your rounded rate. Your held-out benign population is 3,775,421 flows. Your whole corpus is 17,129,715 flows at a positive rate of 0.1184, so its benign population is about 15.1 million — very nearly four times the held-out set. Four times 665 is about two thousand seven hundred false alarms across the entire corpus. No such figure appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Two thousand seven hundred is, I will say plainly, a good number, and if the notebook had computed it the reader would know that. It did not, so the reader knows nothing about admissibility in either direction. And the number that matters is not this one — it is the false-alarm count at whatever threshold recovers the seventy-two percent of Infiltration you are currently missing, and that count is precisely what the unread precision-recall curve would have given you. Printing a rate to four decimal places and never converting it into alerts is the whole base-rate failure in one line.

REQUIRED REPAIR

Print alerts per day at a stated flow volume alongside the rate, and print the same figure at the threshold that achieves a stated Infiltration recall, so the trade is visible as two numbers rather than one.

DEFENSIBLE REPLY

You compute the figure yourself and call it good. What separates this from fault 10 is only which stipulated denominator is used, and both faults require the same fix: read an operating point off the precision-recall curve and price it.

ORDERED CORRECTION

Merge with fault 10, or keep only the distinct half -- that the alert count at the Infiltration-recovering threshold is the number that matters, which I measured at 204,465 for recall 0.392.

A statement about how the corpus was generated, with no measurement of it

EVIDENCE

The conclusion says: the separability is multi-feature, and, in its own words, that reflects how this corpus was generated, not one leaky column. The evidence offered is that dropping DURATION_IN moves the held-out AUC by 0.000056 — which fault two shows is below the noise floor printed one row above. No measure of class-conditional distribution overlap appears in the notebook, no redundancy structure is quantified, and no feature group is ablated as a group.

SCIENTIFIC CONSEQUENCE

There are two competing explanations for a flat ablation — the classes really are separable by many redundant features because the distributions barely overlap, or the ablation removed the wrong column. Your section 11 prose states both possibilities correctly, which is to your credit. But the notebook then selects the first explanation and states it as a fact about how the data was generated, with no measurement of generation, of overlap, or of redundancy anywhere. An unmeasured causal claim in a conclusion is the thing a doctoral defence exists to catch.

REQUIRED REPAIR

Measure the overlap — a two-sample divergence estimate between class-conditional feature distributions — and ablate correlated column groups rather than single columns. If you wish to claim the separability reflects the corpus's generation, that requires a measurement of the corpus.

DEFENSIBLE REPLY

The redundancy I assert is demonstrable in one cell and I have run it: the four header fields alone reach 0.979138 while removing them costs 0.000081, which is direct evidence that the same information is carried several times over across the feature set. The claim is right; you are right that I asserted rather than measured it.

An intrusion detector never shown an evasive intruder

EVIDENCE

The feature set includes source and destination port, minimum and maximum time-to-live, TCP window sizes, retransmission counts, the five packet-size histogram buckets and the two average throughput columns. Section 1 poses the question as whether high scores are feature shortcuts or transferable signal. No perturbation, no evasion experiment and no worst-case recall under any budget appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Every one of those columns is under the attacker's control. Padding packets moves the histogram buckets and the length extremes. Throttling moves both throughput columns and both durations. Setting a time-to-live moves the TTL columns. Choosing a non-standard port moves the header fields. The adversary in your held-out set is a lab script that made none of those choices, so your 0.988914 is a measurement against a non-adversary. Infiltration is already your weakest family at 0.276 against an opponent who is not trying.

REQUIRED REPAIR

State a perturbation budget in flow terms — bytes of padding, milliseconds of delay, permitted port ranges — search for the cheapest perturbation that flips the prediction, and report worst-case per-family recall under that budget beside the clean table.

DEFENSIBLE REPLY

You ask me to pad packets and throttle throughput. This file contains 40 aggregated flow statistics; there are no packets to pad. Mapping a padding budget into the five packet-size histogram buckets, the two throughput columns and both duration columns consistently would require regenerating the corpus from the original pcaps, which is not an experiment you costed. A feature-space budget is constructible and would have been the right ask.

ORDERED CORRECTION

Restate the required experiment in feature space rather than in packets, since the corpus is an aggregate and no packet-level perturbation is applicable to it.

Honesty text describing a different notebook

EVIDENCE

The conclusion carries a paragraph headed disclosed limitation about categorical columns being integer-encoded before the split and the encoder seeing the test set's category values. I inspected the parquet schema: every feature column is numeric, and the one string column, Attack, is dropped. So that loop touches nothing. The same paragraph explains that the de-duplication variant keeps the first label when a feature vector appears twice — on a corpus where cell 20 prints a duplicate rate of 0.000, so no vector appears twice.

SCIENTIFIC CONSEQUENCE

Two of the notebook's stated limitations are inapplicable to the notebook. That is a small thing on its own and a significant thing in aggregate, because it tells the reader that the honesty section was assembled rather than derived, and a reader who notices it has no way to know which of the remaining disclosures were derived and which were assembled. Disclosure earns credibility only when it is specific to what was run.

REQUIRED REPAIR

Delete disclosures that do not apply, and state in their place the limitations that do — that the payload is missing two documented schema fields, and that no entity-grouped split is constructible from it.

DEFENSIBLE REPLY

The paragraph carries its own hedge -- 'on an all-numeric corpus that step is a no-op' -- which is the case that applies here, so the disclosure is technically correct even though it never says which case it is in.

ORDERED CORRECTION

This is one of the sharpest small faults in the batch and its required fix is exactly right: replace the inapplicable disclosures with the two that do apply -- the payload is missing two documented schema fields, and no entity-grouped split is constructible from it.

A zero drawn from one percent of the held-out set

EVIDENCE

Cell 20 measures contamination against the first fifty thousand held-out rows and prints 0.000. The held-out set is 4,282,429 rows, so the sample covers one point two percent of it. No confidence interval accompanies the figure, and the grading line then reads the unqualified zero against cutoffs at five and fifteen percent.

SCIENTIFIC CONSEQUENCE

A zero from fifty thousand draws is consistent with a true rate up to roughly one in seventeen thousand, and the notebook presents it as though it were a census. On this corpus the true rate does appear to be zero, so no harm results — the defect is that the method could not have distinguished a real zero from a small non-zero, and the sub-grade A is awarded as though it could.

REQUIRED REPAIR

Report a binomial confidence interval on the contamination rate, or compute it exactly against the full held-out set, which on a hashed comparison is entirely affordable at this scale.

DEFENSIBLE REPLY

The corpus has 40 mostly-continuous columns and a whole-corpus exact-duplicate rate of 0.000, so a non-zero contamination rate is arithmetically near-impossible: contamination requires exactly duplicated feature vectors and there are none.

A letter from cutoffs on a rank statistic

EVIDENCE

Cell 20 grades the single-feature check against cutoffs at 0.85, 0.95, 0.99 and 0.999, and prints B for a value of 0.9217. On this corpus the letter is stable — the nearest cutoff is 0.95, and the independent re-derivation for this review found eight seeds spanning 0.9203 to 0.9264, all comfortably clear of it, so I will not accuse this notebook of a knife-edge. The objection is different. Notebook 17 in this same series runs this identical instrument, returns its top grade of A, and its own conclusion then explains that the headline is very likely a fill-value artifact the instrument is structurally incapable of seeing.

SCIENTIFIC CONSEQUENCE

The letter is presented as a summary judgement on data trust and it is a summary of two specific checks, neither of which is the check that fails elsewhere in this same series. A grade that is stable and measuring the wrong quantity is not better than an unstable one. A reader who takes the B as a statement about whether this headline is trustworthy has been told something the instrument does not know.

REQUIRED REPAIR

Report the two underlying quantities with intervals and delete the letter, or redefine the grade so it asks whether any single artifact accounts for the whole headline — which is a drop-and-refit on the model's own top feature, repeated enough times to clear the noise floor.

DEFENSIBLE REPLY

The grade is described as the worse of two named sub-checks, both sub-grades are printed separately, and the conclusion says which one drives the letter. It is offered as triage, not as a verdict.

17

MAJOR

UNTESTED

DOWNGRADED

IMPURITY IMPORTANCE REPORTED AS EXPLANATION

An importance chart with no numbers under it

EVIDENCE

Cell 18 plots the model's feature-importances attribute as a horizontal bar chart of the top twelve. No importance value appears as text anywhere in the notebook. The feature set is dominated by high-cardinality continuous columns — byte-per-second rates, average throughputs, durations in milliseconds — sitting beside low-cardinality integers such as protocol and the TCP flag columns.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward columns with many distinct values, which on this feature set means it is biased toward exactly the continuous rate columns and away from the header fields whose shortcut potential your own section 3 flags. So the chart that a reader would use to check the header-shortcut concern is the chart least able to answer it, and it is a picture rather than a table, so nobody can check it without re-running you.

REQUIRED REPAIR

Permutation importance computed on the held-out set, or SHAP values, printed as a numbered table alongside the figure.

DEFENSIBLE REPLY

The chart is decorative here. No sentence in the conclusion depends on it.

18

MAJOR

MEASURED

DOWNGRADED

FALSE PRECISION

Six decimals where three are supported

EVIDENCE

Cell 16 rounds every model AUC to six decimals, so the headline reads 0.988914. The notebook's own variability figure, cell 24's fold spread, is 0.0017 — seventeen ten-thousandths. So the fourth, fifth and sixth decimals sit below the noise floor, and both the ablation deltas discussed in fault two live entirely inside them.

SCIENTIFIC CONSEQUENCE

Six digits is a claim about resolution and three of them carry none. The cost is not cosmetic: it is what allows a fifty-six-millionth movement to be printed in a table and read as a result, and what allows a three-ten-thousandth margin between two learners to look like a ranking.

REQUIRED REPAIR

Report to the resolution the estimator supports — three decimals here — and report every difference as an interval rather than as digits.

DEFENSIBLE REPLY

Six decimals exist so a perfect score cannot hide inside a rounding call.

19

MAJOR UNTESTED DOWNGRADED NO MODEL SELECTION

Defaults throughout, cross-validation on a third of a subsample

EVIDENCE

Cell 16 instantiates four learners at fixed defaults — sixty trees for RandomForest, eighty for the boosters. There is no grid, no random search and no nested cross-validation anywhere in the notebook. Cell 24 runs three folds on the first forty thousand rows of the hundred-and-twenty-thousand-row training subsample, and the conclusion quotes its 0.9883 as the winner's cross-validated ROC-AUC.

SCIENTIFIC CONSEQUENCE

Four learners at four sets of arbitrary defaults is a comparison of defaults, not of learners, and it is the comparison the notebook exists to make. Quoting a figure computed on forty thousand rows beside a headline fitted on a hundred and twenty thousand implies a corroboration that was not performed — a third of the data is not the pipeline.

REQUIRED REPAIR

Nested cross-validation, with an inner loop selecting hyperparameters and an outer loop estimating performance, run on the same data the headline model sees.

DEFENSIBLE REPLY

Untuned defaults applied uniformly compare learners on a fixed budget, and the conclusion labels 0.9883 as a 3-fold CV figure rather than as the headline.

20

MAJOR DOCUMENTED DOWNGRADED SINGLE SEED FOR THE HEADLINE

One seed, used everywhere, as the comment says

EVIDENCE

Cell 10 sets RANDOM-STATE to zero with the comment, in the notebook's own words, single seed used everywhere. That seed drives the split, the stratified subsample of one hundred and twenty thousand rows from seventeen million, all four fits and every ablation re-split. Nothing is repeated across seeds, here or in any of the thirty-six notebooks.

SCIENTIFIC CONSEQUENCE

The notebook reports no distribution for anything, so it cannot support any of the several comparisons it draws between numbers separated by ten-thousandths — and the subsample being drawn is seven tenths of one percent of the corpus, which is the regime in which draw-to-draw variability is largest.

REQUIRED REPAIR

At least ten seeds for the subsample draw, the split and the fit, with mean and standard deviation reported for the headline and for each per-family recall.

DEFENSIBLE REPLY

Infiltration recall rests on 28,778 held-out positives and DDOS attack-HOIC on 267,083. Those are not seed-sensitive numbers. What a seed sweep would move is the third decimal of the aggregate, which no conclusion depends on.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 4,282,429 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So the ruling is this. Two rows of your own ablation table settle the central inference against you: you printed a noise floor of a hundred and ninety-two millionths and then reported a fifty-six-millionth effect as evidence. Around that sits a corpus you advertised at seventeen million and used at a hundred and twenty thousand, read from a mirror whose payload does not match its own documentation. Run the transfer test in notebook 36 on this data, repeat the ablation until the noise is bounded, and print the columns you loaded — and the next version of this will be a real result.

Host-Based Intrusion Detection on BETH Kernel Telemetry

BETH: three million eight hundred and seven thousand real kernel process events captured on cloud honeypots, hand-labelled suspicious and confirmed-evil, reduced here to five behaviour columns.

HEADLINE

LightGBM, held-out ROC-AUC of point nine seven four — printed to six decimals as nought point nine seven three six three nine — on nine hundred and fifty-one thousand seven hundred and ninety-nine held-out kernel events, with the abstract calling the signal for confirmed intrusions real and robust.

DEFENSE

F

Selected-model estimand

DOCTORAL

6

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

8.5

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 0 Honesty / scope

- 22 live faults
- 8 critical (FATAL)
- 14 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me be clear about what this notebook does better than most of the published literature. It measured train-test contamination at all, which almost no intrusion-detection paper does, and when the answer came back ugly it printed the ugly answer and graded its own data F. It dropped `userId` with a precise, citation-grounded argument about the human labeller's cue, and then it went back and corrected its own loader comment for overstating that argument — that is the behaviour of a scientist. It reported evil and suspicious-only recall separately so the weak number, nought point three nine eight, is visible rather than buried in an aggregate. And it flagged the contamination figure as a sampled estimate measured on the first fifty thousand held-out rows; an independent re-run of that cell on a randomly chosen window returned nought point nine nine four five, which vindicates the estimate rather than undermining it. None of that is in dispute. All of it is disclosure. And disclosure is not mitigation.

EXAMINER RULING

I am rejecting this notebook with prejudice, because ninety-nine point five percent of what you called a held-out set is an exact copy of a row you trained on, and a number measured that way is not an estimate of detection — it is an estimate of memory.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The held-out set is the training set

EVIDENCE

Cell twenty prints train-test exact-row contamination of nought point nine nine five. Read that aloud: ninety-nine point five percent of held-out rows are exact copies of a training row. And look at what they are copies of. In cell sixteen the variable Xtr is rebound inside the if-len-greater-than-one-hundred-and-twenty-thousand block, so by the time cell twenty measures contamination it is comparing the test set against a training pool of only one hundred and twenty thousand rows. Apply the printed rate to the printed holdout: nine hundred and fifty-one thousand seven hundred and ninety-nine times nought point nine nine five is nine hundred and forty-seven thousand and forty. Subtract, and roughly four thousand seven hundred and fifty-nine held-out events out of nearly a million were genuinely unseen.

SCIENTIFIC CONSEQUENCE

A held-out score is an estimate of performance on data the model has not seen. If the model has already seen ninety-nine point five percent of it, the quantity being estimated is recall from a lookup table. There is no inferential path from nought point nine seven three six three nine to any claim about detecting an intrusion, because the experiment that would license that claim was never run. This is not an inflated estimate. It is an estimate of a different thing.

REQUIRED REPAIR

GroupKFold or GroupShuffleSplit keyed on hostName or processId. You had both columns in the dataframe — your own DROP list in cell eleven names processId, parentProcessId and hostName — and you removed them from the feature matrix instead of promoting them to grouping variables. That is a twenty-line change. Alternatively, run BETH's official cross-host holdout split, which you name twice and never execute.

Three point eight million rows, seven thousand distinct points

EVIDENCE

Cell twenty prints an exact-duplicate row rate for the whole corpus of nought point nine nine eight. About two rows in a thousand are distinct. Three million eight hundred and seven thousand one hundred and ninety-six times nought point zero zero two is on the order of seven thousand six hundred. Because the printed rate is rounded to three decimals the true count sits somewhere between roughly five thousand seven hundred and nine thousand five hundred — but on any reading it is thousands, not millions. And note that this figure and the contamination figure are two independent printed statistics that corroborate each other: if only about seven thousand distinct vectors exist, then a training draw of one hundred and twenty thousand rows covers essentially all of them, which is exactly why contamination comes back at nought point nine nine five.

SCIENTIFIC CONSEQUENCE

The subtitle claims at least one million real process events. That describes the download, not the experiment. The experiment has an effective n in the thousands, and every uncertainty statement in the notebook — including the cross-validation standard deviation of plus or minus nought point zero zero one zero — is computed as though the sample size were the row count. It is not. Rows are not observations when the rows repeat.

REQUIRED REPAIR

Report the distinct-vector count in the abstract alongside the row count, and compute a learning curve on distinct vectors to saturation. If the answer is that five columns admit only seven thousand distinct states, then say so, and say that the research question is whether seven thousand states suffice — which is a legitimate and interesting question, and a completely different paper.

The F grade is argued away on two thousand rows

EVIDENCE

The cell twenty-two table shows de-duplicated, one hundred percent rows removed, held-out AUC nought point nine nine zero eight six eight, and section thirteen describes that as rising to approximately one point zero. Follow the code. In cell twenty-two, Xdd is X dot drop underscore duplicates, and at a duplicate rate of nought point nine nine eight that leaves on the order of seven thousand rows, which the retrain helper then splits seventy-five twenty-five. So that number was measured on roughly two thousand held-out rows against a headline measured on nine hundred and fifty-one thousand seven hundred and ninety-nine — a factor of about five hundred. The surviving row count is never printed anywhere in the notebook. Your conclusion also states that the de-duplication variant keeps the first label when a feature vector appears twice, so wherever two identical event vectors carried different labels, one label was silently deleted.

SCIENTIFIC CONSEQUENCE

This is the single load-bearing inference in the notebook. It is the sentence that converts a self-assigned F into a survivable caveat, and it rests on an unreported sample size roughly five hundred times smaller than the headline. Worse, the keep-first rule deletes exactly the irreducible label disagreement that sets the ceiling on achievable accuracy. So the de-duplicated task is not the cleaned-up version of the original task. It is an easier task, on a tiny sample, and it cannot be used to certify the harder one.

REQUIRED REPAIR

Print the surviving row count. Print the label-disagreement rate among duplicate feature vectors — that rate is the Bayes noise floor and it is the most scientifically interesting number in this corpus. And do not use a two-thousand-row refit to overturn a diagnostic computed on a million rows.

You named the real test twice and ran neither

EVIDENCE

Section three says: we use a random stratified split, not BETH's official host-holdout split; the latter is the stronger generalization protocol; we flag it as the honest next step. Section thirteen says: we did not run BETH's official cross-host split; that split is the real generalization test; host-based IDS here should be judged on rare-evil recall under that split, not aggregate accuracy. You then report aggregate accuracy and AUC as the headline. Your section-two caveat table goes further and states in its own words that repeated events inflate a random split — which is precisely what your nought point nine nine eight duplicate rate then did.

SCIENTIFIC CONSEQUENCE

At doctoral level, naming the experiment that would settle your central claim is not a defence. It is a confession that you knew which experiment mattered. The split is published with the dataset. The grouping columns were in your dataframe and you deleted them. The cost of the experiment was an afternoon; the cost of the disclosure paragraph was nothing, and it bought nothing. You published the headline anyway, and the abstract calls the result real and robust — a word for which no experiment in this notebook provides any evidence, because robustness was never measured.

REQUIRED REPAIR

Run the official cross-host split, report rare-evil recall under it, and put that number in the abstract. If it collapses, that collapse is the finding and it is a better paper than this one.

Atlas proof

docs/course/security-ml-notebooks/nb21-beth-host-telemetry.ipynb · cells 4 and 25;
zero-based manifest cell IDs: 4=31d51412, 25=1356caab

Ninety-two percent accurate at chance-level ranking

EVIDENCE

Cell sixteen, the ranking table, LogisticRegression row: accuracy nought point nine two three four four three, ROC-AUC nought point six four six six six eight. The majority-class baseline printed two lines above is nought point eight eight zero three. So that model beats the baseline on accuracy by four percentage points while ranking barely better than a coin flip. Both numbers are in the same printed row. And I will be fair here: section thirteen does separate the two axes in general, writing that for ROC-AUC the trivial baseline is nought point five, not the accuracy baseline. What it never does is apply that distinction to the one row where the two axes disagree with each other.

SCIENTIFIC CONSEQUENCE

That single row is the empirical demonstration that on this corpus, at twelve percent positives, accuracy carries almost no information — and the notebook prints accuracy as its first column and as the axis on which it compares every model to the baseline. You had the counter-example in your own output. Reporting it and not reading it is the same failure as not having it.

REQUIRED REPAIR

Report average precision — the area under the precision-recall curve — as the headline, with a confidence interval, and report precision at a fixed operational recall. You already plot the precision-recall curve in cell eighteen. You never compute its area.

DEFENSIBLE REPLY

We rank on ROC-AUC, we print the baseline's ROC-AUC as 0.5 inside the same table, and section 13 separates the two axes explicitly; the charge reduces to our not having written a sentence about one row we already printed in full.

ORDERED CORRECTION

Drop the clause 'and as the axis on which it compares every model to the baseline' — selection is by roc_auc. The surviving charge is that the LR row is the corpus's own demonstration that accuracy is uninformative at 12% positives and it is never read.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You deleted the timestamp, so you had one

EVIDENCE

Cell eleven, the DROP list, contains the string timestamp. You cannot drop a column that does not exist. So the notebook proves, by its own deletion, that a temporal ordering was available on every one of the three point eight million events.

SCIENTIFIC CONSEQUENCE

Kernel telemetry is a stream. An endpoint detection product is judged on whether a model fitted on this week's events flags next week's, and that is the only question an operator asks. Dropping the timestamp as a feature was correct — it would have been a capture-identity leak. Failing to use it as a split axis is a separate decision, and it discards the one ordering that makes the evaluation resemble deployment.

REQUIRED REPAIR

Split temporally within each capture — train on the earlier events of every file, test on the later ones — and report the drop. Or run BETH's official cross-host split, which needs no assumption about the clock at all. What you may not do is delete the column and report nothing.

DEFENSIBLE REPLY

BETH concatenates nine separate captures, and if that timestamp is a per-capture relative clock rather than wall-clock, a naive global sort interleaves captures instead of ordering events — so a temporal split is not the one-line fix it appears to be, and the cross-host split we named is the better protocol anyway.

The no-op disclaimer is false on this corpus

EVIDENCE

Cell eleven fits a LabelEncoder over every object-dtype column of X before any split occurs. The section-thirteen disclosed-limitation paragraph then says, quote, on an all-numeric corpus that step is a no-op. But this corpus is not all-numeric. Section one names the five retained behaviour features as process name, event id, event name, argument count and return value, and process name and event name are strings. The encoder ran, and it ran across the whole corpus.

SCIENTIFIC CONSEQUENCE

The disclosure is boilerplate carried in from notebooks where it is true, and on this corpus it converts a real transductive step into a claimed non-event. A reader who trusts the ledger comes away believing nothing happened. Something happened: the category vocabulary was fitted with the test set in view, so a deployed system meeting an unseen process name has no defined behaviour.

REQUIRED REPAIR

Fit the encoder inside each fold, and add an explicit unseen-category bucket. Then correct the ledger sentence so it describes this corpus rather than a different one.

DEFENSIBLE REPLY

Read the whole sentence: we say the step is a no-op on an all-numeric corpus, and then we say in the next breath that it is still transductive here and that a deployed system would need an unseen-category bucket. The boilerplate is untidy; it is not a claim that nothing happened.

ORDERED CORRECTION

Re-file as: the ledger contains an inapplicable conditional clause. Delete 'converts a real transductive step into a claimed non-event' and 'A reader who trusts the ledger comes away believing nothing happened', both of which the following two sentences refute.

No search, and the cross-validation is on a third of train

EVIDENCE

Cell sixteen instantiates RandomForest at sixty trees, XGBoost at eighty rounds and depth six, and LightGBM at eighty rounds — fixed constants, no grid, no random search, no tuning of any kind. Cell twenty-four then sets cvX and cvy to Xtr dot iloc of colon forty thousand: the first forty thousand rows of the one-hundred-and-twenty-thousand-row training subsample. The nought point nine seven two three plus or minus nought point zero one zero that section thirteen carries as the winning learner's cross-validated score is that forty-thousand-row number.

SCIENTIFIC CONSEQUENCE

A model comparison without model selection compares four defaults, not four methods, so the declared winner is an artifact of which library ships which default. And a cross-validation figure computed on a third of the training subsample estimates the variance of a different estimator on different data than the headline, which means the plus-or-minus in the conclusion attaches to nothing the conclusion claims.

REQUIRED REPAIR

Nested cross-validation: an inner loop selecting hyperparameters, an outer loop producing the generalization estimate. Report the outer-loop mean and spread as the headline.

9

MAJOR

DOCUMENTED

VERIFIED

SINGLE SEED FOR THE HEADLINE

One draw of the split, reported to six decimals

EVIDENCE

Cell twenty-four prints seed equals zero. Cell sixteen and cell twenty-two both pass `random_state` equals `RANDOM_STATE` to `train_test_split`. Seed variation does appear elsewhere in the audit — an independent re-run across eight seeds, which is not something your notebook prints, returned single-feature AUCs from nought point six nine one three to nought point seven zero one zero — but that probe varies the seed of the single-feature audit only. The reported model was fitted once.

SCIENTIFIC CONSEQUENCE

Six printed decimals on a quantity drawn once implies a precision the design cannot support, and the model ranking in cell sixteen is separated by less than the split-to-split spread you never measured.

REQUIRED REPAIR

Refit the headline across at least ten seeds and report mean plus or minus standard deviation. If the spread exceeds the gap between your top three models, say so in the ranking table.

DEFENSIBLE REPLY

The three-fold cross-validation standard deviation of plus or minus nought point zero zero one zero is already a stability statement, and on a corpus this saturated the split-to-split spread is plainly tiny.

10

CRITICAL

UNTESTED

VERIFIED

THE RIGHT DIAGNOSTIC, UNREAD

You drew the precision-recall curve and never read it

EVIDENCE

Cell eighteen calls `precision_recall_curve` on the held-out probabilities and plots recall against precision in the third panel. There is no call to `average_precision_score` anywhere in the notebook, and no precision-recall area appears in the ranking table, the validity ledger or the conclusion. Every ranking in this notebook is by ROC-AUC.

SCIENTIFIC CONSEQUENCE

I want to be exact here, because you have the figure in front of you and it is drawn correctly. The fault is not that the diagnostic is missing. The fault is that you plotted the right diagnostic and then declined to read it. At twelve percent positives, average precision is the quantity that responds to false-positive count; ROC-AUC is not. You produced the curve and then ranked four models on the metric that cannot see the problem.

REQUIRED REPAIR

Compute `average_precision_score` for every model, put it in the ranking table beside ROC-AUC, and rank on it.

Per-group recalls with no interval

EVIDENCE

Cell eighteen prints worst per-family recalls of nought point three nine eight for suspicious-only and nought point nine six seven for evil, and section thirteen carries both into the conclusion as the operationally important result. Neither carries an interval. There is no bootstrap in the notebook and no binomial interval on any per-class recall.

SCIENTIFIC CONSEQUENCE

The conclusion's third result — that the honest gap is the noisy heuristic label at nought point three nine eight — is an inferential claim about a difference between two recalls, and it is asserted without any statement of how precisely either was measured. A difference between two point estimates is not a finding until you know their spread.

REQUIRED REPAIR

Bootstrap the held-out AUC with at least a thousand resamples and report a ninety-five percent interval. Report a binomial interval on every per-group recall, with its positive count printed beside it.

DEFENSIBLE REPLY

0.398 against 0.967 on tens of thousands of held-out positives each is roughly a hundred and fifty standard errors apart; the interval you want would be about four thousandths wide and would confirm every sentence in the conclusion. The estimate that genuinely lacks a spread is the headline, and the reason is duplicate dependence, not a missing binomial.

ORDERED CORRECTION

Re-mechanise to the headline AUC and to the effective sample size: with roughly 7,600 distinct vectors, a row-level interval on 0.973639 would be wrong by construction, and that is the defensible complaint.

12

MAJOR MEASURED VERIFIED FALSE PRECISION

Six decimals on a single draw

EVIDENCE

Cell sixteen rounds accuracy and ROC-AUC to six decimals — nought point nine seven three six three nine for LightGBM against nought point nine seven three five one four for XGBoost. The ablation table in cell twenty-two likewise reports six decimals, and section thirteen quotes the six-decimal drop from nought point nine seven three six three nine to nought point nine seven three zero four six as though the difference of nought point zero zero zero five nine three were a measured effect.

SCIENTIFIC CONSEQUENCE

The sixth decimal encodes noise from one draw of one split. Quoting it in the conclusion invites the reader to treat a difference of six ten-thousandths as a result, when nothing in the design distinguishes it from re-split noise.

REQUIRED REPAIR

Report to the resolution the estimator supports — three decimals here — or attach the resampling spread that would justify more.

DEFENSIBLE REPLY

The code comment in cell sixteen states the six decimals are deliberate, so that a score of one point zero zero zero zero zero zero reads as a red flag rather than rounding to a clean one; and the conclusion does explicitly call small ablation differences re-split noise.

13

CRITICAL UNTESTED VERIFIED THRESHOLD FIXED AT 0.5, NO COST MODEL

The operating point was never chosen

EVIDENCE

Cell eighteen sets pred equal to pb greater than nought point five. The false-positive rate of nought point zero zero two two, the confusion matrix, and both per-group recalls all derive from that constant. No cost matrix appears anywhere in the notebook, and no threshold is selected from the precision-recall curve the same cell draws.

SCIENTIFIC CONSEQUENCE

In host intrusion detection the cost of a missed confirmed-evil event and the cost of an analyst chasing a benign syscall differ by orders of magnitude. When costs are that asymmetric, the operating point is the result — and one half is not an operating point, it is a library default. The recall of nought point three nine eight on suspicious-only that the conclusion calls the operationally important result is an artifact of that default.

REQUIRED REPAIR

State a cost matrix, choose the threshold that minimises expected cost, and report recall and false-positive rate at that threshold rather than at one half.

A winner declared by two ten-thousandths

EVIDENCE

Cell sixteen: LightGBM nought point nine seven three six three nine, XGBoost nought point nine seven three five one four, RandomForest nought point nine seven three three seven two. The gap across all three is nought point zero zero zero two six seven. On accuracy the top two are LightGBM at nought point nine five one two four five and RandomForest at nought point nine five one two two eight — a gap of seventeen millionths. There is no McNemar test, no paired comparison and no interval anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The declared winner then drives everything downstream: the ablation clones the winner, the cross-validation scores the winner, the conclusion names the winner. A selection made on a difference of two ten-thousandths, with no test and one seed, is a coin flip that the rest of the notebook treats as a finding.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top models, or overlapping bootstrap intervals. If they overlap, report a tie and say so.

DEFENSIBLE REPLY

The top three span 0.000267 of AUC; had any of them won, the ablation, the CV and the conclusion would read the same to three decimals. Naming the winner is a label, not a finding, and nothing in the paper rests on it.

ORDERED CORRECTION

Keep the demand for McNemar or overlapping bootstrap intervals, but state the consequence honestly: the tie changes no reported quantity, so this is a claim-strength defect rather than a result-changing one.

The base-rate paper is cited and never applied

EVIDENCE

Section two cites Chandola, Banerjee and Kumar, two thousand and nine, explicitly for framing the rare-malicious base-rate problem. Cell eighteen prints an operational false-positive rate at threshold one half of nought point zero zero two two, being one thousand eight hundred and sixty-two benign events flagged out of eight hundred and thirty-seven thousand eight hundred and fifty. That multiplication is never carried further. Let me carry it, on a stated assumption rather than a measurement from your notebook: take a modest fleet generating ten million kernel events in a day. Nought point zero zero two two times ten million is twenty-two thousand false alarms per day. At one alert per minute an analyst clears four hundred and eighty in an eight-hour shift, so twenty-two thousand divided by four hundred and eighty is about forty-six analyst-shifts, every day, to clear the false alarms alone.

SCIENTIFIC CONSEQUENCE

A false-positive rate of nought point two two percent sounds excellent and is a deployment-blocking number at kernel-event volume, because kernel events are generated in the millions per host per day. You cited the paper that says exactly this and then reported the rate without the multiplication. The reader is left with a reassuring decimal and no idea what it costs.

REQUIRED REPAIR

State an events-per-day volume, multiply, and report alerts per analyst per day. Then choose a threshold that fits an alert budget and report recall at that threshold.

Uncalibrated probabilities used as probabilities

EVIDENCE

Cell eighteen takes best dot predict_proba and compares it to nought point five. There is no Platt scaling, no isotonic regression, no reliability diagram and no Brier score anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A gradient-boosted tree's predict_proba output is a score, not a calibrated probability, and the number one half has no meaning against an uncalibrated score. Every operational figure in section nine — the confusion matrix, the false-positive rate, both per-group recalls — depends on that comparison being meaningful, and nothing in the notebook establishes that it is.

REQUIRED REPAIR

Calibrate on a held-out calibration fold with Platt or isotonic scaling, publish a reliability diagram and a Brier score, and only then threshold.

DEFENSIBLE REPLY

Calibration is a monotone transform. It leaves 0.973639, the ranking, the ablation and the CV exactly where they are. What it moves is the threshold, and you have already charged us for never choosing one.

ORDERED CORRECTION

Merge into fault 13, or restate as: the notebook reports operational figures at a probability cut whose scale is never established — which is the same defect as never selecting an operating point.

Trained on the heuristic, reported on the confirmed

EVIDENCE

Cell eleven sets y equal to df of sus , and builds family from $evil$. Section three describes sus as, quote, a broad, hand-assigned label for unusual activity at twelve percent of events, and $evil$ as the confirmed-malicious subset at about four percent. The printed rates are suspicious nought point one one nine seven and $evil$ nought point zero four four two. The headline AUC is against sus . The number the abstract leads with, ninety-six point seven percent, is recall against $evil$.

SCIENTIFIC CONSEQUENCE

The optimisation target and the advertised result are different quantities. You optimised a noisy heuristic and then reported performance on a clean subset of it, which is a favourable slice by construction — and the printed recall on the part of the target that is only suspicious is nought point three nine eight, meaning the model misses just over sixty percent of its own positive class. No estimate of the noise rate in sus is anywhere in the notebook, so the achievable ceiling on that target is unknown.

REQUIRED REPAIR

Either train directly on $evil$ and report that, or measure the disagreement between sus and $evil$ and publish it as a label-noise floor. State which target the headline belongs to in the abstract.

DEFENSIBLE REPLY

The suspicious flag is BETH's own published benchmark target, so modelling it is what makes the number comparable to other BETH work, and section three discloses the heuristic nature of the label plainly.

Atlas proof

docs/course/security-ml-notebooks/nb21-beth-host-telemetry.ipynb · cells 1, 4, and 11; zero-based manifest cell IDs: 1=48656f74, 4=31d51412, 11=6dc8975a

Rename the binary and the detector is gone

EVIDENCE

The five retained features, named in section one, are process name, event id, event name, argument count and return value. An independent re-run of the ablation — not printed in your notebook — shows the held-out AUC falling from nought point nine seven three nine to nought point eight two four five when process name alone is removed. There is no adversarial experiment, no perturbation budget and no evasion analysis anywhere in the notebook; the words adversarial and evasion do not appear in any markdown cell.

SCIENTIFIC CONSEQUENCE

Process name is a string the attacker chooses. Argument count and return value are properties of calls the attacker issues. Every column in this feature set is under adversary control at essentially zero cost, and the drop from nought point nine seven to nought point eight two on removing one of them tells you how much the detector leans on a field that can be changed by renaming a file. The worst-case performance of this detector is unmeasured, and worst case is the only case that matters in security.

REQUIRED REPAIR

Define a perturbation budget in attacker terms — process renaming, argument padding — apply it to the held-out positives, and report worst-case recall under it.

DEFENSIBLE REPLY

Section 13 scopes every number in the notebook to in-distribution separability and tells the reader not to treat it as a deployment estimate. You are asking for the worst case against a claim we did not make — and the number you cite for the processName drop appears nowhere in our output.

ORDERED CORRECTION

Rebuild on in-record evidence: the ablation drops returnValue, the column already graded A at single-feature AUC 0.6976, and never drops processName — the one attacker-chosen string in the matrix. Ablating the harmless column while leaving the adversary-controlled one untested is stronger than an unprinted 0.8245.

Your own caveat table says the number is incomparable

EVIDENCE

Section two states that the BETH authors' own baselines are unsupervised — an Isolation Forest and a variational-autoencoder method — and the caveat table in the same cell says, in your words, unsupervised and designed around the official host split; not directly comparable to our supervised in-split AUC. The official host split was not run.

SCIENTIFIC CONSEQUENCE

You have written the sentence that disqualifies your own number from the literature. A result that cannot be placed beside any prior result is not a contribution to a field; it is a measurement of one file under one protocol of your own devising. And the protocol that would have made it comparable was published with the data.

REQUIRED REPAIR

Run the official host-holdout split and report the supervised number under it, alongside the published unsupervised baselines. That is the only form in which this result enters the literature.

DEFENSIBLE REPLY

The sentence you read back to us says the authors' baselines are unsupervised and therefore not directly comparable to a supervised in-split AUC. Running their split would not fix that; it would produce a supervised number beside two anomaly detectors, which is still not a comparison. The honest position is the one we wrote down.

ORDERED CORRECTION

Merge into fault 5, or re-mechanise to the true residual: no supervised BETH result under any published protocol is cited for context, so the reader has no anchor at all.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here LightGBM. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 951,799 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So here is where this leaves us. This notebook is more honest than most of what gets published in this area — it measured the thing that kills it, printed the answer, and gave itself an F. And then it spent a paragraph arguing the F away using a two-thousand-row refit whose size it never disclosed, and shipped the headline. Six out of thirty. Reject with prejudice. The path back is not a better write-up; it is BETH's official cross-host split, which is published, which you named twice, and which you did not run.

Credential-Attack Detection on the RBA Login Dataset

The RBA corpus: a synthesized reconstruction of thirty-one point three million logins to a large online service, read here as a six point four eight million row slice with eight metadata columns, labelled by whether the source IP is on an attacker blocklist.

HEADLINE

XGBoost, held-out ROC-AUC of point nine zero nine — printed as nought point nine zero nine four one zero — with accuracy nought point eight five nine five four seven on one million six hundred and nineteen thousand held-out logins.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9.5

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 8 critical (FATAL)
- 15 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook has the best-written honesty apparatus of any in the set, and I want that on the record before I take it apart. It states in section one, in section three and again in section thirteen that the label is a property of the source IP and that the geographic and ASN columns re-encode that IP — it identifies its own central circularity three times, in plain language. It refuses to call the Kaggle release raw traffic. It decodes the two-letter country codes so a reader knows that S-M is San Marino. It warns that a figure title is stale rather than silently regenerating it. And it flagged the contamination figure as a sampled estimate; an independent re-run on a randomly chosen window returned nought point seven three five one against your printed nought point seven three seven, which vindicates the estimate. It also reads its own de-duplication result correctly, saying that removing duplicates makes the task easier rather than cleaner. That is a rare and admirable sentence. And every bit of it is disclosure. None of it is mitigation.

EXAMINER RULING

I am rejecting this notebook with prejudice, because a credential-attack detector that rejects one in every five legitimate logins is not a system with a tuning problem — it is a system that cannot be deployed at any threshold this notebook examined, and the notebook printed that number and moved on.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

One in five legitimate logins rejected

EVIDENCE

Cell eighteen prints, quote, operational false-positive rate at nought point five equals nought point two one one one, one hundred and seventy-eight thousand four hundred and forty benign flagged of eight hundred and forty-five thousand one hundred and sixty-nine. Eight hundred and forty-five thousand divided by one hundred and seventy-eight thousand is four point seven, so that is roughly one in every five legitimate logins. Now the arithmetic the notebook never does. I will use a stated assumption, not a measurement from your notebook: a service handling one million logins a day. Your own loader comment in cell eleven says attack IPs are about ten percent, so that is nine hundred thousand legitimate logins and one hundred thousand attack logins. Nought point two one one one times nine hundred thousand is one hundred and eighty-nine thousand nine hundred and ninety false alarms, every day. Now grant the detector something it does not have — perfect recall, catching all one hundred thousand attacks. Precision is then one hundred thousand over one hundred thousand plus one hundred and eighty-nine thousand nine hundred and ninety, which is one hundred thousand over two hundred and eighty-nine thousand nine hundred and ninety, which is thirty-four point five percent. And at one alert per minute, an analyst clears four

SCIENTIFIC CONSEQUENCE

Even in the best case this notebook's design permits — perfect recall, which it does not achieve — two out of every three alerts are wrong, and the false alarms alone consume nearly four hundred analyst-shifts a day. Risk-based authentication does not send these to an analyst; it sends them to a customer as a step-up multi-factor challenge. So the deployed meaning of nought point two one one one is that one in five honest customers is challenged on every login. That is not a detector with a threshold problem. That is a system no authentication team would ship, and the notebook reports the number as a line item in a ledger and then discusses geography.

REQUIRED REPAIR

State a login volume and a base rate, multiply, and report false alarms per day and precision at the deployed base rate. Then select a threshold against a step-up challenge budget — say, no more than one percent of legitimate logins challenged — and report recall at that threshold. If recall at an admissible false-positive rate is near zero, that is the finding.

hundred and eighty in an eight-hour shift, so one hundred and eighty-nine thousand nine hundred and ninety divided by four hundred and eighty is about three hundred and ninety-six analyst-shifts a day.

2

CRITICAL

DOCUMENTED

VERIFIED

I.I.D. SPLIT ON ENTITY-STRUCTURED DATA

ORAL TOP 5

Seventy-four percent of the holdout was in training

EVIDENCE

Cell twenty prints an exact-duplicate row rate for the whole corpus of nought point eight nine seven and train-test exact-row contamination of nought point seven three seven. Seventy-three point seven percent of held-out rows are exact copies of a training row. Apply that to the printed holdout: one million six hundred and nineteen thousand four hundred and fourteen times nought point seven three seven is one million one hundred and ninety-three thousand five hundred and eight rows already seen, leaving about four hundred and twenty-five thousand nine hundred that were genuinely new. And note what they are copies of — cell sixteen rebinds Xtr to the one-hundred-and-twenty-thousand-row subsample before cell twenty measures contamination, so nearly three quarters of a one-point-six-million-row test set matches a pool of one hundred and twenty thousand rows.

SCIENTIFIC CONSEQUENCE

With eight columns — one timing value and seven categorical fields — the corpus collapses to about six hundred and sixty-seven thousand distinct vectors out of six point four eight million rows. Six point four eight million times nought point one zero three gives that figure. A random split over a coarse categorical space is a lookup-table test, and nought point nine zero nine four one zero is substantially a measurement of that lookup. Your own trust grade says F for exactly this reason and your conclusion says treat it as a benchmark number, not a deployment estimate — and then the abstract and the ranking table present it as the result.

REQUIRED REPAIR

GroupShuffleSplit keyed on ASN, so that no autonomous system appears on both sides of the split. That is the split that answers the question your own conclusion says is the honest one.

Half the feature set is the IP address

EVIDENCE

Cell eleven's USE list gives the eight predictors: round-trip time, Country, Region, City, ASN, browser name and version, operating system name and version, and device type. Country, Region, City and ASN are all derived from the source IP by geolocation lookup — they are not observations of the user, they are re-expressions of the address. The label, set two lines later, is Is Attack IP, which is membership of that same address in a blocklist. And the single-feature audit in cell twenty confirms which one carries it: ASN alone reaches AUC nought point eight zero nine five against a headline of nought point nine zero nine four one zero.

SCIENTIFIC CONSEQUENCE

Half the feature matrix is a lossy hash of the key the label is looked up on. The model is not learning that a login is risky; it is learning which autonomous systems appear on the blocklist in this capture. That is a definition, not a discovery, and it is why ASN alone gets eighty-one percent of the way there. You state this circularity clearly in three sections — and then you report the score produced by it as the headline, and the friendly review awarded full marks on the criterion that asks whether the result is one column, because no single column reaches one point zero. The criterion asked the wrong question. Four columns jointly reconstruct the label's key.

REQUIRED REPAIR

Report the score with all four IP-derived columns removed — round-trip time, browser, operating system and device type only. That number is the behavioural signal. If it collapses to the baseline, the honest conclusion is that this corpus supports IP reputation and nothing else.

Six recalls printed out of one hundred and sixty-six groups

EVIDENCE

This one you can verify from the code alone, without trusting me. Cell eighteen contains if count less than five, continue — the floor. Then show equals pandas concat of s-r-t head nine and s-r-t tail three when there are more than twelve groups — so at most twelve bars are ever drawn. Then the print statement takes s-r-t head six — so exactly six recalls are printed, regardless of how many exist. Cell eighteen's printed line is worst per-family recalls, with six countries all at nought point zero. Now the corroboration, which is an independent re-run of your own cell and not something your notebook prints: there are one hundred and sixty-six country groups carrying held-out attack logins, and seventeen of them fall below the count-five floor and vanish. That leaves one hundred and forty-nine scoreable groups, of which twelve are plotted and six are printed. One hundred and thirty-seven scoreable groups are never shown at all.

SCIENTIFIC CONSEQUENCE

Section thirteen calls the per-country gap, in your words, the operationally important result — and it is computed from a display that shows four percent of the groups and prints under four percent of them. A reader cannot tell whether the six zeros are the tail of a smooth distribution or the visible edge of a systematic geographic blind spot, because one hundred and thirty-seven measured recalls were computed and thrown away before printing. And seventeen more countries have no measured recall at all, with no indication anywhere that they exist.

REQUIRED REPAIR

Print every group with its held-out positive count beside its recall. Report groups below the floor as unscoreable, by name and count — never omit them. A group with three positives is not a group with no information; it is a group with a wide interval, and the interval is the honest report.

You named the test, then used the key as a feature

EVIDENCE

Section thirteen says, in your words: the honest deployment question is cross-campaign transfer; a model that memorises which ASNs attacked in this capture will not flag a fresh campaign from clean infrastructure. ASN is column five of the eight in cell eleven's USE list. Cell twenty names it the strongest single feature at nought point eight zero nine five. Cell sixteen splits with `train_test_split` and `stratify equals y` — there is no groups argument anywhere in the notebook. Your section-two caveat table also states, of the Wiefeling scoring model, that I-P and A-S-N reputation carries much of it — that is the row exactly as printed, with I-P and A-S-N separated by a slash.

SCIENTIFIC CONSEQUENCE

You wrote the sentence that predicts the failure. You identified the exact column that causes it. That column was already loaded, already encoded, already ranked first in your own audit. Grouping the split by it is one line: `GroupShuffleSplit`, `groups equals the ASN column`. You did not write that line, and you published the ungrouped number. Naming the fatal limitation is not a defence at doctoral level — it is a confession that you knew which experiment mattered and shipped without it.

REQUIRED REPAIR

`GroupShuffleSplit` on ASN. Report the held-out AUC on autonomous systems never seen in training, and put that number in the abstract in place of nought point nine zero nine four one zero.

DEFENSIBLE REPLY

You have charged the ungrouped split at fault 2, the ASN circularity at fault 3, and now a third time here with the same one-line remedy. Naming our own limitation is an aggravator on those two, not a separate failure.

ORDERED CORRECTION

Fold the section 13 quotation into fault 2's why-fatal as the aggravating fact, and retire this as a standalone count.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You excluded the clock, and you knew it was clustered

EVIDENCE

Cell eleven's comment says, in your words, we drop the IP address, user ID and timestamp as direct leaks. The usecols list passed to pandas read_csv omits them, so the timestamp is never materialised. The same comment block says attack IPs are about ten percent and time-clustered — the notebook's own words.

SCIENTIFIC CONSEQUENCE

Credential attacks arrive as campaigns. Your loader comment says so. A campaign is a burst in time from a set of addresses, which means the single most important property of the data for evaluation purposes is its time ordering, and you excluded that column at read time. Dropping the timestamp as a feature is correct. Declining to load it at all means the temporal split — train on the first months, test on the last — was made impossible by a choice in the loader rather than by anything about the data.

REQUIRED REPAIR

Read the timestamp column, keep it out of the feature matrix, sort on it, train on the earlier period, test on the later period, and report the drop. That protocol is the closest thing this corpus offers to cross-campaign transfer.

DEFENSIBLE REPLY

You tell us at fault 4 that GroupShuffleSplit on ASN is the split that answers our question, and at fault 20 that this file is a synthesized reconstruction. A temporal split on a synthesized clock is neither the protocol you asked for nor a measurement of real campaign drift.

ORDERED CORRECTION

State which of the two splits is the primary demand and file the other as a secondary. As written the two FATAL charges prescribe different protocols for the same gap.

7

MAJOR

DOCUMENTED

VERIFIED

TRANSDUCTIVE PREPROCESSING

The encoder sees the ASNs you say you must not have seen

EVIDENCE

Cell eleven fits a LabelEncoder across every object column of X — that is Country, Region, City, ASN, browser, operating system and device type, seven of the eight predictors. The section-thirteen disclosed-limitation paragraph then says, quote, on an all-numeric corpus that step is a no-op. This corpus is not all-numeric; seven of eight columns are strings. The same paragraph says a deployed system would need an unseen-category bucket.

SCIENTIFIC CONSEQUENCE

The one preprocessing gap the notebook discloses is exactly the gap that defines its own stated deployment question. Your conclusion says the honest question is a fresh campaign from clean infrastructure — that is, from ASNs the model has never seen. Your pipeline has no defined behaviour for an unseen ASN, and the boilerplate sentence that would excuse the omission describes a corpus other than this one.

REQUIRED REPAIR

Fit the encoder inside the fold with an explicit unseen-category bucket, and correct the ledger sentence to describe this corpus.

DEFENSIBLE REPLY

No label information leaks, since the mapping never consults y; and fixing the encoder would not by itself answer the transfer question, which needs a grouped split, so this is a hygiene defect rather than the cause of the inflated score.

8

CRITICAL

UNTESTED

VERIFIED

NO MODEL SELECTION

Four defaults compared, no search run

EVIDENCE

Cell sixteen fixes RandomForest at sixty trees, XGBoost at eighty rounds and depth six, LightGBM at eighty rounds. There is no grid search, no randomised search, no tuning. Cell twenty-four sets cvX to Xtr dot iloc of colon forty thousand — the first forty thousand rows of the one-hundred-and-twenty-thousand-row training subsample — and the nought point nine zero four two plus or minus nought point zero zero zero seven that section thirteen reports as the winning learner's cross-validated ROC-AUC is that forty-thousand-row figure.

SCIENTIFIC CONSEQUENCE

Comparing four defaults is not comparing four methods. And a cross-validation number computed on a third of the training subsample cannot bound the error of the headline, which was fitted on a different sample and scored on a different one. The plus-or-minus in your conclusion attaches to an estimator you did not report.

REQUIRED REPAIR

Nested cross-validation — an inner loop selecting hyperparameters, an outer loop producing the estimate — and report the outer-loop spread as the headline uncertainty.

One split, six decimals

EVIDENCE

Cell twenty-four prints seed equals zero. Cell sixteen passes random_state equals RANDOM_STATE to a single train_test_split. Seed variation appears only in the single-feature audit: an independent eight-seed re-run, which your notebook does not print, gives single-feature AUCs from nought point eight zero eight three to nought point eight one three five. The headline model itself was fitted once.

SCIENTIFIC CONSEQUENCE

Your ranking table separates first place from third by under two thousandths of AUC, and you have no measurement of how much a different split would move that. Six printed decimals assert a stability that was never tested on the quantity being reported.

REQUIRED REPAIR

Refit across at least ten seeds and report mean plus or minus standard deviation for every model in the table.

DEFENSIBLE REPLY

At one point six million held-out rows the sampling error of a single AUC estimate is genuinely tiny, and the three-fold cross-validation standard deviation of nought point zero zero zero seven already indicates the estimate is stable across resamples.

Benign subsampled to twelve percent, then AUC reported

EVIDENCE

Cell eleven: neg equals the non-attack chunk dot sample with frac equals nought point one two. Every attack row is kept. The printed attack rate is therefore nought point four seven eight one, and the notebook correctly calls this a bounded-sample rate rather than a base rate. But one divided by nought point one two is eight point three three, so benign logins are under-represented by a factor of about eight relative to the roughly ten percent attack rate your own loader comment names. Meanwhile cell eighteen calls precision_recall_curve and plots it, and average_precision_score appears nowhere in the notebook.

SCIENTIFIC CONSEQUENCE

Accuracy and ROC-AUC on a forty-eight percent positive sample say almost nothing about a ten percent positive deployment, because both are computed on class-conditional rates that ignore the eight-fold change in how many benign logins arrive per attack. The metric that does respond to that change is average precision, and you drew its curve and never computed its area. So you plotted the right diagnostic and then declined to read it — on the one notebook in this set where the base rate was reshaped by design.

REQUIRED REPAIR

Compute average precision, and compute it on a test set reweighted to the natural base rate. Report precision at a fixed operational recall on that reweighted set.

DEFENSIBLE REPLY

Subsampling benign uniformly leaves ROC-AUC unchanged in expectation — it is a rank statistic. Our 0.909410 is not inflated by the 12% draw. If you want the metric that moves with the prior, ask for average precision, and say so.

ORDERED CORRECTION

Delete 'Accuracy and ROC-AUC on a forty-eight percent positive sample say almost nothing about a ten percent positive deployment' and replace with: accuracy and the majority baseline are reshaped by the 12% benign draw; ROC-AUC is not; average precision is the prior-sensitive metric and it is drawn but never computed.

11

CRITICAL UNTESTED VERIFIED NO UNCERTAINTY ANYWHERE

Six recalls of exactly zero, no counts attached

EVIDENCE

Cell eighteen prints worst per-family recalls of San Marino, Cambodia, Honduras, Greece, Jamaica and Jordan, all at nought point zero. Section nine decodes the codes, which is good practice. Neither the print nor the plot carries the number of held-out attack logins in each country. Because the count-five floor let them through, each has at least five — but five and five hundred give the same printed zero, and there is no bootstrap or binomial interval anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A recall of zero on five positives and a recall of zero on five thousand positives are different findings — the first is a wide interval consistent with anything, the second is a demonstrated blind spot. Section thirteen calls this the operationally important result. It is not usable as a result without its denominators.

REQUIRED REPAIR

Print the positive count beside every per-group recall and attach a binomial confidence interval. Bootstrap the headline AUC and report a ninety-five percent interval.

12

CRITICAL UNTESTED VERIFIED THRESHOLD FIXED AT 0.5, NO COST MODEL

The worst number in the notebook is a library default

EVIDENCE

Cell eighteen sets pred equal to probability greater than nought point five. That constant produces the nought point two one one one, the confusion matrix and every per-country recall. There is no cost matrix in the notebook and no threshold is selected from the precision-recall curve the same cell draws.

SCIENTIFIC CONSEQUENCE

In risk-based authentication the costs are wildly asymmetric and well understood — a missed credential-stuffing login is an account takeover, a false positive is a friction event for a paying customer. When costs are asymmetric the operating point is the result. Reporting a catastrophic false-positive rate at a default and never sweeping the threshold means the notebook does not know whether an admissible operating point exists on this model at all. That is the question the paper should have answered.

REQUIRED REPAIR

State a cost matrix — challenge cost versus takeover cost — sweep the threshold, and report the expected-cost-minimising operating point with its recall and false-positive rate. Alternatively fix a challenge budget of one percent and report recall there.

The runner-up beats the winner on accuracy

EVIDENCE

Cell sixteen: XGBoost is ranked first with accuracy nought point eight five nine five four seven and ROC-AUC nought point nine zero nine four one zero.

RandomForest is ranked third with ROC-AUC nought point nine zero seven five five eight — but accuracy nought point eight five nine five seven nine, which is higher than the winner's. The three tree models span nought point zero zero one eight five two of AUC. There is no McNemar test, no paired comparison and no interval in the notebook.

SCIENTIFIC CONSEQUENCE

Two printed columns disagree about which model is best, and the notebook resolves the disagreement by an unstated convention rather than by a test. Everything downstream inherits that choice: the ablation clones the winner, the cross-validation scores the winner, the operational false-positive rate is the winner's. A selection made on under two thousandths with one seed and no significance test is not a result.

REQUIRED REPAIR

McNemar's test on paired held-out predictions across the three tree models. If the intervals overlap, declare a tie and report the ablation for all three.

In risk-based authentication the score is the product

EVIDENCE

Cell eighteen calls predict_proba and compares to nought point five. No Platt scaling, no isotonic regression, no reliability diagram and no Brier score appear anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Risk-based authentication does not consume a binary decision. It consumes a risk score and maps bands of that score to actions — allow, challenge, block. That mapping is meaningless unless the score is calibrated, because a band boundary at nought point seven has to mean something about probability. The notebook produces exactly the artifact the application needs and never establishes that it means anything, then thresholds it at a default and reports the consequences as an operational number.

REQUIRED REPAIR

Calibrate on a held-out calibration fold, publish a reliability diagram and a Brier score, and define the risk bands on the calibrated scale.

DEFENSIBLE REPLY

Calibration is monotone. It moves no number we report except the ones at the 0.5 cut, and you have already charged the cut. We publish no risk bands, so there is no banded mapping to invalidate.

ORDERED CORRECTION

Merge into fault 12, keeping the RBA-specific sentence about band boundaries as the reason the operating point matters more here than elsewhere.

Seventy-four percent is the floor, not the leakage

EVIDENCE

Cell twenty computes `dup_rate` as one minus `X dot drop_duplicates` over the length of `X`, and contamination as membership of rounded test rows in the set of rounded training rows. Both are exact-match tests. Seven of the eight columns are categorical, so the only column that can differ between two otherwise-identical logins is round-trip time in milliseconds — a continuous value that differs on essentially every login. No nearest-neighbour distance, no near-duplicate analysis and no distance-to-training-set distribution appears in the notebook.

SCIENTIFIC CONSEQUENCE

The single continuous column is the only thing that can prevent two logins from the same address, city and browser from registering as an exact duplicate. That means nought point seven three seven is a lower bound on the leakage, and the true figure is whatever fraction of the test set has a near-identical neighbour in training. My expectation, which neither of us has measured, is that on a space this coarse it approaches all of it — but the point that stands without any expectation is that your F grade was assigned on the optimistic measurement, and the pessimistic one was never taken.

REQUIRED REPAIR

Compute the nearest-neighbour distance from every held-out row to the training set and plot the distribution. Report the fraction within a stated tolerance. That number, not the exact-match rate, is the leakage.

DEFENSIBLE REPLY

Our exact-match rate is 0.897 — nearly nine rows in ten are byte-identical including round-trip time, so that column is not the barrier you describe. And a tighter measurement can only take contamination from 0.737 up. The grade is F either way and the conclusion already says treat it as a benchmark number.

ORDERED CORRECTION

Drop 'a continuous value that differs on essentially every login' — the printed 0.897 refutes it. Keep the nearest-neighbour request as a methodological improvement rather than as a charge that the F grade was optimistic.

The non-circular label is in your dataframe

EVIDENCE

Cell eleven's USE list includes Is Account Takeover, and the DROP list four lines later removes it. Section three says of it, in your words: it is never evaluated in this notebook, and — crucially — that second label is not IP-derived, so the circularity described above does not apply to it. The section then names it an untested hypothesis whether it is learnable from these eight features.

SCIENTIFIC CONSEQUENCE

You loaded the one label in this file that escapes your central validity problem, wrote down that it escapes it, and then dropped it. The whole notebook is organised around the finding that the target is a definition rather than a behaviour. The remedy was in the same dataframe, four lines up. Swapping one variable would have converted a circular experiment into a real one.

REQUIRED REPAIR

Model Is Account Takeover as the target, report its base rate, and report performance on it. If it is not learnable from eight metadata columns, that negative result is more scientifically valuable than the circular positive one.

DEFENSIBLE REPLY

Account takeover is a far rarer and different phenomenon, its base rate may leave too few positives for a stable estimate, and Is Attack IP is the label the source study itself uses, so modelling it keeps the work comparable to Wiefeling et al.

The evasion costs five dollars a month

EVIDENCE

The strongest single feature is ASN at nought point eight zero nine five, printed in cell twenty. An independent re-run of the ablation, which your notebook does not print, shows the held-out AUC falling from nought point nine one four zero to nought point eight nine one one when ASN is removed and to nought point eight six five seven when Country is removed as well. The words adversarial and evasion appear in no markdown cell of this notebook, and there is no perturbation experiment of any kind.

SCIENTIFIC CONSEQUENCE

The attacker chooses the ASN by choosing a hosting provider, and chooses the browser and operating system strings by setting a header. Six of your eight features are under direct adversary control at trivial cost, and the seventh — round-trip time — is controlled by choosing where to rent. A detector whose signal is entirely attacker-selectable has a worst-case performance near the baseline, and worst case is the only case that matters when the opponent is adaptive. That number is unmeasured here.

REQUIRED REPAIR

Define a budget in attacker terms — the attacker may relocate to any ASN not present in the training data and set arbitrary client strings — apply it to the held-out attack logs, and report worst-case recall.

DEFENSIBLE REPLY

Section 13 tells the reader in terms not to treat any of this as a deployment estimate. And the ablation figures you quote are not ours — cell 22 prints 0.909410 and 0.888466 for the ASN drop, and nothing else.

ORDERED CORRECTION

Rebuild on the printed ablation: dropping ASN costs 0.909410 -> 0.888466, which is the notebook's own measurement of how much one rentable column carries, and it is in the record.

Your own textbook says this test overstates stability

EVIDENCE

Cell zero, under where this connects to the course text, says of Chapter four learning objective five that it separates a one-at-a-time perturbation from the smallest perturbation that reverses a ranking, and warns the first overstates stability — and then says, in your words, dropping only the top feature and refitting is exactly that weaker test, so read it as a floor. The ablation in cell twenty-two drops exactly one feature: ASN, giving nought point nine zero nine four one zero down to nought point eight eight eight four six six. No experiment anywhere searches for the smallest set of features whose removal reverses the conclusion.

SCIENTIFIC CONSEQUENCE

You cited the standard, correctly described why the test you ran falls short of it, and then ran only that test. The friendly review awarded a full point on the criterion asking whether the result is one column, resting entirely on this floor. A floor is not a bound. The stronger experiment — the minimal removal that reverses the finding — is the one that would have shown four IP-derived columns jointly carrying the label.

REQUIRED REPAIR

Search for the minimal feature subset whose removal drops the AUC to the majority baseline, and report that subset. On this corpus I expect it to be the four geolocation columns.

DEFENSIBLE REPLY

The notebook explicitly labels the one-at-a-time result a floor rather than a bound, and section eleven pre-commits to reading a non-collapse as a generation artifact rather than as vindication — so it claims no more than the test supports.

Two published models cited, neither run

EVIDENCE

Section two cites Wiefling and colleagues, two thousand and twenty-two, as the source study and its risk-based-authentication scoring model, and Freeman and colleagues, two thousand and sixteen, as the statistical authenticity model. The caveat table in the same cell says of the first that I-P and A-S-N reputation carries much of it, and of the second, in your words, features overlap with the labelling signal. Neither model is implemented, and no published protocol is reproduced anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The paper that produced this dataset also published a scoring model on it. Reporting a new number without reproducing that baseline leaves the reader unable to say whether nought point nine zero nine four one zero is good, bad or identical to what was already known. And you wrote in your own caveat table that the prior model is carried by ASN reputation — which is a testable claim about your own top feature that you did not test.

REQUIRED REPAIR

Implement the Freeman-style authenticity baseline on this feature set and report both numbers side by side under the same split.

DEFENSIBLE REPLY

We did test that claim. Cell 20 prints ASN alone at 0.8095 and the ledger names it. And Wiefling's scoring numbers are reported on production traffic, not on the synthesized public release we are measuring, so there is no like-for-like figure to sit beside ours.

ORDERED CORRECTION

Remove 'which is a testable claim about your own top feature that you did not test' — cell 20 is that test. The surviving residual is that no baseline of any kind is implemented for context.

This is not production traffic and you said so

EVIDENCE

Cell eleven's comment says, in your words: the public Kaggle release is a synthesized version of the original production traffic, re-generated so customers cannot be re-identified, so it mirrors real behaviour but is not itself the raw log. The abstract two cells earlier opens with: the RBA dataset is thirty-one point three million login attempts to a large online service.

SCIENTIFIC CONSEQUENCE

The joint distribution of country, city, ASN, browser and round-trip time in a privacy-preserving synthesis is a product of the generator that produced it. A classifier that separates classes at nought point nine zero nine on such a corpus has measured how separable the generator made them. That is a statement about a synthesis pipeline, not about credential attacks — and the duplicate rate of nought point eight nine seven is precisely the fingerprint of a generator drawing from a finite pool of category combinations. The honest caveat is in a code comment; the abstract, which is what people read, describes it as real traffic.

REQUIRED REPAIR

Move the synthesis disclosure from a code comment into the first sentence of the abstract, and restrict every claim to the synthesized release. Any claim about production authentication traffic requires production traffic.

DEFENSIBLE REPLY

Read the line directly above the abstract. It says 'RBA synthesized release'. Section 2 says it again in full. Your own concession paragraph credits us for refusing to call this raw traffic. The caveat is not buried in a comment.

ORDERED CORRECTION

Replace the 'buried in a code comment' mechanism with the true residual: the disclosure is present in three places and no reported number is qualified by it — the ledger reports 0.909410 and 0.897 without stating that both are properties of a generator, and the abstract's first sentence describes the corpus rather than the release.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 1,619,414 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

De-duplication removed most of my corpus here, so the movement you see is dominated by a real effect, not by split noise. The criticism still applies — I never measured the nuisance term — but it is at its weakest on this notebook, not its strongest.

Closing. Seven out of thirty. Reject with prejudice. And I want to be precise about why this one is worse than a badly written notebook, not better: this is the most articulate honesty apparatus in the entire set, and it is deployed in service of a number that its own printed output disqualifies three separate ways — seventy-four percent contamination, a label that is a lookup on a key that four of its eight features reconstruct, and a false-positive rate that would challenge one in five honest customers. You wrote down every one of those. Writing them down is what a scientist does first. Running the grouped split is what a scientist does second, and it is one line, and it is still not written.

Cross-Dataset Intrusion Detection on NF-UQ-NIDS-v2 (Merged NetFlow Superset)

NF-UQ-NIDS-v2: four separate intrusion corpora — Bot-IoT, ToN-IoT, UNSW-NB15 and CSE-CIC-IDS2018 — re-expressed on one NetFlow schema and concatenated into a sixty-two point seven million flow superset, sampled here to four million rows.

HEADLINE

XGBoost, held-out ROC-AUC of point nine nine nine — printed as nought point nine nine eight seven two one — with accuracy nought point nine eight seven three one five on a one million flow holdout, offered as evidence that one model can handle the heterogeneous union.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 9 critical (FATAL)
- 13 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

The data hygiene here is genuinely the best in the set, and I will not pretend otherwise. Contamination is nought point zero zero zero and the duplicate rate is nought point zero zero zero — an independent re-run on a randomly chosen window confirms both — so unlike most of this series, the held-out score is a real held-out score. The notebook identified the capture-identity column by name, dropped it before any model saw the data, and said in section one exactly why. It reconciles the canonical seventy-six million flow merge against the sixty-two point seven million cleaned asset instead of quietly using the larger figure. And section eleven pre-commits, before the numbers land, to reading a non-collapsing ablation as a generation artifact rather than as vindication — which is exactly the right epistemic posture and almost nobody writes it down. All of that is real. None of it is the experiment.

EXAMINER RULING

I am rejecting this notebook because it asks whether one model can handle a heterogeneous union of four captures, and then runs the one experiment — a pooled random split — that structurally cannot answer that question, while the answer sits printed in five sibling notebooks of this same series.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Dropping the column removed the name, not the information

EVIDENCE

Here is the composition, and every number is printed in this series. Notebook eleven prints thirty million four hundred and twenty thousand and eighty-six rows for Bot-IoT at a positive rate of nought point nine nine five seven. Notebook twelve prints thirteen million one hundred and thirty-five thousand eight hundred and eighty-one rows for ToN-IoT at nought point seven two five eight. Notebook thirteen prints one million nine hundred and eighty-six thousand seven hundred and forty-five rows for UNSW-NB15 at nought point zero three seven eight. Notebook twenty prints seventeen million one hundred and twenty-nine thousand seven hundred and fifteen rows for CSE-CIC-IDS2018 at nought point one one eight four. Those four row counts sum to sixty-two million six hundred and seventy-two thousand four hundred and twenty-seven — which is the sixty-two point seven million your own loader names. And the volume-weighted average of those four attack rates is nought point six six eight nine seven nine, which rounds to nought point six six nine zero — the exact pooled attack rate your cell eleven prints. The composition is fully determined, and nearly half the corpus, forty-eight point five percent, is one capture in which ninety-nine point six percent of flows are attacks.

SCIENTIFIC CONSEQUENCE

Knowing which capture a flow came from moves the prior on its label from three point eight percent to ninety-nine point six percent. That is an enormous amount of label information, and it does not live in the Dataset column — it lives in the joint distribution of duration, throughput, packet-size and TCP-flag fields, because four different testbeds with four different hardware configurations produce four distinguishable NetFlow fingerprints. A gradient-boosted tree with forty-one columns will find that fingerprint. So the model can reach nought point nine nine eight seven two one by first inferring which testbed it is looking at and then applying that testbed's base rate. Your section one says you dropped the column, quote, so the model cannot cheat by identifying the capture instead of the behaviour. Dropping the name does not stop the inference. Nothing in this notebook measures whether the inference happens.

REQUIRED REPAIR

Fit a classifier whose target is the Dataset column itself, on the same forty-one features, and report its accuracy. If a model can recover which corpus a flow came from — and it will — then corpus identity is available to the label model too, and the headline is confounded by a quantity you have proven is recoverable. That experiment is ten lines and you already have the column.

The answer is already printed at notebook thirty-six

EVIDENCE

Notebook thirty-six in this series trains one model per corpus and tests it on all four, on the NetFlow v2 schema. Its section-thirteen ledger prints: in-distribution diagonal, mean nought point nine nine six six, minimum nought point nine eight eight zero. Cross-corpus off-diagonal, mean nought point five four three four, minimum nought point three three one five. Worse-than-random cells: five of twelve. The worst cell is train on Bot-IoT, test on ToN-IoT, at nought point three three one five. The best off-diagonal cell is train on UNSW-NB15, test on Bot-IoT, at nought point seven eight seven nine — and that best cell still sits below every entry on the diagonal.

SCIENTIFIC CONSEQUENCE

Your abstract asks: can one model handle the heterogeneous union? Notebook thirty-six measured it. On your four corpora, with your feature set, transfer averages nought point five four three four — a coin flip — and in five of twelve directions it is worse than a coin flip, which notebook thirty-six correctly describes as the learned rule being actively inverted on the new corpus. Your headline is nought point nine nine eight seven two one. The gap between those two numbers is the entire finding of this notebook, and this notebook does not contain it. Your section thirteen mentions notebook thirty-six by pointing the reader at the transfer matrix. Pointing at a result is not reporting it, and it is certainly not running it on the pooled model you actually fitted.

REQUIRED REPAIR

Report the off-diagonal figures in this notebook's own ledger, beside nought point nine nine eight seven two one, in the abstract. Then fit the pooled model and test it on a fifth corpus held out entirely — that is the pooled-model version of notebook thirty-six's experiment and it does not exist anywhere in this series.

Pooling made every shared attack family worse

EVIDENCE

Compare cell eighteen's printed per-family recalls against the same families printed by the sibling notebooks. Infiltration: notebook twenty, on CSE-CIC-IDS2018 alone, prints nought point two seven six; this notebook prints nought point one six eight. Theft: notebook eleven, on Bot-IoT alone, prints nought point nine eight two; this notebook prints nought point six six seven. Man-in-the-middle: notebook twelve, on ToN-IoT alone, prints nought point nine three four; this notebook prints nought point seven zero three. Ransomware: notebook twelve prints nought point nine zero four; this notebook prints nought point eight three seven. Reconnaissance: notebook eleven prints one point zero; this notebook prints nought point nine two one. Five families, five declines. Meanwhile the pooled AUC of nought point nine nine eight seven two one does not exceed notebook eleven's nought point nine nine nine nine zero two or notebook thirteen's nought point nine nine nine five zero two.

SCIENTIFIC CONSEQUENCE

Let me be careful about what these five comparisons do and do not establish, because a sharp student will press on it and should. They are not a controlled experiment. The sibling notebooks fit different winning learners, the corpora carry different positive rates, and every recall is taken at the same arbitrary threshold of one half. So I am not asserting that pooling caused the declines. What I am asserting is this: the comparison that would answer your abstract's question runs in one direction in five families out of five, it is assembled entirely from printed outputs of notebooks you cite, and you did not make it. Five out of five in one direction is precisely the pattern that obliges an author to run the comparison properly rather than to leave it unmade. And note what your headline cannot do: forty-eight point five percent of the corpus is a capture that is ninety-nine point six percent trivially separable floods, so the aggregate AUC is dominated by them and is structurally incapable of registering any of this. The single number you put in the headline is the one number in the notebook that cannot see the question.

REQUIRED REPAIR

Print the per-family recall of the pooled model beside the per-family recall of the corresponding single-corpus model, as a table — and this time control it: same learner, same threshold, same held-out proportion, so the comparison means something. Lead the conclusion with it. That comparison is the paper.

The group key was in your hand and you deleted it

EVIDENCE

Cell eleven's DROP list contains Label, Attack, Dataset, y and family. The Dataset column names which of the four source captures each flow came from — your section one says so explicitly. Cell sixteen then calls `train_test_split` with `test_size` nought point two five and `stratify` equals y. There is no groups argument anywhere in the notebook and no `GroupKFold`, `GroupShuffleSplit` or `leave-one-group-out` anywhere in the code.

SCIENTIFIC CONSEQUENCE

Leave-one-corpus-out is the canonical protocol for a merged benchmark, it is the reason NF-UQ-NIDS was constructed, and the grouping column was sitting in the dataframe on the line above the split. Instead the split places flows from all four captures on both sides, which guarantees the model is never asked about a network it has not seen. Every number downstream — the AUC, the accuracy, the false-positive rate, the per-family recalls — is measured under a protocol that structurally cannot fail in the way deployment fails.

REQUIRED REPAIR

`GroupKFold` with groups equal to the Dataset column, four folds, one per corpus. Report all four held-out scores. That is one line of change from what you wrote, and it is the experiment the dataset was built to support.

You cited the theory that predicts your failure

EVIDENCE

Cell zero, under where this connects to the course text, says of Chapter eight learning objective two that it derives why non-I-I-D site distributions produce client drift, then states, in your words: a model fit on one site does not carry to another. This notebook does not run that test: every score here is in-distribution on one corpus. Notebook thirty-six runs the cross-corpus transfer matrix. Section thirteen repeats it: treat cross-environment transfer as the open question; a single pooled split cannot measure that transfer.

SCIENTIFIC CONSEQUENCE

You have written down the theory that predicts the failure, the name of the test that would detect it, and the location of the notebook that already ran it. Three separate pointers to the experiment, and the experiment is absent. At doctoral level that sequence is not a limitations paragraph — it is a record that the author knew precisely which experiment settled the claim and published without it. The disclosure paragraph cost nothing and bought nothing; the grouped split costs one line.

REQUIRED REPAIR

Run the leave-one-corpus-out evaluation in this notebook and lead the abstract with its worst fold. Keep the pooled number as a diagnostic if you like, but it cannot be the headline of a paper about heterogeneous unions.

DEFENSIBLE REPLY

Faults 2, 4 and 5 all end in the same sentence: run the cross-corpus evaluation. Charging the split, the comparator and the disclosure separately is one experiment counted three times.

ORDERED CORRECTION

Fold the cell 0 quotation into fault 4's why-fatal as the aggravating fact and retire this as a standalone count.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Sixty-two point seven million advertised, one hundred and twenty thousand fitted

EVIDENCE

Cell eleven prints, quote, loaded four million flows, a four-million stratified sample of sixty-two point seven million. Cell sixteen then prints materialized four million rows, trained on one hundred and twenty thousand stratified subsample, held-out one million. One hundred and twenty thousand divided by sixty-two point seven million is nought point one nine one percent — under two rows in every thousand of the corpus the title advertises.

SCIENTIFIC CONSEQUENCE

The scientific interest of a merged superset is that it is large and heterogeneous. A one-hundred-and-twenty-thousand-row stratified draw from it is neither: stratification is on the binary label only, so there is no guarantee the four captures are represented in the training subsample in any particular proportion, and no printed check that they are. The claim in the abstract is about the union; the evidence is about a draw whose composition was never inspected.

REQUIRED REPAIR

Print the per-corpus composition of the training subsample and of the holdout. Then run a learning curve on distinct flows until the score saturates, and state the effective sample size in the abstract instead of the download size.

DEFENSIBLE REPLY

The subsample is printed in the loader line, printed again in the model line, stated in the title and stated in section 8. And stratifying on the label still draws uniformly within each stratum, so at $n = 4,000,000$ the four captures are represented to within a rounding error of their true shares.

ORDERED CORRECTION

Drop 'there is no guarantee the four captures are represented in the training subsample in any particular proportion' — at this sample size the sampling error on those proportions is negligible. The surviving and legitimate demand is the printed composition check and a learning curve.

8

CRITICAL

UNTESTED

VERIFIED

NO MODEL SELECTION

Defaults compared, and the cross-validation is on a slice of train

EVIDENCE

Cell sixteen fixes RandomForest at sixty trees, XGBoost at eighty rounds and depth six, LightGBM at eighty rounds — no grid, no randomised search, no tuning. Cell twenty-four sets cvX to Xtr dot iloc of colon forty thousand, the first forty thousand rows of the one-hundred-and-twenty-thousand-row training subsample, and the nought point nine nine eight one plus or minus nought point zero zero zero two your conclusion reports is that forty-thousand-row number.

SCIENTIFIC CONSEQUENCE

The comparison ranks four hand-set configurations, so the declared winner is a property of the packages rather than of the methods. And the plus-or-minus quoted in the conclusion belongs to an estimator fitted on different data than the headline and scored on different data than the headline, so it bounds nothing about the reported number.

REQUIRED REPAIR

Nested cross-validation: an inner loop for selection, an outer loop for the estimate, with the outer spread reported as the headline uncertainty.

9

MAJOR

DOCUMENTED

VERIFIED

SINGLE SEED FOR THE HEADLINE

One draw of the split behind six decimals

EVIDENCE

Cell twenty-four prints seed equals zero and cell sixteen passes random_state equals RANDOM_STATE to a single train_test_split. Seed variation appears only in the single-feature audit — an independent eight-seed re-run, which your notebook does not print, gives single-feature AUCs from nought point eight seven two three to nought point eight seven five seven. The headline model was fitted once.

SCIENTIFIC CONSEQUENCE

The ablation table reports differences of five millionths and twenty millionths between settings and the conclusion characterises them as re-split noise — correctly — but that characterisation is an assertion, because the re-split noise was never measured on the headline.

REQUIRED REPAIR

Refit across at least ten seeds and report mean plus or minus standard deviation, so that the phrase re-split noise is backed by a measured spread.

DEFENSIBLE REPLY

On a one-million-row holdout the sampling error of a single AUC is genuinely negligible, and the conclusion already labels sub-millionth differences as noise rather than treating them as effects.

10

CRITICAL

UNTESTED

VERIFIED

NO UNCERTAINTY ANYWHERE

Six per-family recalls, no denominators

EVIDENCE

Cell eighteen prints Infiltration nought point one six eight, Theft nought point six six seven, man-in-the-middle nought point seven zero three, ransomware nought point eight three seven, Reconnaissance nought point nine two one and Shellcode nought point nine two six. Not one has a positive count printed beside it. The count-five floor at cell eighteen means each has at least five, and an independent re-run confirms twenty groups with none dropped, but that is the only bound available. There is no bootstrap and no binomial interval anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Section thirteen states that the weakest group, at nought point one six eight, is the operationally important result. Whether that is a demonstrated blind spot or an estimate from a handful of flows is not determinable from anything printed, and the two readings lead to opposite operational decisions.

REQUIRED REPAIR

Print each family's held-out positive count beside its recall, attach a binomial interval, and bootstrap the headline AUC for a ninety-five percent interval.

11

MAJOR

MEASURED

VERIFIED

FALSE PRECISION

Ablation deltas quoted at five millionths

EVIDENCE

The cell twenty-two ablation table gives nought point nine nine eight seven two one for the headline, nought point nine nine eight seven one six after deduplication and nought point nine nine eight seven zero one after dropping the top feature. Section thirteen quotes the difference of nought point zero zero zero zero five and the drop of two hundredths of a thousandth as though each were a quantity.

SCIENTIFIC CONSEQUENCE

Differencing sixth decimals invites the reader to treat quantities smaller than any measured error as effects. The conclusion does say these are re-split noise, which is the right reading — but it prints them at a resolution that undercuts its own caution.

REQUIRED REPAIR

Report three decimals, or report six with an accompanying resampling spread that justifies them.

DEFENSIBLE REPLY

The cell sixteen code comment states that six decimals are deliberate so a score of exactly one point zero zero zero zero zero zero cannot hide behind rounding, and the conclusion explicitly names the small differences as noise rather than as findings.

No operating point was chosen

EVIDENCE

Cell eighteen sets pred equal to probability greater than nought point five, and that constant produces the false-positive rate of nought point zero two zero one, the confusion matrix and all six per-family recalls. No cost matrix exists in the notebook and no threshold is chosen from the precision-recall curve the same cell plots.

SCIENTIFIC CONSEQUENCE

In intrusion detection the cost of a missed infiltration and the cost of an analyst chasing a benign flow differ by orders of magnitude, and when costs are that asymmetric the operating point is the result rather than a detail. The nought point one six eight recall on infiltration that the conclusion nominates as the important finding is a property of the default threshold, not of the model.

REQUIRED REPAIR

State a cost matrix, sweep the threshold, and report the expected-cost-minimising operating point with its recall and false-positive rate — or fix an alert budget and report recall at that budget.

Atlas proof

docs/course/security-ml-notebooks/nb23-nf-ug-nids-v2.ipynb · cells 16 and 18; zero-based manifest cell IDs: 16=394c9faf, 18=f2a01c0e

The third-place model has the best accuracy

EVIDENCE

Cell sixteen: XGBoost ranked first, accuracy nought point nine eight seven three one five, ROC-AUC nought point nine nine eight seven two one. RandomForest ranked third, ROC-AUC nought point nine nine eight one five seven — but accuracy nought point nine eight eight four nine two, higher than the winner's. The three tree models span nought point zero zero zero five six four of AUC. There is no McNemar test, no paired comparison and no interval.

SCIENTIFIC CONSEQUENCE

Two printed columns disagree about the best model and the notebook resolves the disagreement by convention. The choice then propagates: the ablation clones the winner, the cross-validation scores the winner, the false-positive rate and the per-family recalls are the winner's. A selection made on five ten-thousandths with one seed and no test is not a finding, and here it changes which model's per-family recalls the conclusion reports.

REQUIRED REPAIR

McNemar's test on paired held-out predictions across the three tree models, or overlapping bootstrap intervals. Declare a tie if they overlap.

Two percent of a NetFlow feed is two hundred thousand alerts

EVIDENCE

Cell eighteen prints an operational false-positive rate at nought point five of nought point zero two zero one, being six thousand six hundred and forty-one benign flows flagged of three hundred and thirty thousand nine hundred and ninety. Note first that benign flows are only about a third of the holdout, because the pooled attack rate is nought point six six nine zero. Now the arithmetic, on a stated assumption rather than a measurement from your notebook: a mid-sized enterprise sensor seeing ten million benign flows a day. Nought point zero two zero one times ten million is two hundred and one thousand alerts a day. At one alert a minute an analyst clears four hundred and eighty per eight-hour shift, so two hundred and one thousand divided by four hundred and eighty is about four hundred and nineteen analyst-shifts a day.

SCIENTIFIC CONSEQUENCE

Two percent reads as excellent and is unshippable at flow volume, and the notebook's own benign population is three times smaller relative to attacks than any real network's, so nought point zero two zero one is measured under conditions kinder than deployment in both directions. No multiplication appears anywhere, and the notebook cites Sommer and Paxson — the paper whose central argument is precisely this — in section two.

REQUIRED REPAIR

State a flows-per-day volume, multiply, report alerts per analyst per day, and select a threshold against an alert budget. Report recall at that threshold, including infiltration recall.

Uncalibrated scores thresholded as probabilities

EVIDENCE

Cell eighteen calls `predict_proba` and compares to nought point five. There is no Platt scaling, no isotonic regression, no reliability diagram and no Brier score anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

A boosted tree's probability output is a score, and comparing an uncalibrated score to nought point five is an arbitrary cut. Every operational number in section nine depends on that cut being meaningful, and nothing establishes that it is. On a pooled corpus this is worse than usual, because the four constituent captures have wildly different base rates and a single global calibration cannot be right for all of them.

REQUIRED REPAIR

Calibrate on a held-out calibration fold, publish a reliability diagram and a Brier score, and check calibration separately within each source corpus.

DEFENSIBLE REPLY

Calibration is monotone. It changes none of our reported rank statistics. The only numbers it touches are the ones at the 0.5 cut, and you have already charged the cut.

ORDERED CORRECTION

Merge into fault 12, keeping the per-corpus sentence — that a single global calibration cannot be right for four captures with different priors — as the reason the operating point matters more on a merge than elsewhere.

Every feature is a timing or volume knob

EVIDENCE

The NetFlow v2 fields are durations, byte and packet counts, throughputs, inter-arrival statistics and TCP flag aggregates. The strongest single feature printed in cell twenty is DURATION_IN at nought point eight seven four three. The words adversarial and evasion appear in no markdown cell of this notebook, and there is no perturbation experiment of any kind.

SCIENTIFIC CONSEQUENCE

An attacker who slows a scan, pads packets, or splits a session changes duration, throughput and packet-size statistics directly, at essentially no cost and with no loss of attack effect. Every column here is a knob under the adversary's hand, and the worst-case recall of this detector under even a modest padding budget is unmeasured. In security, the average case is not the case that matters.

REQUIRED REPAIR

Define a perturbation budget in flow terms — inflate duration by up to a factor, pad packets up to a size — apply it to held-out attack flows, and report worst-case recall under it.

DEFENSIBLE REPLY

Section 13 restricts every number in the notebook to in-distribution separability. You are asking for a worst case against a claim we did not make.

Nothing you removed mattered, and that is the finding

EVIDENCE

Cell twenty-two: headline nought point nine nine eight seven two one, de-duplicated nought point nine nine eight seven one six, top feature dropped nought point nine nine eight seven zero one.

An independent re-run, not printed in your notebook, drops the model's own top importance feature and then the top two, and the AUC stays at nought point nine nine eight eight through both. Your section thirteen says of this, in your own words: the separability is multi-feature; that reflects how this corpus was generated, not one leaky column.

SCIENTIFIC CONSEQUENCE

You wrote the correct interpretation and then declined to act on it. When no feature and no pair of features matters, the classes are not merely separable — they barely overlap, which on a merge of four synthetic-testbed captures is the fingerprint of the testbeds, not of intrusion. Section eleven pre-commits to reading exactly this result as a generation artifact rather than as vindication. Section thirteen then reports nought point nine nine eight seven two one as the headline. The pre-commitment was made and not honoured.

REQUIRED REPAIR

Measure the class overlap directly — the fraction of held-out flows whose nearest neighbour of the opposite class lies within a stated distance — and report it in the abstract as the reason the score is high.

DEFENSIBLE REPLY

Section 11 said we would read a flat ablation as a generation artifact. Section 13 reads it as a generation artifact, in those words. The pre-commitment was to an interpretation, not to withholding the number, and we honoured it.

ORDERED CORRECTION

Delete 'The pre-commitment was made and not honoured'. Re-file as: the correct interpretation is stated but never measured — no class-overlap or nearest-opposite-neighbour statistic appears anywhere, so the generation-artifact reading is asserted rather than demonstrated. Also mark the 0.9988 two-feature figure as auditor output, not notebook output.

Gain importance on forty-one correlated columns

EVIDENCE

Cell eighteen reads best dot feature_importances_ and plots the top twelve. There is no permutation importance and no SHAP anywhere in the notebook. An independent re-run notes that the importance-ranked top feature differs from the AUC-ranked top feature that your validity audit names — the importance panel points at DST_TO_SRC_AVG_THROUGHPUT while cell twenty names DURATION_IN.

SCIENTIFIC CONSEQUENCE

NetFlow fields are heavily correlated with one another — duration, throughput and byte count are near-deterministic functions of each other — and impurity importance splits credit arbitrarily among correlated columns. Two of your own diagnostics disagree about which feature is strongest, which is exactly what that bias produces, and neither is a held-out attribution.

REQUIRED REPAIR

Permutation importance computed on the held-out set, or SHAP values, with the correlated NetFlow fields grouped before attribution.

DEFENSIBLE REPLY

No claim in section thirteen rests on the importance panel — it is presented as a diagnostic figure, and the validity audit deliberately uses a separate single-feature AUC ranking rather than the importances.

You declined the protocol the dataset was built for

EVIDENCE

Section two cites Sarhan, Layeghy and Portmann, twenty twenty-two, and describes the contribution as the forty-three-field NetFlow v2 schema and the NF-star-v2 datasets, adding, in your words, enables cross-dataset comparison. The caveat table in the same cell says of cross-dataset NetFlow studies, in your words, standardized features expose generalization gap. The notebook then runs a pooled random split and reports nought point nine nine eight seven two one, which corresponds to no protocol in the cited literature.

SCIENTIFIC CONSEQUENCE

You named the purpose of the artefact you are using — cross-dataset comparison — and then used it for something else. The resulting number cannot be placed beside any published NF-UQ-NIDS result, because no published result uses a pooled random split on a four-million-row sample. It is a measurement of one file under a protocol invented here.

REQUIRED REPAIR

Report the cross-dataset evaluation Sarhan and colleagues designed the schema to enable, and place your figure beside theirs under the same protocol.

DEFENSIBLE REPLY

You ask us to place 0.998721 beside a published figure but you do not name one. Fault 2 does name numbers — nb36's — and asks for the same protocol. This is the third filing of one experiment.

ORDERED CORRECTION

Either name the published NF-UQ-NIDS result the headline should sit beside, or merge into fault 2.

A grade blind to this corpus's actual defect

EVIDENCE

Cell twenty computes the grade as the worse of a single-feature letter and a contamination letter, and prints data trust grade B, with contamination nought point zero zero zero scoring A. Section thirteen then writes: train-test overlap separately scores A, so overlap is not the issue here.

SCIENTIFIC CONSEQUENCE

The grading scheme tests for two failure modes and this corpus has a third, and the third is the one the whole notebook is about. A clean letter on a two-item checklist reads to a student as a clean bill of health, and this notebook's B is earned entirely by the absence of duplicates — which is genuinely true and entirely beside the point when nearly half the corpus is one capture at ninety-nine point six percent attack. A grade that cannot see the confound should not be reported in a sentence that begins so overlap is not the issue here.

REQUIRED REPAIR

Add a third sub-check to the scheme: the accuracy of a classifier predicting the source corpus from the features. Report that as its own number, and drop the composite letter.

DEFENSIBLE REPLY

Your remedy here is the identical experiment you demand at fault 1. And the ledger says which check drove the grade and reports both underlying quantities, so nothing is hidden behind the letter.

ORDERED CORRECTION

Keep only the narrow point: the sentence 'Train/test overlap separately scores A, so overlap is not the issue here' generalises past what the check tests. Merge the third-sub-check remedy into fault 1.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 1,000,000 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Eight out of thirty. Reject. This notebook has the cleanest data in the entire series — no duplicates, no contamination, an honest reconciliation of the corpus size — and it spends that clean data on the one protocol that cannot answer its own question. The answer is not hidden: five notebooks in this same series print it, the four constituent attack rates reproduce your pooled rate to within one ten-thousandth, and notebook thirty-six reports a cross-corpus mean of nought point five four three four against your nought point nine nine eight seven two one. You had the merge, you had the group column, and you had the answer. Report it.

Static PE Malware Detection on EMBER-2018

EMBER-2018: seven hundred and ninety-nine thousand eight hundred and seventy-six labelled Windows PE files, each reduced to two thousand three hundred and forty-one anonymised static features, balanced fifty-fifty between malware and benign.

HEADLINE

XGBoost, held-out ROC-AUC of point nine nine three — printed as nought point nine nine two nine one eight — with accuracy nought point nine five nine four nine nine on a two hundred thousand file holdout, on a corpus the notebook grades A for trustworthiness.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self A

D1 1 External validity	D2 2 Statistical inference	D3 1 Construct validity	D4 1 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 7 critical (FATAL)
- 16 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This is the cleanest notebook of the four I have, and its data-integrity story is genuinely sound. Contamination is nought point zero zero zero and the duplicate rate is nought point zero zero zero — an independent re-run reads nought point zero zero zero one on both a first and a random window — so the held-out score is a real held-out score, which is more than most of this series can say. It is honest about the anonymised features: it states plainly that it cannot attribute the decision to imports, headers or entropy. And it handles the degenerate family panel better than any comparable notebook — it says in four separate places that family is derived from the label, that the recall panel therefore shows one bar, and that nought point nine five eight is overall malware recall rather than a weak-family finding. It even prints the one-line check a reader could run to confirm the join key is absent. That is exemplary. And every word of it is disclosure.

EXAMINER RULING

I am rejecting this notebook because EMBER ships a published temporal partition built for exactly the question this notebook says it is asking, the notebook names that partition twice and names the metadata field it runs on, and then reports a random split — which measures whether today's malware resembles today's malware, a question nobody was asking.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

You cannot exclude the packer, by your own admission

EVIDENCE

The abstract lists the feature groups: byte histograms, imports, header fields, strings statistics. Section thirteen then says, in your words: the feature-importance view can only name columns as F-hash-hash-hash, because this release ships anonymised features; we therefore cannot attribute the decision to imports, header anomalies or byte-entropy without a column mapping that is not published with the data. Section thirteen also concedes that static features miss packed and obfuscated behaviour. Cell zero's course-text link cites Chapter nine's warning that screening signals are not independent and says, in your words, both apply to artifact-derived features.

SCIENTIFIC CONSEQUENCE

Byte histograms and byte-entropy statistics are dominated by whether a file is compressed or packed. Header fields record which compiler and linker produced it. If the malware and benign halves of this corpus were collected from different sources, as they typically are, then packer and compiler provenance separates the classes without any malicious function being learned at all. That is the single most important alternative explanation for a ROC-AUC of nought point nine nine three on static features, and this notebook cannot test it, because it selected an asset with anonymised columns. I want to be precise: I am not saying packedness is an illegitimate signal. I am saying you cannot tell what your model is using, you said so, and the abstract nonetheless offers the notebook as an audit of whether the score is a benchmark artifact. You cannot audit what you cannot attribute.

REQUIRED REPAIR

Use a release with named features, group the columns by feature family — byte histogram, entropy, header, section, imports, strings — and report the held-out AUC with each group ablated in turn. Report the score on imports and section characteristics alone, with all byte and entropy features removed. That number is the closest thing to a packer-independent estimate.

Forty false alarms for every real detection

EVIDENCE

Cell eleven prints a malware rate of nought point five zero zero zero. Cell eighteen prints an operational false-positive rate at threshold nought point five of nought point zero three eight eight, being three thousand eight hundred and seventy-seven benign files flagged of ninety-nine thousand nine hundred and ninety-four, and a malware recall of nought point nine five eight. Now the arithmetic, on a stated prevalence assumption rather than a measurement from your notebook: an endpoint fleet scanning one million files in which one in a thousand is malicious — so one thousand malware and nine hundred and ninety-nine thousand benign. Nought point zero three eight eight times nine hundred and ninety-nine thousand is thirty-eight thousand seven hundred and sixty-one false alarms. Recall nought point nine five eight on one thousand malware gives nine hundred and fifty-eight true detections. Thirty-eight thousand seven hundred and sixty-one divided by nine hundred and fifty-eight is about forty. So roughly forty false alarms for every true detection, and precision is nine hundred and fifty-eight over thirty-nine thousand seven hundred and nineteen, which is two point four percent.

SCIENTIFIC CONSEQUENCE

Three point eight eight percent reads as a small number and it is a catastrophic number for file scanning, because benign files outnumber malware by three orders of magnitude on any real endpoint. Ninety-seven and a half percent of everything this detector quarantines would be a legitimate file, and quarantining legitimate files is how endpoint products lose customers. The fifty-fifty balance is a property of a curated benchmark and it flatters both metrics in the table; nothing in the notebook restores the deployed prevalence or reports precision under it.

REQUIRED REPAIR

Reweight the held-out set to a stated deployment prevalence, report precision and average precision on it, and select the threshold that meets a stated false-positive budget — for file scanning, on the order of one in ten thousand. Then report recall at that threshold.

You named the field, then split at random

EVIDENCE

Section three says, in your own words: EMBER's own design point is the temporal split — train on earlier months, test on later. We use a random split here. We flag the temporal test as the honest generalization measure we did not run. Section nine goes further and names the field: it lists the upstream metadata keys as sha-two-five-six, appeared, label and avclass. The field called appeared is the month the binary was first observed — it is the field the temporal split is computed from, and you wrote it down. Section nine then states that the Kaggle parquet you chose carries only the feature columns and Label, so that field never reaches the dataframe. Cell sixteen splits with `train_test_split`, `test_size` nought point two five, `stratify` equals `y`.

SCIENTIFIC CONSEQUENCE

Section one of this notebook states the research problem in one sentence: the open question EMBER was built to study is how well such models generalize to future malware, and a single in-distribution split cannot show that. You then ran a single in-distribution split. The canonical protocol is not something you had to invent — it is published with the dataset, it is the dataset's stated design point, and you named the exact column it runs on. This is not an oversight, and it is not an inability. It is a choice, disclosed and not remedied, and it means nought point nine nine two nine one eight answers a question the notebook itself says is not the question.

REQUIRED REPAIR

Obtain the EMBER release that carries the appeared field, sort on it, train on the earlier months and test on the later months, and report the drop from the random-split figure. That drop is the number the abstract should lead with.

DEFENSIBLE REPLY

You quote our own section 9 saying the field never reaches the dataframe. We cannot sort on a column we do not have. If the objection is that we chose the wrong distribution, that is one decision in one line — charge it once, at fault 6, not once per downstream consequence.

ORDERED CORRECTION

Re-file as a consequence of fault 6 rather than as an independent A2 count, and change 'This is not an oversight, and it is not an inability' — on the chosen asset it is precisely an inability, produced by the asset choice.

The canonical baseline is cited and never run

EVIDENCE

Section two cites Anderson and Roth, twenty eighteen, and describes their contribution as, in your words, the dataset, its two-thousand-three-hundred-and-eighty-one-feature vector, and a LightGBM baseline. The caveat table in the same cell says of that baseline: strong in-distribution, time-split generalization is the hard part. No result under that protocol appears anywhere in this notebook, and the LightGBM this notebook does fit is fitted on a random split at eighty rounds with untuned defaults and comes third at nought point nine eight nine one seven six.

SCIENTIFIC CONSEQUENCE

There is a published number for this dataset, produced by its authors, on a defined split. You cited it, you characterised its weakness correctly, and then you produced a figure under a protocol nobody else uses, which means no reader can say whether nought point nine nine two nine one eight is better than, worse than, or identical to what was already known in twenty eighteen. A number that cannot be placed beside a prior number is not a contribution to a literature. It is a measurement of one file.

REQUIRED REPAIR

Reproduce the Anderson and Roth baseline under their split and report both numbers in the same table. That is the minimum for a benchmark result to enter the record.

DEFENSIBLE REPLY

Their protocol is the temporal split, which needs the `appeared` field this asset does not ship. You have asked for the same missing column three times under three family codes.

ORDERED CORRECTION

Reduce to the part that is achievable on this asset: the notebook cites a published baseline and never states its number, so a reader has no anchor even qualitatively.

You cited the evasion paper and ran no evasion

EVIDENCE

Section two cites Kolosnjaji and colleagues, twenty eighteen, described in your own words as adversarial evasion of PE malware classifiers. The caveat table in the same cell says of both engineered-feature and raw-byte approaches: both are evadable, Kolosnjaji twenty eighteen. There is no perturbation experiment, no evasion budget and no worst-case recall anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Static PE detection is the one machine-learning security task where the adversarial literature is unambiguous and operationally realised — attackers append bytes, pad sections, repack, and rewrite headers as routine practice, and the published attacks flip these classifiers with small append-only modifications that do not change program behaviour. You cited the paper. You wrote the word evadable in your own caveat table. And the number you report is the average case against a non-adapting opponent, which is the only case that never occurs in deployment. The worst case of this detector is unmeasured and it is the only number an endpoint vendor would ask for.

REQUIRED REPAIR

Apply an append-and-pad perturbation budget of a stated size to the held-out malware and report recall under it. Report the smallest byte-append budget that halves recall. That figure, not nought point nine nine two nine one eight, is the security claim.

DEFENSIBLE REPLY

Section 13 restricts every number to in-distribution separability and says outright that this does not measure catching tomorrow's malware. Citing the evasion literature and then scoping our claim to exclude it is the correct move, not a contradiction.

ORDERED CORRECTION

Recast as E4 rather than D2: the aggravating fact is that the notebook cites Kolosnjaji and writes 'evadable' in its own table, which is naming the experiment — that framing survives the scoping defence where a bare 'no adversarial evaluation' does not.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

The mirror you chose deleted three capabilities

EVIDENCE

Cell eleven downloads Kaggle d-hoogla slash ember-two-thousand-eighteen-v-two-features. Section nine names the authoritative source itself: ember slash double-underscore-init-double-underscore dot p-y in the elastic slash ember repository, which lists the metadata keys sha-two-five-six, appeared, label and avclass. Section nine then states that the Kaggle parquet carries only the feature columns and Label. So the mirror removes appeared, which is the temporal split key; avclass, which is the malware family label; and sha-two-five-six, which is the join key that would restore both. The abstract also says two thousand three hundred and eighty-one features while the loader prints two thousand three hundred and forty-one, a difference produced by the constant-column drop at the end of cell eleven.

SCIENTIFIC CONSEQUENCE

Three of this notebook's four most serious methodological gaps — no temporal split, no family grouping, no attribution of the decision — are consequences of a single upstream choice made in one line of cell eleven, and the notebook identifies the authoritative alternative by repository name in section nine. The convenience of a pre-vectorised parquet was purchased with the ability to run the protocol the dataset exists for.

REQUIRED REPAIR

Download from elastic slash ember, or join this parquet back to the original JSONL metadata on sha-two-five-six. Section nine already names the join. Run it.

Ten thousand variants of one family, split at random

EVIDENCE

Cell sixteen splits with `train_test_split` and `stratify equals y`, with no `groups` argument. Cell eleven sets `family` as `y` mapped to `benign` and `malware` — derived from the label, so no real grouping variable exists in the dataframe. An independent re-run confirms one group among held-out positives, and cell eighteen prints the whole per-group recall panel as the single entry `malware` nought point nine five eight. Section nine names the field that would fix this: `avclass`, produced by the `AVClass` labelling tool of Sebastian and colleagues, twenty sixteen — absent from the chosen `parquet`.

SCIENTIFIC CONSEQUENCE

Malware arrives in families. Thousands of samples of one family differ by a few bytes of configuration and produce nearly identical static feature vectors. And now hold this against the clean contamination number I conceded a moment ago, because a student will: nought point zero zero zero duplicates does not contradict this, it is the reason the fault matters. An exact-match duplicate check cannot see a family sibling — two variants differing in a few configuration bytes hash to different rows and register as entirely distinct. So a corpus can be free of duplicates and still be saturated with near-siblings, and only a family group split can tell the difference. Under a random split the test set is populated by siblings of training samples, so nought point nine nine two nine one eight measures recognition of families the model has already seen. The question a detector is deployed to answer is whether it flags a family it has never seen — and the standard protocol for that is a group split on family, which is impossible on this asset because you chose the asset that lacks the family column.

REQUIRED REPAIR

Restore `avclass` by joining on `sha-two-five-six`, then `GroupKFold` on the family label. Report held-out AUC on malware families entirely absent from training.

DEFENSIBLE REPLY

Section 9 says the family column is absent and prints the one-line check a reader can run to confirm it. You cannot group on a column that is not there. The asset choice is the charge, and you have already made it.

ORDERED CORRECTION

Keep the exact-match-cannot-see-a-sibling paragraph — it is the best argument in the filing for why clean contamination is not clean data — but attach it to fault 6 as the cost of the mirror rather than filing it as an independent A1.

The hardest files were dropped without a count

EVIDENCE

Cell eleven contains the line `df equals df where Label is not equal to minus one`, followed by `reset index`, then an assertion that the length is at least four hundred thousand, then a print of the seven hundred and ninety-nine thousand eight hundred and seventy-six survivors. The count removed is never printed. The comment gives the reason: you cannot train supervised on them. Section three then presents the label as, flatly, zero benign, one malware.

SCIENTIFIC CONSEQUENCE

The reason is operationally true and epistemically silent. The minus-one rows are precisely the files whose maliciousness the labelling process could not resolve — the contested, the ambiguous, the ones a vendor's analysts argued over. Deleting them yields the confidently-labelable subset and reports a score on it as though it were a score on Windows PE files. The hardest cases were removed from the exam before it was graded, and the reader is not told how many. That number is a first-order measurement of how hard this problem actually is, and it costs one print statement.

REQUIRED REPAIR

Print the number of unlabelled rows removed and their proportion. State in the abstract that the corpus is the confidently-labelable subset. If possible, score the model on those rows and report the distribution of its outputs — that is an informative experiment about the model's behaviour on genuine ambiguity.

DEFENSIBLE REPLY

The -1 rows are EMBER's designated unlabeled split, not a subset we chose to discard. The dataset's own baseline drops them too. What we owe you is the count, which is fair.

ORDERED CORRECTION

Reduce to the reporting demand — print the removed count and its proportion, and say in the abstract that the corpus is the labelled split — and drop 'the hardest cases were removed from the exam before it was graded', which implies a deviation from the published protocol that did not occur.

Two thousand three hundred features, fifty-one rows each

EVIDENCE

Cell sixteen prints materialized seven hundred and ninety-nine thousand eight hundred and seventy-six rows, trained on one hundred and twenty thousand stratified subsample, held-out one hundred and ninety-nine thousand nine hundred and sixty-nine. One hundred and twenty thousand divided by two thousand three hundred and forty-one is fifty-one — about fifty-one training rows per feature. No learning curve appears in the notebook and no regularisation parameter is searched.

SCIENTIFIC CONSEQUENCE

In a two-thousand-dimensional space, fifty-one observations per dimension is thin, and the notebook has no evidence about whether the score has saturated or is still climbing. It also means the subsample cap — introduced as an honest disclosure about compute — silently discards eighty-five percent of the available labelled data on the one corpus in this set where the dimensionality makes that expensive.

REQUIRED REPAIR

Run a learning curve from ten thousand rows to the full six hundred thousand available for training, and report where it saturates. If it has not saturated at one hundred and twenty thousand, the headline understates and the number should be re-run at full size.

DEFENSIBLE REPLY

Gradient-boosted trees do not require observations to exceed dimensions, the cap is disclosed in section eight as a subsample number rather than a full-corpus claim, and the three-fold cross-validation spread of plus or minus nought point zero zero zero six suggests the estimate is already stable.

No search in a two-thousand-dimensional space

EVIDENCE

Cell sixteen fixes RandomForest at sixty trees, XGBoost at eighty rounds and depth six, LightGBM at eighty rounds. No grid, no randomised search, no regularisation tuning. Cell twenty-four sets cvX to Xtr dot iloc of colon forty thousand — the first forty thousand rows of the training subsample — and the nought point nine eight nine zero plus or minus nought point zero zero zero six that section thirteen carries is that forty-thousand-row figure.

SCIENTIFIC CONSEQUENCE

On a two-thousand-three-hundred-and-forty-one column feature set, the number of boosting rounds and the depth and the column-sampling rate are the parameters that decide the answer, and none were searched. Eighty rounds at depth six is a shallow default for this dimensionality. So the ranking compares three packages' defaults rather than three methods, and the plus-or-minus in the conclusion belongs to an estimator fitted on different data than the headline.

REQUIRED REPAIR

Nested cross-validation with an inner selection loop over depth, rounds and column subsampling, and an outer loop producing the estimate.

The conclusion names a feature that loses on the full corpus

EVIDENCE

Cell twenty-four prints seed equals zero, and cell sixteen draws one split. Section thirteen names F-four-nine-nine in bold as the strongest single feature at nought point seven one one two. An independent eight-seed re-run — which your notebook does not print — finds values from nought point seven zero nine three to nought point seven one three six, and finds that on the full corpus the strongest single column is F-four-nine-seven rather than F-four-nine-nine, with F-four-nine-seven winning six of the eight seeds. The two are separated by about four ten-thousandths of AUC.

SCIENTIFIC CONSEQUENCE

The conclusion puts a specific column name in bold as a finding, and which column that is depends on which sixty thousand rows were drawn. No number in the ledger changes, and I want to be fair about that — but a notebook that names a feature in bold has made a claim about that feature, and the claim does not survive a re-draw.

REQUIRED REPAIR

Report the top-k single features with their spread across seeds rather than naming one, and refit the headline across at least ten seeds with mean and standard deviation.

DEFENSIBLE REPLY

Both columns grade A and differ by four ten-thousandths, the ledger's substantive claim is that no single feature is near-sufficient, and that claim is unaffected by which of two tied columns is named.

You drew the precision-recall curve and never read it

EVIDENCE

Cell eighteen calls `precision_recall_curve` on the held-out probabilities and plots it in the third panel. There is no call to `average_precision_score` anywhere in the notebook, and every ranking in cell sixteen is by ROC-AUC. The corpus is fifty-fifty by construction.

SCIENTIFIC CONSEQUENCE

Be precise about this one, because the figure is in front of you and it is drawn correctly. The fault is not that the diagnostic is missing — it is that you plotted the right diagnostic and declined to read it. At fifty-fifty, ROC-AUC and average precision agree and nothing much turns on it. At one malware in a thousand, they diverge enormously, and average precision is the one that tracks what an operator experiences. You produced the curve whose area answers the deployment question and never computed the number.

REQUIRED REPAIR

Compute average precision on a test set reweighted to a stated deployment prevalence, put it in the ranking table, and rank on it.

DEFENSIBLE REPLY

This ends in the same sentence as fault 5: reweight to a deployment prevalence and report precision there. One experiment, one charge.

ORDERED CORRECTION

Merge into fault 5 as its methodological half — the arithmetic there and the missing metric here are the same finding.

Every figure is a point estimate

EVIDENCE

The ranking table in cell sixteen, the false-positive rate and recall in cell eighteen, the single-feature audit in cell twenty and the ablation table in cell twenty-two are all bare point estimates. The only dispersion figure anywhere is the three-fold cross-validation standard deviation of nought point zero zero zero six in cell twenty-four, computed on forty thousand rows of the training subsample. There is no bootstrap and no binomial interval.

SCIENTIFIC CONSEQUENCE

The ablation in cell twenty-two turns on differences of eleven and thirty-seven millionths, and the conclusion interprets both — one as a shortcut result, one as re-split noise. Neither interpretation is supportable without knowing the resampling spread, and it was never measured on the reported quantity.

REQUIRED REPAIR

Bootstrap the held-out AUC with at least a thousand resamples and report a ninety-five percent interval, and attach a binomial interval to the malware recall of nought point nine five eight.

DEFENSIBLE REPLY

We interpret both ablation differences as nothing. An interval around a difference of eleven millionths on a 200,000-row holdout would contain zero and would confirm exactly what we wrote. And a binomial interval on 0.958 is about a thousandth wide.

ORDERED CORRECTION

Note that the de-duplication row is labelled '0% rows removed' and the corpus was genuinely unchanged, so the 37 millionths is re-split noise by construction — which the notebook says. The durable demand is a bootstrap on the headline AUC, not intervals on differences already read as null.

14

MAJOR MEASURED VERIFIED FALSE PRECISION

Six decimals, differenced at eleven millionths

EVIDENCE

Cell twenty-two: headline nought point nine nine two nine one eight, de-duplicated nought point nine nine two eight eight one, top feature dropped nought point nine nine two nine zero seven. Section thirteen quotes both the eleven-millionth drop and the thirty-seven-millionth de-duplication difference as though each were a quantity.

SCIENTIFIC CONSEQUENCE

Quoting sixth decimals invites a reader to treat differences far smaller than any measured error as effects — and here the de-duplication row is labelled zero percent rows removed, meaning the corpus was not changed at all, so the thirty-seven millionths is pure re-split noise by construction.

REQUIRED REPAIR

Report three decimals, or six accompanied by a measured resampling spread.

DEFENSIBLE REPLY

The cell sixteen code comment states six decimals are deliberate so that a score of one point zero zero zero zero zero zero cannot hide behind rounding, and the conclusion explicitly labels the de-duplication difference as re-split noise rather than an effect.

15

CRITICAL UNTESTED VERIFIED THRESHOLD FIXED AT 0.5, NO COST MODEL

An antivirus verdict at a library default

EVIDENCE

Cell eighteen sets pred equal to probability greater than nought point five, producing the nought point zero three eight eight false-positive rate, the confusion matrix and the nought point nine five eight recall. No cost matrix appears in the notebook and no threshold is selected from the precision-recall curve the same cell draws.

SCIENTIFIC CONSEQUENCE

For file scanning the cost asymmetry is extreme and well documented — quarantining a system binary is a production outage, missing one sample of a family the fleet has ten thousand copies of is often survivable. When costs are asymmetric the operating point is the result, and one half is not an operating point. Endpoint vendors specify false-positive budgets in the range of one in ten thousand to one in a million; this notebook reports three point eight eight in a hundred and does not remark on the distance.

REQUIRED REPAIR

State a cost matrix, sweep the threshold, and report recall at a false-positive rate of one in ten thousand. If recall at an admissible false-positive rate is unusable, that is the finding.

Atlas proof

docs/course/security-ml-notebooks/nb24-ember-pe-malware.ipynb · cells 16, 18, and 25; zero-based manifest cell IDs: 16=1b27f611, 18=57af9499, 25=169a5da3

A winner declared with no significance test

EVIDENCE

Cell sixteen: XGBoost nought point nine nine two nine one eight, RandomForest nought point nine nine one zero eight zero, LightGBM nought point nine eight nine one seven six. The gap between first and second is nought point zero zero one eight three eight. There is no McNemar test, no paired comparison and no interval anywhere in the notebook, and the winner determines everything downstream — the ablation clones it, the cross-validation scores it, the operational figures are its.

SCIENTIFIC CONSEQUENCE

A ranking asserted without a test is a ranking asserted, and the notebook's own reference point — Anderson and Roth's published baseline — is a LightGBM, which comes third here at untuned defaults. Whether that ordering reflects the methods or the defaults is exactly what a test and a search would resolve.

REQUIRED REPAIR

McNemar's test on paired held-out predictions between the top two models, with a bootstrap interval on the AUC difference.

DEFENSIBLE REPLY

On a one-hundred-and-ninety-nine-thousand-row holdout the standard error of an AUC estimate is small enough that a gap of nearly two thousandths is very likely a real ordering, and the accuracy column agrees with the AUC column here, unlike elsewhere in this series.

Uncalibrated scores drive a quarantine decision

EVIDENCE

Cell eighteen calls `predict_proba` and compares to nought point five. No Platt scaling, no isotonic regression, no reliability diagram and no Brier score appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

An endpoint product does not consume a binary verdict; it consumes a confidence and routes it — auto-quarantine at high confidence, cloud lookup in the middle, allow at low. That routing requires the score to mean a probability, and a boosted tree's raw output does not. Every operational number in section nine rests on a comparison that has not been shown to be meaningful.

REQUIRED REPAIR

Calibrate on a held-out calibration fold, publish a reliability diagram and a Brier score, and define the verdict bands on the calibrated scale.

DEFENSIBLE REPLY

Calibration is monotone and changes none of our rank statistics. It moves the cut, and you have already charged the cut. We publish no confidence bands for it to invalidate.

ORDERED CORRECTION

Merge into fault 15, keeping the auto-quarantine / cloud-lookup / allow routing sentence as the reason the operating point matters more in endpoint security than elsewhere.

Two columns removed out of two thousand three hundred

EVIDENCE

Cell twenty-two drops one column, F-four-nine-nine, and reports nought point nine nine two nine one eight falling to nought point nine nine two nine zero seven. An independent re-run drops the model's own top importance feature and then the top two, and the AUC goes from nought point nine nine four one to nought point nine nine four zero to nought point nine nine four two — it goes up. Two columns out of two thousand three hundred and forty-one is nought point zero eight five percent of the feature set. Section thirteen reads the result as: the ablation refutes a single-feature story, so the separability is multi-feature, reflecting redundancy across many static features.

SCIENTIFIC CONSEQUENCE

The reading is correct and the test is vacuous. On two thousand three hundred correlated engineered features, removing one or two columns cannot move any score, so the experiment has no power to detect anything and its null result carries no information. Presenting it in the validity ledger as evidence that the headline is not an artifact gives a reader false assurance, and it is the criterion on which the friendly review awarded full marks. A test that cannot fail is not a test.

REQUIRED REPAIR

Ablate by feature group, not by column — remove all byte-histogram features, then all entropy features, then all header features — and report the AUC after each. On an anonymised asset that requires the named release, which is fault six.

Nothing here explains the score away

EVIDENCE

Cell twenty computes the trust grade as the worse of a single-feature letter and a contamination letter and prints data trust grade A. Section thirteen then writes, in your words: neither check flags a problem, so nothing here explains the score away.

SCIENTIFIC CONSEQUENCE

That sentence is the most consequential in the notebook and it is not supported. The two checks test for a near-sufficient single feature and for exact-row overlap. They cannot see the absence of a temporal split, the absence of family grouping, the impossibility of feature attribution, the untested packer confound, or the missing adversarial evaluation — five explanations for the score, all of which this notebook names elsewhere in its own prose. Passing a checklist of two is not the same as no explanation existing, and a grade of A printed beside a headline invites exactly the reading the sentence then makes explicit.

REQUIRED REPAIR

Restrict the grade's claim to what it tests, in the sentence that reports it, or drop the letter and report the two underlying quantities with intervals.

Fifty-fifty is a curation decision, not a fact

EVIDENCE

Section three says, in your words: these are static features, no dynamic or behavioural signal, and the set is a curated benchmark; a high in-distribution score therefore does not imply detection of novel malware. Cell eleven prints a malware rate of nought point five zero zero zero. Cell sixteen prints a majority-class baseline of nought point five zero zero zero. The abstract describes the work as auditing whether the score reflects generalizable structure or a benchmark artifact.

SCIENTIFIC CONSEQUENCE

Both the class balance and the sample composition are curation decisions taken by the benchmark's authors, and both determine the headline. Fifty-fifty makes accuracy meaningful and ROC-AUC flattering; a real fleet makes neither true. So the notebook's central audit question — structure or artifact — has a printed partial answer sitting in its own loader output, and the ledger reports the score without carrying it forward. You wrote the disqualifying sentence in section three and did not apply it to the number in section thirteen.

REQUIRED REPAIR

State the curated balance in the first sentence of the abstract, report every operational figure at a stated deployment prevalence, and restrict the claim to the benchmark unless a prevalence-corrected number is given.

DEFENSIBLE REPLY

The balance is printed by the loader, printed again as the majority baseline, and stated in the provenance table. And your fix here is the fix you already demanded at fault 5.

ORDERED CORRECTION

Merge the prevalence remedy into fault 5 and keep only the narrow residual: section 3's disqualifying sentence about curation is never carried into the section 13 ledger that reports the score.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 199,969 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Eight out of thirty. Reject. And I want to end on what makes this the most frustrating of the four, because its data is clean and its prose is careful: every one of the five faults I ranked highest was already identified, in writing, inside this notebook. Section three names the temporal split. Section nine names the field it runs on and the repository that carries it. Section thirteen names the attribution you cannot perform. Section two names the evasion paper. Section three names the curation. Five experiments, all named, none run — and then a grade of A and a sentence saying nothing here explains the score away.

Encrypted DNS-over-HTTPS Tunnel Detection (CIRA-CIC-DoHBrw-2020)

Four hundred and ninety-nine thousand, one hundred and six DoH flows described by twenty-eight timing statistics, half of them benign browsing and half of them tunnels built with three tools in a laboratory.

HEADLINE

XGBoost separates benign DoH from tunnelled DoH at a held-out ROC-AUC of 0.999999 and an accuracy of 0.999800, on a corpus the notebook grades A for trustworthiness.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 1 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 21 live faults
- 4 critical (FATAL)
- 17 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me start with what is genuinely right here, because it is not small. This is the cleanest data file in my set: a single-feature audit that tops out at 0.7762 and an exact-row contamination of one point four percent, both of which earn a genuine A on your own scale. And unlike its neighbours, this corpus is not linearly trivial - your logistic regression reaches only 0.960478 while the trees reach six nines, which means there is genuine non-linear structure here rather than two distributions sitting on opposite sides of a hyperplane. Section 3 tells me, unprompted, that the fifty-fifty balance was manufactured by the distributor and is not the field rate. You did the audit. My objection is to what you did after the audit.

EXAMINER RULING

This notebook measured, with great care and complete honesty, how well a machine can recognise three specific pieces of tunnelling software running on one laboratory network, and then reported that number as a tunnel detector.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

1

CRITICAL

UNTESTED

VERIFIED

FEATURES ENCODE THE GENERATOR, NOT THE ADVERSARY

ORAL TOP 5

You detected three tools, not tunnelling

EVIDENCE

Section 3 of your own provenance table names them: dns2tcp, DNSCat2 and Iodine. Cell 11 loads twenty-eight timing features and drops every identifier. Cell 16 prints a held-out ROC-AUC of 0.999999 across one hundred and twenty-four thousand, seven hundred and seventy-seven flows.

SCIENTIFIC CONSEQUENCE

I do not have to argue this against you, because section 3 argues it for me. Your own words: a high score may reflect tool-specific timing fingerprints rather than a general tunnel detector. That is the alternative hypothesis, you wrote it down, and the notebook contains no experiment that separates it from the hypothesis in your title. When a model reaches six nines on twenty-eight timing statistics drawn from three programs, the tool-fingerprint explanation is at least as well supported by the evidence as the detection explanation, and the notebook offers the reader no way to choose. The claim in your title is about DoH tunnels in general; the evidence supports a claim about three named tools as captured.

REQUIRED REPAIR

Hold out one tool entirely. Train on dns2tcp and DNSCat2, test on Iodine, then rotate. Report all three leave-one-tool-out scores next to the pooled number. If the pooled number is 0.999999 and the leave-one-tool-out numbers are 0.6, you have measured a fingerprint, and you will have measured it in about forty lines of code.

4

CRITICAL

UNTESTED

VERIFIED

NO ADVERSARIAL EVALUATION

ORAL TOP 5

You named the evasion and did not run it

EVIDENCE

The opening cell of this notebook says, in your own words, that several features used here are flow statistics - sizes, timings, rates - an attacker can reshape by padding or pacing. Your literature review in cell 3 cites Bushart and Rossow, 2020, on the padding and anonymity limits of encrypted DNS. Nowhere in cells 10 through 24 is a single byte of padding added or a single packet delayed.

SCIENTIFIC CONSEQUENCE

You have identified the exact perturbation that defeats your feature set, cited the paper that establishes it, and then reported 0.999999 without applying it. For a security detector the worst case is the result. A number measured only against a passive adversary is not a security claim; it is a claim about a static file.

REQUIRED REPAIR

Define a perturbation budget - say, pad every packet to the next multiple of one hundred and twenty-eight bytes and jitter inter-arrival times by up to fifty milliseconds - apply it to the held-out positives only, and report recall after the perturbation. That is the number your title promises.

The grouping key was read, then deleted

EVIDENCE

Cell 11 contains an explicit drop list. It removes, by name, sourceip, destinationip, timestamp and flowid. Cell 16 then calls train underscore test underscore split with test size zero point two five, stratified on the label and nothing else.

SCIENTIFIC CONSEQUENCE

This is not a case of a missing column. Your loader saw the source IP, named it in a list, and threw it away. The very act of naming it proves the grouping key was available at the moment the split was drawn. A tunnel is a session: hundreds of flows carrying one conversation between one client and one server. Splitting those flows at random puts most of a session in training and the rest in test, and the test flows are then near-copies of training flows. The held-out set is not held out in any sense that supports the word generalisation.

REQUIRED REPAIR

GroupShuffleSplit keyed on the source IP, or on the source-destination pair. Twenty lines. Report the pooled score and the grouped score side by side and let the gap be the finding.

DEFENSIBLE REPLY

Run the expression against my header. It is a comprehension guarded by `if c.lower() in (...)`, and the file has twenty-nine columns — twenty-eight statistics and Label. It matches nothing. Naming a column in a defensive filter is not evidence that the column was present; it is evidence that the same loader pattern is shared across a series in which some corpora do carry it. The source IP was never in my hand, so GroupShuffleSplit keyed on it is not twenty lines, it is a different dataset. What I will concede, and it is not small, is the charge underneath: I chose a redistribution that had already stripped every entity key, and I therefore cannot demonstrate that my held-out set is independent of my training set in any sense that supports the word generalisation. That is a defect of corpus selection, and it belongs in section 3 rather than in section 8.

ORDERED CORRECTION

Recharge at CORRECTION 3 Tier 2: the entity keys exist in the authoritative CIRA release and were stripped by the chosen mirror, so the defect is mirror selection, not split design. Delete 'Your loader saw the source IP, named it in a list, and threw it away' and delete the GroupShuffleSplit prescription.

A manufactured prior makes every operational number fiction

EVIDENCE

Section 3 states it plainly: the exact fifty-fifty balance is constructed, BCCC re-balanced the natively skewed CIRA-CIC-DoHBrw-2020 capture, and that is why the majority baseline prints as exactly 0.5000. Cell 18 then computes an operational false-positive rate of 0.0002 against sixty-two thousand, three hundred and eighty-nine benign flows, which is exactly half the held-out set.

SCIENTIFIC CONSEQUENCE

Let me be precise, because one part of this survives and one part does not. The false-positive rate itself is computed on negatives only, so it does not move with prevalence, and I am not disputing the 0.0002. Everything built around it does move. Do the arithmetic with me. At your manufactured prior of one half, the always-benign baseline is 0.5000 and XGBoost's accuracy of 0.999800 clears it by 0.4998, which looks like an enormous achievement. Now state a prior a network operator would recognise - malicious DoH under one flow in a thousand. The always-benign baseline is then above 0.999, and 0.999800 clears it by essentially nothing. The same printed number, the same model, and the achievement evaporates. And precision, the quantity that decides whether an analyst can work the queue, is never computed anywhere in this notebook at any prior at all. You disclosed the manufactured balance in section 3 and then reported accuracy against a baseline that the manufacture created.

REQUIRED REPAIR

Re-weight the held-out set to a stated field prior, or report accuracy and precision across a range of priors. Report precision at the operating point, which this notebook never computes, at a prevalence you are willing to defend in writing.

DEFENSIBLE REPLY

Your arithmetic on accuracy is right and I accept it: at a field prior my 0.999800 clears the always-benign baseline by essentially nothing, and I should not have let accuracy sit in a ranking table without that caveat attached to it rather than three sections earlier. But my headline is a ranking statistic. Re-weight the held-out set to one in a thousand and 0.999999 does not move, because ROC-AUC is a function of the score ordering and not of the base rate — and my ledger says exactly that in the sentence you did not quote. The half of this that lands, and lands hard, is the last one: precision is never computed at any prior anywhere in this notebook.

ORDERED CORRECTION

Concede that ROC-AUC is prevalence-invariant and quote section 13's own sentence saying so, then rest the charge on the accuracy comparison and on the total absence of precision — which is also faults 11, 13 and 15, and should be scored once.

5

MAJOR

MEASURED

DOWNGRADED

PER-GROUP RECALL REPORTED ON A VACUOUS GROUPING

ORAL TOP 5

Your headline honesty check prints one bar

EVIDENCE

Cell 18 prints, verbatim, worst per-family recalls colon, open brace, Malicious, one point zero, close brace. One entry. Cell 0 lists as learning objective four: report per-group recall, because the rare classes carry the risk. Section 9 concedes that on binary corpora the grouping collapses to the positive class.

SCIENTIFIC CONSEQUENCE

The diagnostic exists to expose failures on rare classes. Here there are no classes, rare or otherwise - there is one label called Malicious and its recall is one point zero by construction of a binary problem on a fifty-fifty file. Your conclusion nonetheless reads: every group listed is recovered essentially perfectly, so there is no rare-class failure to report on this split. That sentence describes a check that could not have failed. Presenting an unfalsifiable check as a passed check is the failure.

REQUIRED REPAIR

The grouping variable available to you is the tunnelling tool. Recover it from the source label, print recall for dns2tcp, DNSCat2 and Iodine separately with the positive count beside each, and say which tool the detector is weakest on.

DEFENSIBLE REPLY

You say I presented an unfalsifiable check as a passed check. Read the next sentence of my own ledger: 'That uniformity suggests the corpus is easy to separate, not that the detector is strong.' That is the opposite of reassurance and it is the sentence directly after the one you quoted. And your remedy — recover the tunnelling tool from the source label — cannot be executed: the label in this mirror has two values, two hundred and forty-nine thousand five hundred and fifty-three each. The residual I accept is that an advertised learning objective produced a one-bar panel, and that I should have said so in section 7 rather than let the panel print.

ORDERED CORRECTION

Delete 'The grouping variable available to you is the tunnelling tool. Recover it from the source label.' Delete the claim that the ledger presents the check as passed, or quote the ledger's contrary sentence and charge the weaker true thing: an advertised diagnostic that could not fail was run and printed anyway.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

7

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

You wrote down the missing experiment and shipped anyway

EVIDENCE

Section 13 reads: generalization to unseen tools and to padded or normalized DoH - not measured here - is the real test. The same section reads: the split is random, not temporal or entity-grouped. Every number above therefore measures in-distribution separability only. And the same section reports 0.999999.

SCIENTIFIC CONSEQUENCE

At doctoral level, naming a fatal limitation is not a defence. It is a confession. You told me which experiment decides your claim, you told me you did not run it, and you kept the headline in the abstract, in the title slide, and in the ledger. The disclosure paragraph costs nothing and buys nothing. The experiment was affordable and was declined.

REQUIRED REPAIR

Either run the leave-one-tool-out and the grouped split, or delete 0.999999 from every summary position and replace it with the sentence: on this corpus, under a random split, the classes are separable. That sentence is true and defensible. The number as presented is neither.

8

MAJOR

UNTESTED

REMECHANISED

NO MODEL SELECTION; THE REPORTED CV IS NOT THE REPORTED PIPELINE

Four hardcoded configurations and a cross-validation on a slice

EVIDENCE

Cell 16 hardcodes four learners: logistic regression with max iter three hundred, a random forest of sixty trees, XGBoost with eighty estimators at depth six, and LightGBM with eighty estimators. Cell 24 sets cvX equal to Xtr dot iloc, colon forty thousand - the first forty thousand rows of the one hundred and twenty thousand row training subsample - and prints a three-fold CV ROC-AUC of 1.0000 plus or minus 0.0000.

SCIENTIFIC CONSEQUENCE

The conclusion presents that 1.0000 as corroboration of the headline. It is not. It is a three-fold estimate on a third of the training data, never touching the held-out set, on a corpus where the classes are almost disjoint. And because no search was ever run, the reader cannot distinguish a well-chosen model from an arbitrary one. Model selection and model assessment have been collapsed into a single unsearched pass.

REQUIRED REPAIR

Nested cross-validation. An inner loop that selects, an outer loop that estimates. If the argument is that these learners saturate and search is pointless, then show a learning curve and a small grid demonstrating saturation, rather than asserting it.

DEFENSIBLE REPLY

A fixed-configuration comparison is not a claim that the configuration is optimal, and I never made one. The error you are reaching for is more specific and worse: I chose the winner on the test partition and then quoted the test partition. Nested cross-validation would fix that, but so would a validation split at a fraction of the cost, and neither is present.

ORDERED CORRECTION

Recast as selection-on-test with no validation partition. Keep the cross-validation-on-a-training-slice observation and attach it to fault 12.

One draw of the split decides the ranking

EVIDENCE

Cell 24 prints seed equals zero.
Cell 16 draws one split at random
state RANDOM underscore STATE
and every subsequent cell reuses
it.

SCIENTIFIC CONSEQUENCE

The winner is separated from the
runner-up by one part in a
million. Without a distribution
over splits you cannot say
whether that ordering is a
property of the models or of the
draw. Every seed-sensitive
quantity in the ledger - the six-
decimal AUCs, the false-positive
count of fourteen, the
contamination of one point four
percent - is a single realisation
reported as a value.

REQUIRED REPAIR

Ten seeds. Report mean and
standard deviation for the
headline AUC and for the false-
positive count, and let the reader
see the spread.

DEFENSIBLE REPLY

The three-fold cross-validation in cell 24 returns a standard deviation of exactly zero, so on this corpus the estimate is visibly not seed-fragile, and a ten-seed sweep would print the same number ten times.

You plotted the right diagnostic and declined to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and draws it in the third panel of the first figure. Search the notebook for average precision score: it is not there. The ranking table in cell 16 is sorted on roc underscore auc.

SCIENTIFIC CONSEQUENCE

This is worse than omitting the diagnostic. You built it, you looked at it, and you ranked on something else. ROC-AUC is insensitive to the absolute count of false positives when negatives are plentiful, and in deployment - as opposed to this manufactured fifty-fifty file - negatives are overwhelmingly plentiful. The quantity that survives a change of prevalence is average precision, and you had the curve in your hand.

REQUIRED REPAIR

Compute average precision, print it, and rank on it. One import and one line, in a cell that already imports the curve.

DEFENSIBLE REPLY

Two things. On this file there are 62,388 positives and 62,389 negatives in the held-out set, so a false positive moves precision and the false-positive rate by the same amount. The asymmetry you invoke needs one class to be scarce and neither is. And your remedy is backwards: precision is a function of the base rate, so average precision computed on a manufactured fifty-fifty file would tell a deployed resolver at one in a thousand nothing at all — ROC-AUC is the invariant one. What I accept is the narrow fact: I drew the curve and never computed its area, and I ranked on something else.

ORDERED CORRECTION

Delete 'ROC-AUC is insensitive to the absolute count of false positives when negatives are plentiful' and 'The quantity that survives a change of prevalence is average precision'. Charge only the omission — the curve is drawn and its area never computed — and note that on a manufactured 50/50 file average precision would itself be a fiction, which is fault 3.

Six decimals, zero intervals

EVIDENCE

Cell 16 reports 0.999999 as a point. Cell 18 reports a false-positive rate of 0.0002 resting on fourteen misclassified flows. Cell 20 reports a contamination of 0.014 estimated from a fifty-thousand-row window. Cell 24 reports a standard deviation of 0.0000 across three folds.

SCIENTIFIC CONSEQUENCE

Fourteen events is a small number. The binomial interval around fourteen out of sixty-two thousand, three hundred and eighty-nine is wide relative to the quantity, and you have printed the point estimate to four decimals with no interval at all. A standard deviation of exactly zero across three folds is not evidence of stability; on a near-separable corpus it is what a degenerate estimator prints. Every comparison in the ledger is made between numbers whose sampling spread is unknown.

REQUIRED REPAIR

Bootstrap the held-out AUC with at least one thousand resamples and report the interval. Put an exact binomial interval on the false-positive rate and on the contamination estimate.

DEFENSIBLE REPLY

I accept the criticism. The exact binomial interval on 14 of 62,389 runs from roughly 0.00012 to 0.00038 — a factor of three — and I printed the point to four decimals. That is a real defect in how I reported an operational rate. It is not a reason to doubt the ranking statistic, which is measured on 124,777 rows.

ORDERED CORRECTION

Strike 'A standard deviation of exactly zero across three folds' — the line formats with `:.4f`, so any spread below $5e-5$ renders as 0.0000 and nothing establishes exactness. The durable residual is the 3-fold CV on a 40,000-row slice of train being carried into the ledger as corroboration, which is the stronger charge anyway.

13

MAJOR

UNTESTED

DOWNGRADED

THRESHOLD FIXED AT 0.5 WITH NO COST MODEL

The word operational in front of an arbitrary threshold

EVIDENCE

Cell 18 computes pred as pb greater than zero point five and prints operational FALSE-POSITIVE RATE at zero point five equals 0.0002, fourteen benign flagged of sixty-two thousand, three hundred and eighty-nine.

SCIENTIFIC CONSEQUENCE

One half is where scikit-learn puts the decision boundary. It is not where a security operations centre puts it, and nothing in this notebook says what a missed tunnel costs relative to a blocked user. Under asymmetric costs the operating point is the result, not a footnote to it. You have named a number operational without doing any of the work the word implies.

REQUIRED REPAIR

State a cost matrix. Choose the threshold that minimises expected cost, or that meets a stated alert budget, and report the false-positive rate and recall at that threshold rather than at one half.

DEFENSIBLE REPLY

0.5 is where I cut the confusion matrix, and the headline is threshold-free. I accept that the word 'operational' is unearned without a cost matrix; I do not accept that it reaches the AUC. And the cost matrix, the field prior and the alerts-per-day are one missing analysis charged three times, as faults 3, 13 and 15.

ORDERED CORRECTION

Merge faults 3, 13 and 15 into a single charge: the notebook states no prior, no cost matrix and no operating point, so nothing it labels 'operational' is interpretable.

14

MAJOR

MEASURED

DOWNGRADED

MODEL RANKING WITHOUT A TEST

A winner declared by one part in a million

EVIDENCE

Cell 16 prints XGBoost at 0.999999 and RandomForest at 0.999998. The table is then sorted and the line best model colon XGBoost is printed, and XGBoost is carried into every downstream cell as the object of study.

SCIENTIFIC CONSEQUENCE

The gap is one in a million on one draw of one split with no interval. Every diagnostic in sections 9 through 12 - the confusion matrix, the importances, the ablation, the cross-validation - is computed for XGBoost because of a difference that has not been shown to exist. That is a selection made by rounding.

REQUIRED REPAIR

McNemar's test on the paired predictions of the top two models, or overlapping bootstrap intervals. If they do not separate, say so and report both.

DEFENSIBLE REPLY

Whichever model the sort returns, the number in my ledger is six nines. What the ordering picks is the object of sections 9 through 12, and I concede that a tie should have been reported as a tie.

Two in ten thousand, multiplied by nothing

EVIDENCE

Cell 18 prints 0.0002. Your literature review in cell 3 cites Sommer and Paxson, 2010, the closed-world critique. Nowhere in the notebook is that rate multiplied by a traffic volume.

SCIENTIFIC CONSEQUENCE

The arithmetic is one line and it changes the verdict. At 0.0002, one million benign DoH flows produce two hundred false alarms. A mid-sized enterprise resolver sees far more than a million DoH flows in a day. So the honest reading of your best number is: this detector generates hundreds of alerts a day on a network where the true positive count may be zero. You cite the paper that makes this argument and you never do the multiplication.

REQUIRED REPAIR

State a daily flow volume for a named deployment context, multiply, and report alerts per analyst per day beside the AUC. Then state whether that figure is acceptable.

DEFENSIBLE REPLY

The multiplication is printed: fourteen benign flows flagged out of sixty-two thousand three hundred and eighty-nine, on the actual held-out set. To turn that into alerts per analyst per day I would have to invent a daily DoH flow volume that appears nowhere in this corpus. Name an enterprise and I will do the arithmetic in one line — but I will not invent a denominator and call it a finding.

ORDERED CORRECTION

Change 'multiplied by nothing' to 'never extrapolated to a named deployment volume', acknowledge the printed absolute count, and merge with faults 3 and 13.

The contamination check is smaller than it looks

EVIDENCE

Cell 20 builds the training key set from Xtr, which cell 16 has already reduced to one hundred and twenty thousand rows, and it slices the test side with a colon fifty thousand. Section 13 discloses the fifty-thousand window and calls the figure a sampled estimate. It does not disclose the second half.

SCIENTIFIC CONSEQUENCE

Contamination measured against a smaller training set is smaller by construction, because there are fewer rows available to match. Your printed 0.014 is therefore an estimate of the contamination of the pipeline you ran, not of the corpus, and the conclusion uses it to argue that the corpus is clean. Those are different claims and the ledger conflates them.

REQUIRED REPAIR

Measure overlap against the entire training partition, on a random sample of the held-out set rather than a contiguous prefix, and report it with a confidence interval.

DEFENSIBLE REPLY

The prefix is not a convenience window. The stratified split permutes the test indices, so the first fifty thousand held-out rows are a simple random sample — which is why my ledger calls it a sampled estimate rather than a full count. On the training side you are right that measuring against 120,000 rows understates overlap against the full partition, but only 1.4 per cent of this corpus is duplicated at all, so the correction is bounded and small.

ORDERED CORRECTION

Delete 'on a random sample of the held-out set rather than a contiguous prefix' from the remedy and state the magnitude bound implied by a 0.014 corpus duplicate rate.

The leakage that matters is not an exact match

EVIDENCE

Cell 20 computes the duplicate rate as one minus X dot drop underscore duplicates over the length of X, and builds the contamination key set from numpy dot round of the array at six decimals. Both figures print at 0.014.

SCIENTIFIC CONSEQUENCE

Twenty-eight continuous timing statistics almost never repeat exactly. So a low exact-duplicate rate on continuous features is close to uninformative, and the notebook rests two of its three trust checks on it. The leakage in a session-structured capture is not identity, it is proximity: a held-out flow that sits at Euclidean distance nought point nought one from a training flow is memorised just as thoroughly as a copy.

REQUIRED REPAIR

Compute the nearest-neighbour distance from each held-out row to the training set and plot the distribution. Report the fraction of held-out rows within a stated small distance of a training row. That is the number the trust grade should have been built on.

DEFENSIBLE REPLY

You are right that twenty-eight continuous statistics rarely repeat exactly and that a low exact-duplicate rate is close to uninformative. But I did not rest on the count: cell 22 de-duplicates the corpus and refits, and the AUC does not move. A nearest-neighbour distribution would be a better test and I should have plotted it; the one I ran is not nothing.

ORDERED CORRECTION

Acknowledge the de-duplicated retrain in cell 22 and charge the trust grade's inputs rather than the notebook's entire treatment of leakage.

One column out of twenty-eight is not a stress test

EVIDENCE

Cell 22 prints three rows all at 0.999999: headline as-is, de-duplicated with one percent of rows removed, and shortcut feature dropped, PacketLengthMean. The conclusion reads: the ablation refutes a single-feature story. So the separability is multi-feature.

SCIENTIFIC CONSEQUENCE

Dropping one column of twenty-eight is a three point six percent perturbation of the feature space. It cannot distinguish a robust detector from a corpus in which twenty different columns each independently separate the classes - and the second reading is what a laboratory generation produces. Section 11 actually states this correctly and then section 13 states the reassuring version.

REQUIRED REPAIR

Greedy or group ablation: remove features in order of importance until the score falls below a stated floor, and report how many columns you had to remove. Also ablate the whole family of packet-length statistics at once, since they are correlated by construction.

DEFENSIBLE REPLY

Section 11 already prints the deflationary reading in full - that flatness on a simulated corpus is itself a generation artifact and not vindication - so the notebook cannot fairly be accused of the misreading it published a paragraph warning against.

19

MAJOR

UNTESTED

DOWNGRADED

NOT COMPARABLE TO THE PUBLISHED PROTOCOL

The dataset authors ran two layers; you ran half of one

EVIDENCE

Cell 3 describes MontazeriShatoori and colleagues, 2020, as the dataset and a two-layer approach: DoH versus non-DoH, then benign versus malicious. This notebook implements only the second layer, on the BCCC re-balanced redistribution named in section 3, and reports 0.999999.

SCIENTIFIC CONSEQUENCE

A doctoral result must be locatable in a literature. Yours is not: it is the second stage of a two-stage pipeline, run in isolation, on a file whose class balance a third party altered, with no comparison to any number the original authors published. There is no reader who can tell whether 0.999999 is better, worse, or the same as prior work.

REQUIRED REPAIR

Run the original two-layer protocol on the authoritative CIRA-CIC distribution, report both stages, and place your figure in the same table as the published one.

DEFENSIBLE REPLY

I marked that table as literature context, not as a reproduced baseline, and I set no number of mine beside a number of theirs. And the first layer of the protocol you want — DoH versus non-DoH — requires non-DoH traffic, which the BCCC redistribution does not contain: my file is fifty-fifty benign and malicious DoH. So this is the same single defect as fault 2, the choice of mirror, arriving for the third time.

ORDERED CORRECTION

Fold into fault 2's Tier 2 mirror-selection charge rather than scoring it as an independent comparability defect.

20

CRITICAL

MEASURED

VERIFIED

THE SELF-ASSIGNED TRUST GRADE IS NOT A MEASUREMENT

An A-grade file produced a six-nines headline

EVIDENCE

Cell 20 assigns the letter by hardcoded cutoffs - F at 0.999, D at 0.99, C at 0.95, B at 0.85 - and prints data trust grade colon A. Section 13 then reads: neither check flags a problem, so nothing here explains the score away.

SCIENTIFIC CONSEQUENCE

Read those two sentences together. The file scored A, and the model scored 0.999999. If an A-grade file yields six nines, then the grade is not measuring what makes a six-nines result implausible. The two statistics behind the letter - single-feature sufficiency and exact-row overlap - are simply not the failure modes operating here; session leakage, tool fingerprinting and a manufactured prior are, and none of the three is in the grade. The letter reassures against risks that were not the risks.

REQUIRED REPAIR

Delete the letter. Report the two underlying quantities with intervals, and add the checks that correspond to the actual threats: grouped-split degradation, leave-one-tool-out degradation, and nearest-neighbour proximity across the split boundary.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 124,777 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So here is the ruling. You built the most careful audit in this series, you wrote down every one of its limits, and you did not spend twenty more lines running the experiments those limits demanded. What you can defend tonight is one sentence: on a re-balanced laboratory file containing three tunnelling tools, under a random split, the classes are separable. That is a real finding and it is not the finding on your title slide. Take the six nines out of the abstract, run the leave-one-tool-out, and come back.

DDoS Detection with Per-Type Recall on CIC-DDoS2019

Four hundred and thirty-one thousand, three hundred and seventy-one CICFlowMeter flows across sixty-five features, seventy-seven percent of them attack traffic, taken from a third party's cleaned re-upload of a testbed capture.

HEADLINE

XGBoost separates DDoS from benign at a held-out ROC-AUC of 0.999997 and an accuracy of 0.999471, with a benign false-positive rate of 0.0011.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 6 critical (FATAL)
- 17 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Two things here are genuinely well done and I want them on the record. You refused to let accuracy carry the story: section 1 says outright that DDoS is volumetric, so aggregate accuracy is easy and misleading, and you print the majority baseline of 0.7732 inside the ranking table so nobody can mistake 0.999471 for an achievement. And your audit found the problem itself - a single column, Packet_Length_Min, reaches 0.9182 alone, and you graded your own corpus B because of it. You saw the flaw. My complaint is about what you did with a corpus you had already diagnosed.

EXAMINER RULING

This notebook establishes, to six decimal places and with a full validity audit, that a flood is voluminous.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Nought point eight nine five out of how many?

EVIDENCE

Cell 18 prints worst per-family recalls colon WebDDoS 0.895, DrDoS_DNS 0.996, Syn 0.999, DrDoS_NTP 0.999, UDP-lag one point zero, DrDoS_UDP one point zero. The code above it filters with if count less than five, continue, and never prints the count or the names of anything dropped. Section 13 reports: the weakest group sits at 0.895, which is where detection is thinnest.

SCIENTIFIC CONSEQUENCE

A recall without a denominator is not a measurement. If WebDDoS has nineteen positives in the held-out set, 0.895 means seventeen of nineteen and the binomial interval spans roughly two-thirds to virtually all. If it has nineteen thousand, it means something quite different. Your own abstract promises that the rare attack classes are visible, not just the dominant floods - and the one number that speaks to rare classes is the one you made impossible to read. Worse, your section 3 provenance table says the label carries eighteen classes, so seventeen attack types plus benign, and cell 18 prints six of them. Eleven attack types are computed and then discarded by a head-of-six slice, unnamed and unreported.

REQUIRED REPAIR

Print every family with its held-out positive count and a binomial confidence interval on its recall. Print, by name, any family excluded by the count-of-five floor, and label it unscorable rather than omitting it.

You audit a duplicate rate someone else set

EVIDENCE

Section 3 states: source Kaggle dhoogla slash cicddos2019, cleaned parquet; rows, roughly four hundred and thirty-one thousand flows, dhoogla deduplicated the raw fifty million plus. Cell 20 then reports exact-duplicate row rate, whole corpus, equals 0.029 and awards contamination grade A.

SCIENTIFIC CONSEQUENCE

Four hundred and thirty-one thousand out of fifty million is under one percent of the capture. Which ninety-nine percent was removed, on what criterion, and with what effect on the class mix, is unknown to you and unknown to me. And then the notebook measures the duplicate rate of that file and treats the low figure as a property of the data. It is not. It is a property of somebody else's deduplication, applied before you saw it. You have audited the output of an unaudited transformation and reported the result as a trust grade.

REQUIRED REPAIR

Obtain the authoritative Canadian Institute for Cybersecurity release, or checksum against it. Report the true corpus size, the transformation applied, and recompute the duplicate and contamination figures on data whose preparation you can describe.

One column already knows the answer

EVIDENCE

Cell 20 prints best single-feature AUC equals 0.9182, feature `Packet_Length_Min`. Cell 16 prints logistic regression - a linear model - at ROC-AUC 0.999328, within seven ten-thousandths of the winning gradient-boosted tree at 0.999997.

SCIENTIFIC CONSEQUENCE

When a single unweighted column reaches nought point nine one eight two, and a linear separator reaches four nines, the two class distributions are not overlapping in any interesting way. That is a statement about how volumetric flood traffic looks in a flow record, not about a detector. The headline 0.999997 is therefore close to uninformative: it tells the reader that a flood is large, which was known before the notebook was written. A doctoral result must be able to fail; this experiment could not have.

REQUIRED REPAIR

Ablate the volumetric block entirely - every packet-count, byte-count, rate and length column - and report the score on behavioural features only. If the remaining columns cannot separate the classes, that is the finding, and it is a far more interesting one than 0.999997.

DEFENSIBLE REPLY

One column at 0.9182 does not know the answer. My own audit scale puts near-sufficiency at 0.99 and grades 0.9182 a B for exactly that reason, and section 10 says in terms that a near-1.0 single-feature AUC is the thing to worry about. In pairwise error, `Packet_Length_Min` is wrong about eight per cent of the time and the fitted model about three ten-thousandths of one per cent. What you actually have is my logistic regression at 0.999328, and that is a much better charge: the classes are close to linearly separable, which is a statement about flow records of floods rather than about a detector. Make that charge and I have no answer.

ORDERED CORRECTION

Lead with logistic regression at 0.999328 and demote 0.9182 to supporting evidence. Delete 'trivially solvable from packet size alone' and 'One column already knows the answer'. Keep the volumetric-block ablation as the remedy — that is the right experiment.

You named the stronger test and ran the weaker one

EVIDENCE

Cell 0 says, in your own words, that Chapter 4 objective 5 separates a one-at-a-time perturbation from the smallest perturbation that reverses a ranking, and warns the first overstates stability - dropping only the top feature and refitting is exactly that weaker test, so read it as a floor. Cell 22 then drops `Packet_Length_Min` and prints 0.999997 unchanged. Nowhere in the notebook is the smallest reversing perturbation computed.

SCIENTIFIC CONSEQUENCE

This is the master fault of this defence in miniature. You named the correct experiment, in your own words, in the first cell of the notebook, described precisely why the one you ran is inferior, and then ran the inferior one and reported nothing else. The stronger test is a loop over the importance ranking - a dozen lines in a cell that already refits the model. I will grant that section 13's narrow sentence, that the separability is multi-feature, does follow from a single-column drop; that is not what I am charging. I am charging that the experiment you told the reader to want was never run.

REQUIRED REPAIR

Run the perturbation you cited: greedy removal of features in importance order until the held-out AUC falls below a stated floor, and report how many of the sixty-five columns you had to remove. Report that count, not the one-column result.

DEFENSIBLE REPLY

You are quoting my own warning back to me as though I had suppressed it. Cell 0 says the one-at-a-time test overstates stability and to read it as a floor; section 11 says a flat result is not vindication and is its own generation artifact. I ran the test I labelled a floor and called it a floor. What survives is that the stronger test was affordable and I did not run it — which is your fault 8, charged once.

ORDERED CORRECTION

Merge with fault 8 as a second instance of the same integrity charge, or drop the 'you named it' framing and charge only the absent greedy ablation.

Eleven ten-thousandths, on a link carrying a flood

EVIDENCE

Cell 18 prints operational FALSE-POSITIVE RATE at zero point five equals 0.0011, twenty-six benign flagged of twenty-four thousand, four hundred and fifty-eight. Section 1 of this notebook states that the operational questions are per-type recall and the false-positive rate on benign traffic, because those decide whether legitimate users get collateral-blocked.

SCIENTIFIC CONSEQUENCE

Do the multiplication you promised. At 0.0011, one million benign flows produce one thousand one hundred false positives. A gateway under DDoS conditions handles far more than a million benign flows an hour, and your stated consequence is that those users get collateral-blocked. So the honest reading of your best operating point is: thousands of legitimate users blocked per hour, during the exact incident when availability is the thing you are defending. You wrote down why the number mattered and then did not compute what it meant.

REQUIRED REPAIR

State a link volume for a named deployment. Multiply. Report legitimate flows blocked per hour, and state whether that is acceptable against the attack traffic it stops.

DEFENSIBLE REPLY

The multiplication is printed: twenty-six benign flows flagged out of twenty-four thousand four hundred and fifty-eight, on the actual held-out set. To turn that into users blocked per hour I would have to invent a gateway throughput that appears nowhere in this corpus, and inventing one is the fabrication this course forbids. Name a deployment and I will do the arithmetic in one line.

ORDERED CORRECTION

Change 'multiplied by nothing' to 'never extrapolated to a named deployment volume', and acknowledge the printed absolute count.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Flows from one attack run land on both sides

EVIDENCE

Cell 16 calls train underscore test underscore split with test size zero point two five, stratified on the label only. Cell 11's drop list contains just the label, y and family - there is no grouping variable of any kind in the pipeline.

SCIENTIFIC CONSEQUENCE

There are seventeen attack types in this corpus and each was generated by running one attack tool against one testbed target for one interval. Splitting the resulting flows at random means the held-out flows come from the same runs as the training flows. The model does not need to learn what a DDoS is; it needs to recognise flows from a run it has already seen most of. Your section 13 sentence - every number above therefore measures in-distribution separability only - is exactly right, and it means the reported score is not evidence for the claim in the title.

REQUIRED REPAIR

GroupShuffleSplit keyed on attack run, or at minimum on attack type, so that whole events are held out. Report the drop.

You never ask whether the capture had a clock

EVIDENCE

Cell 11 loads the cleaned parquet and drops only the label, y and family, leaving sixty-five feature columns. Section 3 describes the source only as dhoogla's cleaned parquet with the raw fifty million plus deduplicated. Section 13 concedes: the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

DDoS traffic is generated by attack runs that begin and end, so capture order is the natural holdout boundary, and your sibling notebook on the same platform drops a timestamp column by name from its CIC-family export. Here you neither use an ordering nor state that none was available. A reader cannot tell whether the temporal experiment was declined or foreclosed, and the notebook does not distinguish those cases anywhere. That is a gap in the provenance record, not merely in the experiment.

REQUIRED REPAIR

State, in section 3, whether the source file carries a timestamp. If it does not, obtain the authoritative Canadian Institute for Cybersecurity release, keep the timestamp for splitting while excluding it as a feature, train on the earlier capture and test on the later one.

DEFENSIBLE REPLY

I have to correct the charge before I answer it, because it is gentler than the facts. You ask whether the ordering survived. It did. The mirror ships seventeen per-attack files, my loader walks them and throws each filename away on the line that reads it; one assign inside that comprehension would have kept the capture key. So the boundary was in my hand and I concatenated across it. As corroboration, no file mixes DrDoS-prefixed attack labels with bare ones — though I will not claim the prefix is itself the key, because bare labels appear in files named -testing too. What I cannot give you is a within-capture clock; no timestamp column exists in the 78-column schema. The honest verdict is worse than the one you wrote, and I would rather you charged it correctly.

ORDERED CORRECTION

Recharge as a Tier 1 destroyed-partition fault rather than a provenance gap: the capture partition was present in the filenames the loader iterates and was concatenated away, and no cell notices. A professor holding this fact could charge it as FATAL alongside fault 6, and should. Do not attach calendar dates or assert which capture is the source release's training day — the mirror's own -training/-testing naming does not track the label vocabulary in the direction one would expect, and the argument does not need it.

8

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

The scope paragraph and the headline in the same section

EVIDENCE

Section 13, verbatim: scope - the split is random, not temporal or entity-grouped. Every number above therefore measures in-distribution separability only. The same section reports 0.999997 and a three-fold cross-validated ROC-AUC of 1.0000.

SCIENTIFIC CONSEQUENCE

Naming a fatal limitation is a confession, not a defence. You know that a grouped split is the experiment that decides your claim; GroupShuffleSplit is twenty lines and it is in the same library you already import in cell 16. The disclosure paragraph cost nothing and bought nothing. The experiment was affordable and was not run.

REQUIRED REPAIR

Run it. Then either report the grouped number as the headline, or withdraw the headline and report the sentence that the evidence supports.

Atlas proof

docs/course/security-ml-notebooks/nb26-cic-ddos2019.ipynb · Notebook cells 16 and 25; zero-based manifest cell IDs: 16=52792534, 25=4265270f

9

MAJOR

UNTESTED

REMECHANISED

NO MODEL SELECTION; THE CROSS-VALIDATION IS NOT THE PIPELINE

Four fixed configurations and a CV on a slice of train

EVIDENCE

Cell 16 hardcodes sixty trees for the random forest, eighty estimators at depth six for XGBoost and eighty for LightGBM. Cell 24 sets cvX equal to Xtr dot iloc, colon forty thousand, and prints a three-fold CV ROC-AUC of 1.0000 plus or minus 0.0000.

SCIENTIFIC CONSEQUENCE

Section 13 offers that 1.0000 as corroboration. It corroborates nothing: it is three folds of forty thousand training rows, never touching the held-out set, on a corpus where a linear model already reaches 0.999328. And because no search was run, the reader cannot separate a chosen model from an arbitrary one. Selection and assessment have been merged into one unsearched pass.

REQUIRED REPAIR

Nested cross-validation - inner loop selects, outer loop estimates - and if the claim is that search is pointless because the problem saturates, demonstrate the saturation with a learning curve rather than asserting it.

DEFENSIBLE REPLY

A fixed-configuration comparison is not a tuning claim, and I never made one. The error you are reaching for is sharper: I selected the winner on the test partition and then quoted the test partition. That is selection on the assessment set, and no hyperparameter search fixes it — a validation split does, at a fraction of the cost.

ORDERED CORRECTION

Recast as selection-on-test with no validation partition; keep the cross-validation-on-a-training-slice observation but attach it to fault 12.

10

MAJOR

DOCUMENTED

VERIFIED

SINGLE SEED FOR THE HEADLINE

One draw decides the ranking and the counts

EVIDENCE

Cell 24 prints seed equals zero.
Cell 16 draws a single split and
cells 18, 20 and 22 all reuse it.

SCIENTIFIC CONSEQUENCE

The margin between XGBoost at 0.999997 and LightGBM at 0.999995 is two parts in a million, and the false-positive count of twenty-six is a small integer. Neither is stable enough to report as a value from one draw.

REQUIRED REPAIR

Ten seeds, mean and standard deviation on the headline AUC and on the false-positive count.

DEFENSIBLE REPLY

The three-fold cross-validation returns a standard deviation of exactly zero, which on a near-separable corpus is fair evidence that additional seeds would move nothing.

11

MAJOR

UNTESTED

REMECHANISED

ROC-AUC AS HEADLINE; THE PR CURVE IS DRAWN AND NEVER READ

You plotted the right diagnostic and declined to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and plots it in the third panel. There is no call to average precision score anywhere in the notebook. The ranking table in cell 16 is sorted on roc underscore auc.

SCIENTIFIC CONSEQUENCE

This is not an oversight of omission; you built the diagnostic and ranked on something else. And on this corpus the choice matters twice over: the positive class is seventy-seven percent of the file, so the ROC has almost no negatives to be wrong about, and any deployment inverts that ratio completely. Average precision is the quantity that survives the inversion, and you had the curve on screen.

REQUIRED REPAIR

Compute average precision, print it, rank on it. One line in a cell that already has the curve.

DEFENSIBLE REPLY

Both halves of that sentence are backwards. Having few negatives makes FPR more sensitive to a false positive, because the denominator is small; and precision is a function of the base rate, so average precision does not survive an inversion of prevalence — ROC-AUC does. An average precision computed at seventy-seven per cent positives would tell a deployed sensor at one per cent positives nothing.

ORDERED CORRECTION

Re-mechanise: the precision-recall panel is drawn with the majority class as the positive class, so it says nothing about the rare class, and no benign-as-positive operating point is established anywhere in the notebook.

12

MAJOR

UNTESTED

DOWNGRADED

NO UNCERTAINTY ANYWHERE

Six decimals, zero intervals

EVIDENCE

Cell 16 reports 0.999997 as a point. Cell 18 reports 0.0011 resting on twenty-six misclassified flows, and WebDDoS recall at 0.895 with no count. Cell 20 reports contamination 0.021 from a fifty-thousand-row window. Cell 24 reports a standard deviation of 0.0000.

SCIENTIFIC CONSEQUENCE

Twenty-six events is a small number and 0.895 may rest on a handful. The entire validity ledger consists of point estimates compared against hard thresholds, with no statement of how far any of them could move. A standard deviation of exactly zero across three folds is not stability; on a near-separable corpus it is what a degenerate estimator returns.

REQUIRED REPAIR

Bootstrap the held-out AUC with a thousand resamples. Put exact binomial intervals on the false-positive rate, on the contamination estimate, and on every per-type recall.

DEFENSIBLE REPLY

The interval that decides anything in this notebook is the one on WebDDoS, and you have already charged that as fault 3. Charging it again as a general absence of intervals scores one defect twice.

ORDERED CORRECTION

Strike 'A standard deviation of exactly zero across three folds' — the line formats with ``:.4f``, so any spread below 5e-5 renders as 0.0000 and nothing establishes exactness. Merge the per-type interval requirement into fault 3.

13

MAJOR

MEASURED

VERIFIED

FALSE PRECISION

Six decimals on a quantity with no interval

EVIDENCE

Cell 16 rounds each ROC-AUC to six places; the table prints 0.999997, 0.999995, 0.999912 and 0.999328. Cell 22 prints the whole ablation column at 0.999997.

SCIENTIFIC CONSEQUENCE

Reporting to a resolution finer than the sampling spread invites the reader to compare digits that are noise. And section 13 does exactly that when it says the ablation barely moves the AUC from 0.999997 to 0.999997 - two numbers that are identical to the printed precision being offered as an experimental result.

REQUIRED REPAIR

Report to the resolution the estimator supports, which means reporting an interval, not more digits.

DEFENSIBLE REPLY

The comment in cell 16 states the reason for six places explicitly - a 1.000000 is a red flag, not a win - so the extra digits exist to expose saturation rather than to rank, which is a defensible use.

14

MAJOR

UNTESTED

DOWNGRADED

THRESHOLD FIXED AT 0.5, NO COST MODEL

The word operational in front of a library default

EVIDENCE

Cell 18 forms pred as pb greater than zero point five and prints operational FALSE-POSITIVE RATE at zero point five.

SCIENTIFIC CONSEQUENCE

One half is where the library puts the boundary. In DDoS response the costs are wildly asymmetric and directional: a missed flood degrades a service, a false positive blocks a paying customer during an outage. Nowhere does this notebook state the exchange rate between those two. Under asymmetric costs, the operating point is the result, and yours was chosen by a default argument.

REQUIRED REPAIR

State a cost matrix. Select the threshold that minimises expected cost or that meets a stated alert budget, and report recall and false-positive rate at that point.

DEFENSIBLE REPLY

0.5 is where I cut the confusion matrix, and the headline is a threshold-free ranking statistic that no choice of threshold moves. I accept that the word 'operational' is unearned without a cost matrix. I do not accept that it touches the AUC.

ORDERED CORRECTION

Merge with fault 5 into one charge: the notebook never selects an operating point and never converts one into a deployment quantity.

15

MAJOR

MEASURED

DOWNGRADED

MODEL RANKING WITHOUT A TEST

A winner chosen by two parts in a million

EVIDENCE

Cell 16 prints XGBoost at 0.999997 and LightGBM at 0.999995, then prints best model colon XGBoost and carries that object into every downstream cell.

SCIENTIFIC CONSEQUENCE

Every diagnostic in sections 9 through 12 - confusion matrix, importances, ablation, cross-validation - is computed for XGBoost on the strength of a two-in-a-million difference measured once. That is a selection made by rounding, and it silently determines which model the rest of the notebook is about.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two, or overlapping bootstrap intervals. If they do not separate, say so and carry both forward.

DEFENSIBLE REPLY

Whichever of the two the sort returns, the reported figure is six nines. What the ordering selects is the object of sections 9 through 12, not the number in the ledger, and I concede that should have been reported as a tie.

The overlap check is smaller than it looks

EVIDENCE

Cell 20 builds the training key set from Xtr, which cell 16 has already cut to one hundred and twenty thousand rows, and slices the test side with colon fifty thousand. Section 13 discloses the fifty-thousand window; it does not disclose the training-side truncation.

SCIENTIFIC CONSEQUENCE

Overlap measured against a smaller training set is smaller by construction. Your 0.021 therefore describes the pipeline you ran, and section 13 uses it to argue about the corpus - saying train-test overlap is negligible, so neither issue explains the result. Those are different claims, and the trust grade rests on the conflation.

REQUIRED REPAIR

Measure overlap against the entire training partition, on a random sample of held-out rows rather than a contiguous prefix, and report it with an interval.

DEFENSIBLE REPLY

Two answers. The prefix is not a convenience window: the stratified split permutes the test indices, so the first fifty thousand held-out rows are a simple random sample, which is why section 13 calls it a sampled estimate. And the training-side point, which I accept as a reasoning error, cannot move 0.021 far when only 2.9 per cent of the corpus is duplicated at all. Contrast nb27, where the identical code path understates a 64.8 per cent duplicate corpus by a factor of ten — there it is fatal, here it is a footnote.

ORDERED CORRECTION

Delete 'on a random sample of held-out rows rather than a contiguous prefix' from the remedy, and state the magnitude: at a 0.029 corpus duplicate rate the correction is bounded and small.

The leakage that matters is proximity, not identity

EVIDENCE

Cell 20 computes the duplicate rate via drop underscore duplicates and builds the contamination keys from numpy dot round at six decimals, printing 0.029 and 0.021.

SCIENTIFIC CONSEQUENCE

A volumetric flood emits enormous numbers of flows that are similar but not identical. Exact matching is precisely the wrong test for that structure, and it is the test on which two of your three trust checks rest. A held-out flow at negligible distance from a training flow is memorised as thoroughly as a copy, and your audit reports it as clean.

REQUIRED REPAIR

Compute nearest-neighbour distances from held-out rows to the training set and report the distribution, plus the fraction of held-out rows within a stated small distance of a training row.

DEFENSIBLE REPLY

Exact matching is the weaker test and I accept the criticism of the trust grade's inputs. But I did not only count duplicates: I removed them and refit, and the score did not move. That is a direct test of the memorisation story, not an inference from a match count. A nearest-neighbour distribution would be better and I should have plotted it.

18

CRITICAL

UNTESTED

VERIFIED

NO ADVERSARIAL EVALUATION

A security detector with no adversary in the experiment

EVIDENCE

Cells 0, 1, 2 and 3 contain no adversarial evaluation, no perturbation budget and no evasion experiment. The related-work table in cell 3 lists only base-rate and false-positive caveats. Cells 10 through 24 contain no perturbation of any held-out example.

SCIENTIFIC CONSEQUENCE

Your top feature is `Packet_Length_Min`. An attacker who pads the smallest packets in the flood changes that column at essentially zero cost to the attack - the flood still floods. Because no adversarial experiment exists, the notebook has no worst-case number at all, and for a security detector the worst case is the result. Note that this axis is not merely unexecuted here; unlike its sibling notebooks, it is not even named.

REQUIRED REPAIR

State a perturbation budget - pad packets to a minimum size, jitter inter-arrival times - apply it to held-out attack flows only, and report recall after perturbation.

19

MAJOR

UNTESTED

DOWNGRADED

NOT COMPARABLE TO THE PUBLISHED PROTOCOL

You cite their results and never run their protocol

EVIDENCE

Cell 3 lists Sharafaldin and colleagues, 2019, with the caveat entry ID3, random forest and naive Bayes on CICFlowMeter features. Cell 11 loads a third party's cleaned parquet, and cell 16 evaluates on a fresh random split.

SCIENTIFIC CONSEQUENCE

A doctoral number must be locatable in a literature. Yours is computed on a re-upload of under one percent of the capture, under a split of your own construction, with no baseline from the source paper reproduced. No reader can say whether 0.999997 is better than, worse than, or identical to what has already been published.

REQUIRED REPAIR

Run the published partition on the authoritative release and report your figure in the same table as theirs.

DEFENSIBLE REPLY

That table is marked as literature context in the cell you are quoting, and I put no number of mine beside a number of theirs. The half of this that lands — that the file I evaluated on is a third party's reduction — is your fault 4.

ORDERED CORRECTION

Fold the surviving half into fault 4 and drop the 'not locatable in a literature' framing.

You graded the corpus B and printed six nines anyway

EVIDENCE

Cell 20 assigns the letter by hardcoded thresholds - F at 0.999, D at 0.99, C at 0.95, B at 0.85 - and prints data trust grade colon B, driven by the single-feature AUC of 0.9182. Section 13 then reports 0.999997 as the result.

SCIENTIFIC CONSEQUENCE

The grade of B is the notebook telling itself that one column nearly solves the problem. The correct response to that finding is to question whether the task is a task. Instead, the letter is reported as a caveat and the headline is reported as a finding, and the two sit four lines apart in the same ledger. A letter that can coexist with six nines is not constraining anything.

REQUIRED REPAIR

Delete the letter. Report the single-feature AUC with an interval, and let the reader draw the conclusion that 0.9182 from one column already implies.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 107,843 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. The ruling is this. You audited your corpus honestly, you found that one column carries most of the answer, you graded your own data B for that reason, and then you published six nines as the result. What this notebook actually demonstrates is that flood traffic is voluminous, which the field established twenty years ago. Ablate the volumetric block, run the grouped split, print the per-type positive counts, and there may be a paper here. As submitted, there is a tautology with a validity audit attached to it.

IoT Botnet Detection on N-BaIoT (Mirai / BASHLITE Device Behaviour)

Just over seven million damped-window behaviour snapshots across one hundred and fifteen features, drawn from nine commercial IoT devices infected in a laboratory with Mirai and BASHLITE.

HEADLINE

A random forest separates infected from benign device behaviour at a held-out ROC-AUC of 1.000000 and an accuracy of 0.999865, on a corpus the notebook grades C for trustworthiness.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self C

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 1 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 22 live faults
- 6 critical (FATAL)
- 16 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me give you the strongest thing in this notebook first, because it is genuinely rare. You reported a duplicate rate of sixty-four point eight percent - a number most authors would have quietly not computed - and you then re-ran the entire experiment with those rows removed and published the result. You graded your own corpus C. Section 1 and the loader comment both name cross-device transfer as the real question. Nobody here is hiding anything. That is precisely why the ruling is severe: you knew.

EXAMINER RULING

There are nine devices in this corpus, you split it at random, and so the most economical explanation of a perfect score is that the model learned which device it was looking at.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Nine devices, and you split by row

EVIDENCE

Section 3 states nine devices. Cell 11 splits each filename on the dot and assigns parts index zero to nothing - it computes the family from parts one onward and discards the device number. Cell 16 then calls `train underscore test underscore split` at test size zero point two five, stratified on the label alone, and cell 16 prints ROC-AUC 1.000000.

SCIENTIFIC CONSEQUENCE

This is not a missing column. Your loader parses the device identifier out of the filename and drops it on the same line that keeps the family. So the grouping key was in your hands at the moment of the split. With nine devices and one hundred and fifteen behavioural statistics, the model has a trivially available strategy: recognise the device from its idle traffic profile, then recall whether that device's snapshots are mostly infected. Given that ninety-two point one three percent of the corpus is attack traffic, that strategy alone gets you very close to 1.000000. You cannot distinguish it from detection, and nothing in the notebook tries to.

REQUIRED REPAIR

`GroupKFold` or `GroupShuffleSplit` keyed on the device number that cell 11 already parses. Train on some devices, test on the held-out devices, rotate, and report the mean. That is the number the abstract is claiming and it is twenty lines away.

Atlas proof

`docs/course/security-ml-notebooks/nb27-nbaiot-iot-botnet.ipynb` · cells 18 and 25; zero-based manifest cell IDs: 18=33efe013, 25=248da676

Sixty-four point eight percent duplicates, six point six percent contamination

EVIDENCE

Cell 20 prints exact-duplicate row rate, whole corpus, equals 0.648, and train-test exact-row contamination equals 0.066. Cell 20 builds its training key set from Xtr, which cell 16 has already reduced to one hundred and twenty thousand rows. Cell 16 prints materialized seven million and sixty-two thousand, six hundred and six rows, trained on one hundred and twenty thousand, held-out one million, seven hundred and sixty-five thousand, six hundred and fifty-two.

SCIENTIFIC CONSEQUENCE

Here is the arithmetic a student can check live. One hundred and twenty thousand divided by seven million and sixty-two thousand is one point seven percent. Contamination is the chance that a held-out row exactly matches some training row, and that chance rises monotonically with the number of training rows available to match against. You measured it against one point seven percent of the file. When sixty-four point eight percent of the file consists of exact duplicate rows, the low contamination figure is not evidence that the corpus is clean - it is an artifact of how few rows you looked in. And section 13 uses that figure to award a contamination sub-grade of B and to argue that overlap is not the issue here.

REQUIRED REPAIR

Recompute overlap against the full five point three million row training partition, on a random sample of held-out rows rather than the first fifty thousand, and report it with an interval. Then re-state the trust grade against that number.

3

CRITICAL

UNTESTED

VERIFIED

A SERIAL PROCESS SPLIT AS THOUGH IT WERE INDEPENDENT DRAWS

ORAL TOP 5

Damped windows overlap, so the split is not a holdout

EVIDENCE

Your abstract calls them behaviour snapshots - damped-window traffic statistics. Cell 11 repeats the phrase. Cell 20 prints an exact-duplicate rate of 0.648 across just over seven million rows, and cell 22 labels the de-duplicated ablation as sixty-five percent rows removed - roughly four point six million rows.

SCIENTIFIC CONSEQUENCE

A damped window is by definition a decaying summary of history. Two snapshots emitted milliseconds apart summarise almost the same packets. The sixty-four point eight percent exact-duplicate rate is the direct arithmetic evidence of this: it is what you get when a process is sampled far faster than it changes. Randomly splitting a serial process does not produce a holdout, it produces two interleaved views of one recording. Every number downstream of cell 16 inherits that defect, and no cell in the notebook measures it.

REQUIRED REPAIR

Split on capture order within each device: train on the earlier portion of each capture, test on the later portion, with a guard band between them wider than the longest decay window. Report the drop from 1.000000.

5

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

ORAL TOP 5

The words flag but do not run, in your own source code

EVIDENCE

Cell 11, verbatim in the source: per-device transfer, train on some devices, test on others, is the harder question we flag but do not run. Section 3 repeats it: cross-device transfer, not the pooled split, is the real test. Section 13 then reports ROC-AUC 1.000000.

SCIENTIFIC CONSEQUENCE

You wrote the sentence flag but do not run into the source of the notebook. That is not a limitation section, that is a decision recorded at the moment it was made, in the same cell that already parses the grouping key out of the filename. At doctoral level, naming a fatal limitation is a confession, not a defence. The experiment was affordable, it was identified, it was declined, and the number was published anyway.

REQUIRED REPAIR

Run leave-one-device-out across all nine devices. Report the nine scores and their mean. Then decide which number belongs in the abstract.

Every family at one point zero, none of them by device

EVIDENCE

Cell 18 prints worst per-family recalls colon gafgyt dot junk one point zero, gafgyt dot udp one point zero, gafgyt dot tcp one point zero, gafgyt dot combo one point zero, gafgyt dot scan one point zero, mirai dot syn one point zero. Six entries, all one point zero. Section 1 states: the deployment question is whether such a model transfers across devices, which pooling alone cannot answer.

SCIENTIFIC CONSEQUENCE

Your own conclusion draws the correct inference and then stops: every group listed is recovered essentially perfectly, so there is no rare-class failure to report on this split - that uniformity suggests the corpus is easy to separate, not that the detector is strong. Correct. And then the notebook prints nothing else. The grouping that would have been informative was the device, it was available in the filename, and swapping the grouping variable in cell 18 is a one-line change. You ran the uninformative version of the diagnostic and left the informative one unwritten.

REQUIRED REPAIR

Print recall per device, with the held-out positive count beside each, in the same panel. If one device is recovered at 0.6 while the pooled figure reads 1.000000, that is the paper.

DEFENSIBLE REPLY

You are asking me to swap a grouping variable inside a split that already contains every device. That panel would read nine bars at 1.0 and would tell the reader nothing. The only version of this diagnostic that carries information is leave-one-device-out, which is your fault 5. The residual charge — that an advertised honesty check produced six saturated bars — I accept.

ORDERED CORRECTION

Replace the required remedy with: report per-device recall under leave-one-device-out, not under the pooled split, and state plainly that the pooled per-device panel is uninformative by construction.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Seven million in the title, one hundred and twenty thousand in the fit

EVIDENCE

The subtitle of the notebook reads: model comparison plus per-botnet-family recall plus validity audit, seven million and sixty-two thousand, six hundred and six behaviour snapshots. Cell 16 prints trained on one hundred and twenty thousand, stratified subsample. One hundred and twenty thousand over seven million and sixty-two thousand, six hundred and six is one point seven percent.

SCIENTIFIC CONSEQUENCE

The seven-million figure describes the download, not the experiment. And on this corpus the gap is not cosmetic, because sixty-four point eight percent of those seven million rows are exact duplicates, so the effective distinct sample is smaller still. A reader who takes the subtitle at face value believes a claim was established at a scale at which it was never tested.

REQUIRED REPAIR

Put the effective training n in the abstract. Add a learning curve showing the score against training size, so the reader can see whether one hundred and twenty thousand rows was saturation or convenience.

DEFENSIBLE REPLY

Cell 16 prints materialized, trained on and held-out on a single line, so the subsample is disclosed at the point of use and in the same breath as the corpus size - no reader working through the notebook in order can be misled.

Two models tied at one point zero zero zero zero zero zero

EVIDENCE

Cell 16 prints RandomForest at roc underscore auc 1.000000 and XGBoost at roc underscore auc 1.000000. The table is sorted and the notebook prints best model colon RandomForest, and every diagnostic in sections 9 through 12 is then computed for the random forest.

SCIENTIFIC CONSEQUENCE

There is no difference between the top two entries. The winner was determined by the order in which pandas performed a stable sort, and that arbitrary outcome selected the model whose confusion matrix, importances, ablation and cross-validation the reader is shown. A model was chosen by tie-break and reported as best. The code comment in the same cell even anticipates this - six decimal places, a 1.000000 is a red flag, not a win - and the pipeline treats it as a win regardless.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the two tied models, or bootstrap intervals on both. Report a tie as a tie and carry both models forward.

DEFENSIBLE REPLY

The number you are attacking is identical for both models. Whichever one the sort returns, the reported ROC-AUC is 1.000000. What the tie-break selects is the object of the downstream diagnostics, not the headline, and I concede that should have been reported as a tie.

ORDERED CORRECTION

'the order in which pandas performed a stable sort' is wrong — DataFrame.sort_values defaults to quicksort, which is not stable, so the tie-break is implementation-defined rather than insertion-ordered.

The linear model already solves it

EVIDENCE

Cell 16 prints LogisticRegression at ROC-AUC 0.999799 and accuracy 0.999529 - four nines from a scaled linear separator. Cell 20 prints a single-feature AUC of 0.9665 from HH underscore jit underscore L0 underscore 01 underscore mean.

SCIENTIFIC CONSEQUENCE

When a linear model reaches four nines and one column of one hundred and fifteen reaches 0.9665, the two class distributions are effectively disjoint in the feature space. A botnet flooding from a doorbell produces packet rates orders of magnitude above the doorbell's idle behaviour, in a lab where nothing else is happening. That is a property of the capture, not a detection capability. Your own section 13 says as much - that uniformity suggests the corpus is easy to separate, not that the detector is strong - and then the abstract keeps the perfect score.

REQUIRED REPAIR

Restrict the feature set to statistics an attacker cannot trivially inflate, refit, and report. Or state in the abstract that the measured quantity is the separability of laboratory flood traffic from laboratory idle traffic.

One column out of a hundred and fifteen

EVIDENCE

Cell 22 prints three rows, all at one point zero: headline as-is, de-duplicated with sixty-five percent of rows removed, and shortcut feature dropped, HH underscore jit underscore L0 underscore 01 underscore mean. Section 13 reads: the ablation refutes a single-feature story, so the separability is multi-feature.

SCIENTIFIC CONSEQUENCE

Removing one of one hundred and fifteen columns perturbs zero point eight seven percent of the feature space, and these columns are damped-window statistics computed over multiple decay factors, so they are correlated by construction. The test could not have moved the score. Reporting that it did not is not a finding.

REQUIRED REPAIR

Greedy ablation in importance order until the score falls below a stated floor, plus a block ablation removing every column that shares a statistic family, so correlated substitutes go together.

DEFENSIBLE REPLY

The notebook only claims the ablation refutes a single-feature story, and for that narrow claim a one-column drop is exactly the right test; section 11 separately warns that flatness is a generation artifact and not vindication.

You plotted the right diagnostic and declined to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and draws it in the third panel of the first figure. There is no call to average precision score anywhere in the notebook, and cell 16 sorts the ranking table on roc underscore auc.

SCIENTIFIC CONSEQUENCE

You built the diagnostic and ranked on something else. On this corpus the choice is decisive in a way it is not elsewhere: the positive class is ninety-two percent of the file, so the ROC curve has almost no negatives to be wrong about. A deployed IoT monitor sees the inverse - overwhelmingly benign snapshots - and average precision is the quantity that survives that inversion. The curve was on your screen.

REQUIRED REPAIR

Compute average precision, print it, and rank on it. One import in a cell that already imports the curve.

DEFENSIBLE REPLY

Two things in that sentence are backwards. Having almost no negatives makes the false-positive rate MORE sensitive per error, not less — the denominator is small. And average precision does not survive a change of prevalence; precision is a function of the base rate and ROC-AUC is not. Ranking on average precision computed at 92% positives would tell a deployed IoT monitor at 0.1% positives nothing at all.

ORDERED CORRECTION

Re-mechanise to the true defect: the precision-recall panel is drawn with the majority class as the positive class, so it is uninformative about the rare class, and no benign-as-positive operating point — precision, recall or alert budget for the class that is actually scarce — is established anywhere in the notebook.

The word operational in front of a library default

EVIDENCE

Cell 18 forms pred as pb greater than zero point five and prints operational FALSE-POSITIVE RATE at zero point five equals 0.0002, twenty-nine benign flagged of one hundred and thirty-eight thousand, nine hundred and eighty-three.

SCIENTIFIC CONSEQUENCE

One half is where the library puts the boundary. In IoT defence the costs are asymmetric and specific: a missed infection enlists a device into a botnet, a false positive quarantines a working camera or doorbell. Nothing in this notebook states that exchange rate. Under asymmetric costs the operating point is the result, and yours was inherited from a default argument.

REQUIRED REPAIR

State a cost matrix, choose the threshold that minimises expected cost or meets a quarantine budget, and report recall and false-positive rate at that point.

DEFENSIBLE REPLY

0.5 is where the confusion matrix was cut, and the headline is a threshold-free ranking statistic that no choice of threshold would change. I accept that the word 'operational' is unearned without a cost matrix; I do not accept that it invalidates the AUC.

ORDERED CORRECTION

Merge with fault 13 into one charge: the notebook never selects an operating point and never converts one into a deployment quantity.

Two in ten thousand, multiplied by nothing

EVIDENCE

Cell 18 prints 0.0002. Your literature review in cell 3 cites Sommer and Paxson, 2010, the closed-world critique. No cell multiplies that rate by a snapshot rate or a fleet size.

SCIENTIFIC CONSEQUENCE

One line of arithmetic changes the verdict. At 0.0002, one million benign snapshots produce two hundred false alarms. These are damped-window snapshots emitted continuously by each device, so a fleet of a few hundred devices generates millions of snapshots per day, which is thousands of quarantine alerts a day against a true infection count that is normally zero. You cite the paper that makes this argument and never perform the multiplication.

REQUIRED REPAIR

State a snapshot emission rate and a fleet size, multiply, and report alerts per analyst per day beside the AUC. Then say whether that figure is deployable.

DEFENSIBLE REPLY

I printed the numerator and the denominator on the actual held-out set. To give you alerts per analyst per day I would have to invent a snapshot emission rate and a fleet size that appear nowhere in this corpus, and inventing them is the fabrication this course forbids. Name the deployment and I will do the arithmetic.

ORDERED CORRECTION

Change 'multiplied by nothing' to 'never extrapolated to a named deployment volume', and acknowledge the printed absolute count.

Four and a half million rows removed, and no disagreement rate reported

EVIDENCE

Cell 22 computes Xdd as X dot drop underscore duplicates and takes y at Xdd dot index, and labels the row de-duplicated, sixty-five percent rows removed - roughly four point six million of just over seven million. Section 13 discloses the rule: the de-duplication variant keeps the first label when a feature vector appears twice.

SCIENTIFIC CONSEQUENCE

Where one feature vector carries two different labels, that disagreement is irreducible Bayes error - a floor on any achievable accuracy on this feature set. Silently keeping the first label deletes that floor from view. On a corpus where four and a half million rows were dropped, the disagreement rate is a substantial quantity that was computed away rather than measured.

REQUIRED REPAIR

Group by the feature vector, count groups carrying both labels, and report that rate as the noise floor. State the maximum accuracy attainable given it.

DEFENSIBLE REPLY

With one hundred and fifteen continuous statistics, an exact duplicate almost certainly comes from the same device and the same capture phase, so genuine label disagreement is likely to be near zero and the omission likely costs nothing.

Atlas proof

docs/course/security-ml-notebooks/nb27-nbaiot-iot-botnet.ipynb · cells 22 and 25;
zero-based manifest cell IDs: 22=57032322, 25=248da676

Nine of them were infected, or some of them

EVIDENCE

The abstract in cell 1 reads: behaviour snapshots from nine commercial IoT devices. Nine of them were infected in the lab with the Mirai and BASHLITE botnets. The comment at the top of cell 11 reads: nine commercial IoT devices, some infected with the Mirai and BASHLITE, gafgyt, botnets.

SCIENTIFIC CONSEQUENCE

Those two sentences cannot both be true, and the difference is not editorial. If some devices carry only one botnet family, or none, then device identity is partially confounded with the label, and a random split converts that confound directly into score. The notebook never prints a device-by-family table, so the reader cannot resolve the contradiction or assess the confound. On a corpus where I am already arguing the model may be identifying devices, this is the one table that was needed and it is absent.

REQUIRED REPAIR

Print a device-by-family cross-tabulation with row counts, in section 7, before any model is fitted. Reconcile the abstract to it.

DEFENSIBLE REPLY

You are right that the two sentences cannot both be true, and I will tell you which is false rather than make you guess: the download directory contains no mirai captures for devices 3 and 7, so all nine devices carry BASHLITE and only seven carry Mirai. The abstract sentence is the wrong one and the loader comment is the correct one. The cross-tabulation you ask for should be in section 7 and is not.

ORDERED CORRECTION

Replace 'Those two sentences cannot both be true' with the resolved fact — devices 3 and 7 ship benign plus five gafgyt captures and no mirai captures — and charge the abstract with the error directly.

Gini importance across a hundred and fifteen correlated columns

EVIDENCE

Cell 18 takes best dot feature underscore importances underscore and plots the top twelve under the title top importances slash absolute coefficient.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward high-cardinality and high-variance columns and it distributes credit arbitrarily among correlated features. This feature set is one hundred and fifteen damped-window statistics computed over shared decay factors, which is close to the worst case for that estimator. The panel therefore cannot support any statement about mechanism, and it is the only mechanism evidence the notebook offers.

REQUIRED REPAIR

Permutation importance computed on the held-out set, or SHAP values, with the correlated blocks permuted together rather than one column at a time.

DEFENSIBLE REPLY

Impurity importance on 115 correlated damped-window statistics is a bad estimator and I do not defend it. But point me at the sentence it supports. Section 13 contains no statement about which features the model uses. The panel is decorative, and a decorative panel is a presentation fault, not a validity fault.

ORDERED CORRECTION

Re-charge as: an unlabelled impurity-importance panel is printed with no caveat on a feature set that is close to the worst case for that estimator, inviting a mechanistic reading the notebook never earns.

Meidan built one model per device; you pooled

EVIDENCE

Cell 3 describes Meidan and colleagues, 2018, as providing the dataset and a per-device autoencoder detector, and the caveats table in the same cell contrasts it with supervised classifiers on pooled devices, our setting, noting that pooling can hide poor cross-device transfer.

SCIENTIFIC CONSEQUENCE

You wrote the sentence pooling can hide poor cross-device transfer into your own literature review, chose pooling, and reported 1.000000. The published result and yours are not on the same axis - theirs is per-device and unsupervised, yours is pooled and supervised - so no reader can place your number beside the literature. And the caveat you recorded predicts precisely the failure mode you did not test.

REQUIRED REPAIR

Report a per-device evaluation alongside the pooled one, so the pooled figure can be read against the protocol the corpus was built for.

DEFENSIBLE REPLY

I marked that table as literature context, not as reproduced measurement, in the cell you are quoting. Meidan's detector is unsupervised and per-device; mine is supervised and pooled; I never placed the two numbers side by side and never invited a reader to. The part of your charge that lands — that my own recorded caveat predicts the failure mode I did not test — is fault 5.

ORDERED CORRECTION

Fold the surviving half into fault 5 and drop the 'not locatable in a literature' framing.

You graded the corpus C and printed a perfect score

EVIDENCE

Cell 20 assigns the letter by hardcoded thresholds - F at 0.999, D at 0.99, C at 0.95, B at 0.85 - and prints data trust grade colon C. Section 13 reports ROC-AUC 1.000000 four lines earlier.

SCIENTIFIC CONSEQUENCE

A C is the notebook telling itself that a single column reaches 0.9665 and that overlap is not negligible. The correct response to a C is to withhold the headline pending a defensible protocol. Instead the letter is filed as a caveat and the perfect score is filed as a result. A grade that can coexist with 1.000000 is constraining nothing, and the cutoffs it uses - single-feature sufficiency and exact-row overlap - are not the failure modes operating on this corpus, which are device confounding and window overlap.

REQUIRED REPAIR

Delete the letter. Report the two underlying quantities with intervals, and add the checks that match the real threats: leave-one-device-out degradation and nearest-neighbour proximity across the split boundary.

Sixty trees, chosen by nobody

EVIDENCE

Cell 16 hardcodes a random forest of sixty estimators, XGBoost at eighty estimators and depth six, and LightGBM at eighty. Cell 24 runs three-fold cross-validation on Xtr dot iloc, colon forty thousand.

SCIENTIFIC CONSEQUENCE

The winning model's capacity was set by a literal in the source and never varied, so no reader can distinguish a chosen configuration from an arbitrary one. And the cross-validated 1.0000 that section 13 offers as corroboration is computed on forty thousand rows drawn from the training side only, never touching the held-out set. Selection and assessment have been collapsed into a single unsearched pass.

REQUIRED REPAIR

Nested cross-validation: an inner loop that selects, an outer loop that estimates. If the claim is that the problem saturates, show the learning curve that demonstrates it.

DEFENSIBLE REPLY

I never claimed sixty trees was optimal, and a model comparison with fixed configurations is not a tuning claim. The methodological error you are reaching for is real and more specific than the one you charged: I picked the winner on the test set and then quoted the test set. That is selection on the assessment partition, and no amount of hyperparameter search would fix it.

ORDERED CORRECTION

Recast as: model selection is performed on the held-out partition and the selected model's held-out score is reported as an unbiased estimate; there is no validation split. The cross-validation-on-a-training-slice observation belongs with fault 11.

One draw of the split, one tie-break, one winner

EVIDENCE

Cell 24 prints seed equals zero.
Cell 16 draws one split at that seed, and cells 18, 20, 22 and 24 all reuse it.

SCIENTIFIC CONSEQUENCE

The top two models are tied to six decimals, so which one becomes the object of every downstream diagnostic is a function of one draw. The false-positive count of twenty-nine is a small integer reported without a spread. None of these is stable enough to state as a value from a single realisation.

REQUIRED REPAIR

Ten seeds, and report mean and standard deviation for the headline AUC, the false-positive count, and the identity of the winning model.

DEFENSIBLE REPLY

On a corpus where a linear model already reaches four nines, ten seeds would print ten identical numbers, and the notebook's zero-variance cross-validation is reasonable evidence of that.

Atlas proof

docs/course/security-ml-notebooks/nb27-nbaiot-iot-botnet.ipynb · cell 25, Scope paragraph; zero-based manifest cell IDs: 25=248da676

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here `RandomForest`. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 1,765,652 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So the ruling. There are nine devices in this corpus, the device identifier passes through your loader and is discarded on the same line that keeps the label, and you split by row anyway. You wrote the words flag but do not run into your own source code. Until leave-one-device-out is on the page, one point zero zero zero zero zero zero zero is not a detection result - it is the strongest available evidence that the model can tell a doorbell from a thermostat.

Real E-Commerce Card-Fraud Detection on IEEE-CIS (Vesta)

Five hundred and ninety thousand, five hundred and forty real Vesta e-commerce transactions across three hundred and ninety-one features, three and a half percent of them fraudulent, most of the features anonymised.

HEADLINE

XGBoost separates fraud from legitimate transactions at a held-out ROC-AUC of 0.919380, on a corpus the notebook grades A for trustworthiness.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self A

D1 1 External validity	D2 2 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 3 Reproducibility	D6 1 Honesty / scope

23 live faults

6 critical (FATAL)

17 major (CONTESTED)

0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be generous here, because this notebook earns it and the others do not. Your single-feature audit tops out at 0.6882 - the second lowest figure across the entire series - which means no column is quietly carrying the answer. Your headline is 0.919380 rather than six nines, and you say plainly in section 13 that fraud is three and a half percent so accuracy is meaningless. Your cross-validation returns a real spread, plus or minus 0.0114, rather than the degenerate zero its siblings print. And your section 9 does something I have not seen elsewhere in this series: it names the weakest stratum, not the aggregate, as the operational result. This is real work on real production data, and I am rejecting it on grounds that have nothing to do with the quality of the audit.

EXAMINER RULING

This is the most honest notebook of the four in front of me, and it fails for the single most avoidable reason available: the transactions are ordered in time, the ordering column was in your hands, you dropped it as an identifier, and you let the model see the future.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

You dropped the clock and called it an identifier

EVIDENCE

Cell 11 contains this line, verbatim in the comment: drop the label, the row id and the raw time offset, identifiers. The drop list is isFraud, y, family, TransactionID, TransactionDT. Cell 16 then calls train underscore test underscore split at test size zero point two five, stratified on the label only, and reports 0.919380.

SCIENTIFIC CONSEQUENCE

TransactionDT is the transaction time offset. Dropping it as a feature is defensible. Dropping it as an experimental design variable is a different decision, and one line of code made both at once - the comment even records the reasoning, that it is an identifier. The consequence is that your training set contains transactions that occurred after your test transactions. A fraud model that has seen the future is not a fraud model. And you do not have to take that from me: your section 13 cites Dal Pozzolo and colleagues, 2018, precisely for the finding that concept drift dominates real card fraud, and then reports a number from a split that cannot see drift at all.

REQUIRED REPAIR

Sort by TransactionDT, train on the earlier portion, test on the later portion, exclude it as a feature exactly as you already do, and report the drop from 0.919380. That is one sort and one slice, and it is the number the title claims.

The card is a feature and never a grouping key

EVIDENCE

The comment at the top of cell 11 lists the feature families as card, address and email fields, counting features, timedelta features, match features and anonymised Vesta engineered features. None of them appears in the drop list, and cell 16 stratifies on the label alone.

SCIENTIFIC CONSEQUENCE

One compromised card produces a burst of fraudulent transactions. When those transactions are scattered at random, the model can memorise the card identifier from the training rows and recognise it in the test rows. That is not fraud detection, it is card lookup. And because the card fields also serve as features, you cannot even bound how much of the 0.919380 comes from that route - the notebook contains no experiment that separates recognising a known compromised card from detecting a fraudulent transaction.

REQUIRED REPAIR

GroupShuffleSplit keyed on the card identifier, and separately on the billing address, so a card never appears on both sides. Report the pooled and the grouped scores together and let the gap be the finding.

Three hundred and thirty-nine columns nobody can name

EVIDENCE

Section 3 states: many features are anonymized, V1 through V339, so we can audit whether a single feature carries the score, but not always name the mechanism. Cell 20 reports the strongest single feature as C4 at AUC 0.6882. Cell 18 plots the top twelve importances by name.

SCIENTIFIC CONSEQUENCE

You have written the correct sentence and then let the notebook behave as though it were not true. An importance panel over anonymised columns names nothing. The ablation drops C4 and reports 0.917591, which tells the reader that C4 is not necessary and nothing at all about what replaced it. Construct validity is the axis on which a security or fraud claim lives - does the model measure the adversary's behaviour or the data collector's bookkeeping - and on this corpus that axis is not merely unmeasured, it is unmeasurable. That is a fatal limit on what the notebook can conclude, and the conclusion does not carry it.

REQUIRED REPAIR

State in the abstract that no mechanistic claim is available. Then do the one thing that is available: partition the features into the named block - card, address, email, transaction amount, product code - and the anonymised V block, fit on each separately, and report both scores. If the named block alone reaches 0.9, you have a result you can interpret. If only the V block does, you have a result nobody can.

DEFENSIBLE REPLY

I wrote that we can audit whether a single feature carries the score but not always name the mechanism, and then I made no mechanism claim. Quote me the sentence that names what the model keys on. The charge that lands is narrower and I accept it: the abstract promises to audit whether the score is an artifact of the anonymised features, and the named-block-versus-V-block fit that would have answered that was never run.

ORDERED CORRECTION

Re-charge on the abstract's unmet promise — 'audit whether the score is genuine fraud signal or an artifact of the anonymized features' — and on the unrun named-versus-V partition, not on a mechanism claim the notebook declines to make.

Nought point nine one nine on top, nought point one seven seven underneath

EVIDENCE

Cell 16 reports ROC-AUC 0.919380. Cell 18 prints worst per-family recalls colon W, 0.177; H, 0.455; S, 0.47; R, 0.636; C, 0.677. Section 9 states: the product carrying the most fraud is also the one detected worst.

SCIENTIFIC CONSEQUENCE

Read those together. In the stratum that carries the most fraud in this corpus, this model misses roughly four fraudulent transactions in five. The aggregate ROC-AUC is 0.919380 because ranking most legitimate transactions below most fraudulent ones is easy; catching the fraud is not. And the notebook never computes a single summary quantity that would expose this: no average precision, no precision at the operating point, no recall at a stated false-positive budget. What it computes is ROC-AUC, and ROC-AUC is what the abstract, the subtitle and the ranking table carry, while 0.177 appears once, inside a printed dictionary in cell 18. Section 9 says the right thing in prose; every position where a reader looks for the result says 0.919380.

REQUIRED REPAIR

Put recall at a stated false-positive budget, per product code, in the abstract. Retire ROC-AUC from the headline position on a three and a half percent base rate, and replace it with average precision plus per-stratum recall.

DEFENSIBLE REPLY

Take the substance first, because you are right about it: this notebook computes no average precision, no precision at any operating point, and no recall at any false-positive budget, and it should have. Now the positioning. Read cell 1. There is no number in my title, my subtitle or my abstract. And section 13 says, of the 0.177, that this gap and not the aggregate score is the operationally important result. You are charging me with burying a figure that my conclusion promotes.

ORDERED CORRECTION

Delete 'the abstract, the subtitle' and '0.177 appears once, inside a printed dictionary in cell 18'. Recharge as: the notebook computes no prevalence-aware summary quantity anywhere, so the per-stratum figure it correctly promotes has no companion metric a reader can act on.

Atlas proof

docs/course/security-ml-notebooks/nb28-ieee-cis-fraud.ipynb · Code cell 18 output; Markdown cell 25 validity ledger; zero-based manifest cell IDs: 18=cec7d47f, 25=64a3bf22

Your own literature review specifies the protocol you did not use

EVIDENCE

Cell 3, in the caveats table, describes the IEEE-CIS competition leaders using gradient boosting with, in your own words, heavy feature engineering plus time-aware validation, and notes the leaderboard is AUC on a hidden split. Cell 11 loads only train underscore transaction dot csv, and cell 16 evaluates on a fresh random quarter.

SCIENTIFIC CONSEQUENCE

You wrote down the evaluation protocol that the entire published literature on this dataset uses, in your own literature review, and then used a different one without comment. 0.919380 therefore sits beside nothing. It cannot be said to be better, worse, or equal to any leaderboard figure, any published baseline, or any prior model. And the difference between the two protocols is exactly the difference this defence is about - time-aware versus random.

REQUIRED REPAIR

Reproduce the competition protocol: hold out the final portion of the transaction timeline, report AUC there, and place your figure in the same table as a published result.

DEFENSIBLE REPLY

I labelled that table as literature context, not as a reproduced baseline, in the same cell you are quoting, and I placed no figure of mine beside a leaderboard number. What you have found is that I recorded the right protocol and used the wrong one, which is your fault 1, charged twice.

ORDERED CORRECTION

Fold into fault 1 as supporting evidence rather than scoring it as an independent defect.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

6

CRITICAL

DOCUMENTED

VERIFIED

DISCLOSURE SUBSTITUTED FOR MITIGATION

Flagged but not run, written twice

EVIDENCE

Section 3 reads: the honest generalization test is time-ordered, which we flag but do not run. Section 13 reads: the limit a random split cannot show is concept drift, and a time-ordered evaluation, not run here, is the deployment-realistic test, citing Dal Pozzolo and colleagues, 2018. Section 13 then reports 0.919380.

SCIENTIFIC CONSEQUENCE

Naming a fatal limitation is a confession, not a defence. You identified the experiment, cited the paper that establishes why it matters, wrote the phrase flag but do not run, and shipped the number. The sort is one line. TransactionDT was in the dataframe when you dropped it. The experiment was affordable, identified, and declined.

REQUIRED REPAIR

Run the time-ordered split, report the degraded number as the headline, and keep the random-split figure only as a labelled upper bound.

7

MAJOR

DOCUMENTED

VERIFIED

TRANSDUCTIVE PREPROCESSING, AND A DISCLOSURE THAT IS FALSE ON THIS CORPUS

The no-op sentence is not true here

EVIDENCE

Cell 11 fits LabelEncoder on the full feature frame before the split. Cell 11's own comment states that ProductCD stays as a feature, and lists card, address and email fields among the columns retained. Section 13 states: on an all-numeric corpus that step is a no-op.

SCIENTIFIC CONSEQUENCE

This corpus is not all-numeric. ProductCD takes the string values W, C, H, R and S - you say so yourself in the provenance table - and the card, email-domain and match columns are categorical too. So the encoder actually fires on this notebook, and the sentence offered to excuse it is boilerplate carried over from a numeric corpus where it was true. It is a false statement sitting in the validity ledger of a notebook whose entire purpose is the validity ledger, and a reader who trusts the ledger is misinformed about what the pipeline did.

REQUIRED REPAIR

Delete the sentence or condition it on an actual check of the column types. Fit the encoder inside the training fold and route unseen categories to an explicit unknown bucket, which a deployed fraud system will need on its first day.

DEFENSIBLE REPLY

The encoding never consults the label, so nothing leaks in the sense that matters, and the notebook already flags the step as transductive and names the unseen-category bucket a deployment would need - the erroneous clause is a copy-paste slip, not a defect in the experiment.

You named the stronger test and ran the weaker one

EVIDENCE

Cell 0 states that Chapter 4 objective 5 separates a one-at-a-time perturbation from the smallest perturbation that reverses a ranking, warns the first overstates stability, and says dropping only the top feature and refitting is exactly that weaker test, so read it as a floor. Cell 22 drops C4 and prints 0.919380 falling to 0.917591. Nowhere in the notebook is the smallest reversing perturbation computed.

SCIENTIFIC CONSEQUENCE

You named the correct experiment in the first cell, explained why the one you ran is inferior, ran the inferior one, and reported nothing else. Dropping one column of three hundred and ninety-one perturbs about a quarter of one percent of the feature space, and on anonymised, heavily redundant engineered features it could not plausibly have moved anything. I will grant that the narrow sentence in section 13, that no single column is load-bearing, does follow from what you ran; the charge is that the experiment your own opening cell told the reader to want was never performed.

REQUIRED REPAIR

Greedy removal in importance order until the held-out AUC falls below a stated floor, and report how many of the three hundred and ninety-one columns had to go. Report that count in place of the single-column result.

DEFENSIBLE REPLY

I told the reader in cell 0 that this is the weaker test and that it overstates stability, and I told them again in section 11 that a flat result is not vindication. You are charging me with running the experiment I labelled as a floor and calling it a floor. What survives is that the stronger test was affordable and not run, and that is your fault 6.

ORDERED CORRECTION

Merge with fault 6 as a second instance of the same integrity charge, or drop the 'named the stronger test' framing and charge only the absent greedy ablation.

Nought point eight seven seven nine versus nought point nine one nine four

EVIDENCE

Cell 24 prints XGBoost three-fold CV ROC-AUC equals 0.8779 plus or minus 0.0114. Cell 16 prints held-out ROC-AUC 0.919380. The difference is 0.919380 minus 0.8779, which is nought point nought four one five - about three and a half times the printed cross-validation standard deviation.

SCIENTIFIC CONSEQUENCE

You have two estimates of the same quantity that disagree by more than three standard deviations of the one that carries a spread, and section 13 reports both without remarking on the gap. Whenever a resampled estimate falls well below a single-split estimate, the single split is the number under suspicion, not the resample. That discrepancy should have been the subject of a paragraph and instead it is two entries in a ledger.

REQUIRED REPAIR

Reconcile them. Report a bootstrap interval on the held-out AUC, and state whether it contains 0.8779. If it does not, the split, not the model, is the finding.

DEFENSIBLE REPLY

Look at what the cross-validation is fitted on. `cvX` is `Xtr.iloc[:40_000]`, and three folds means each model sees about twenty-six thousand seven hundred rows. The headline model saw one hundred and twenty thousand — roughly four and a half times as much. Of course the smaller fit scores lower; that is a learning curve, not a warning. Your inference — that whenever a resample falls below a single split the split is under suspicion — requires the two estimates to be of the same estimator, and here they are not.

ORDERED CORRECTION

Recharge as: two estimates of different pipelines are placed side by side in the ledger with no statement that they are not comparable, so a reader is invited to read a training-size artifact as agreement or disagreement about the same quantity.

One half is the worst possible threshold on a three-and-a-half-percent base rate

EVIDENCE

Cell 18 forms pred as pb greater than zero point five and prints the operational false-positive rate at that threshold. Section 1 of this notebook states that the operative trade-off is catching fraud without swamping analysts or blocking legitimate customers.

SCIENTIFIC CONSEQUENCE

At a three and a half percent positive rate, a threshold of one half sits far out on the tail and the result is exactly what cell 18 shows: recall of 0.177 on the largest stratum. That is not a property of the model, it is a property of the threshold. You named the trade-off in section 1 as the operative question of the notebook and then never varied the one parameter that controls it. Under asymmetric costs the operating point is the result.

REQUIRED REPAIR

State a cost matrix - what a missed fraud costs against what a blocked customer costs - and choose the threshold that minimises expected cost. Report recall and false-positive rate at that point, per product code.

Seven thousandths, declared a winner

EVIDENCE

Cell 16 prints XGBoost at 0.919380 and LightGBM at 0.911913, a gap of nought point nought zero seven five. The table is sorted, best model colon XGBoost is printed, and XGBoost carries into every downstream cell.

SCIENTIFIC CONSEQUENCE

Unlike its sibling notebooks the margin here is not absurd, which is exactly why the test was owed and would have been informative. Your own cross-validation carries a standard deviation of 0.0114 - larger than the margin between the top two models. So on the evidence in this notebook, XGBoost and LightGBM are indistinguishable, and the choice that determined every diagnostic in sections 9 through 12 was made by sorting a column.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, or overlapping bootstrap intervals. If they do not separate, say so and report both.

DEFENSIBLE REPLY

The 0.0114 you are using is the fold-to-fold spread of a model fitted on twenty-six thousand rows. It is not the standard error of a held-out AUC on one hundred and forty-seven thousand rows, which is about three thousandths — so the seven-and-a-half-thousandth gap is close to three standard errors, and a paired McNemar test would be more powerful still. Run your own test and I expect it to separate them. I accept that I should have run it.

ORDERED CORRECTION

Delete 'Your own cross-validation carries a standard deviation of 0.0114 - larger than the margin' and 'on the evidence in this notebook, XGBoost and LightGBM are indistinguishable'. Charge only the absence of the paired test.

Three and a half percent positives, and every learner left at its default

EVIDENCE

Cell 16 constructs logistic regression with max iter three hundred, a random forest of sixty trees, XGBoost with eighty estimators at depth six, and LightGBM with eighty estimators. Not one carries class underscore weight or scale underscore pos underscore weight. Cell 11 prints fraud rate 0.0350.

SCIENTIFIC CONSEQUENCE

This is the standard remedy for the exact condition the notebook names in its first sentence of section 1, and it is one keyword argument per model. Its absence is a substantial part of why recall on the largest product stratum is 0.177: the loss function was never told that a missed fraud matters more than a false alarm. The accuracy column shows the consequence - every model in the table sits within one and a third points of the majority baseline of 0.9650.

REQUIRED REPAIR

Set scale underscore pos underscore weight to the negative-to-positive ratio for the boosted models and class underscore weight balanced for the others, refit, and report the change in per-stratum recall.

DEFENSIBLE REPLY

scale_pos_weight would have moved the confusion matrix and left the ranking essentially where it is, because ROC-AUC is invariant to a monotone rescaling of scores. What you have found is that I never chose an operating point, which is your fault 10. Charging the missing keyword argument separately double-counts one decision.

ORDERED CORRECTION

Merge with fault 10, and drop 'Its absence is a substantial part of why recall on the largest product stratum is 0.177' — the threshold is.

Twenty-four ten-thousandths, multiplied by nothing

EVIDENCE

Cell 18 prints operational FALSE-POSITIVE RATE at zero point five equals 0.0024, three hundred and forty benign flagged of one hundred and forty-two thousand, four hundred and sixty-nine.

Section 1 states the trade-off is catching fraud without swamping analysts or blocking legitimate customers.

SCIENTIFIC CONSEQUENCE

The arithmetic is one line. At 0.0024, one million legitimate transactions produce two thousand four hundred blocked customers. Against that, the model recovers seventeen point seven percent of the fraud in the largest product stratum. So on the evidence in this notebook, the system blocks thousands of paying customers per million transactions in order to catch fewer than one fraud in five where fraud is most common. That sentence is the operational result and it appears nowhere in the notebook, even though every number needed to write it is printed.

REQUIRED REPAIR

State a transaction volume, multiply, and report blocked legitimate customers per day beside fraud recovered per day. Then state whether that exchange is acceptable.

DEFENSIBLE REPLY

The multiplication you want is printed: three hundred and forty legitimate transactions blocked out of one hundred and forty-two thousand four hundred and sixty-nine, on the actual held-out set. To convert that into customers blocked per day I would have to invent a daily volume that appears nowhere in this corpus. Name one and I will do the arithmetic.

ORDERED CORRECTION

Change 'multiplied by nothing' to 'never extrapolated to a named deployment volume', and acknowledge the printed absolute counts.

The overlap check is smaller than it looks

EVIDENCE

Cell 20 builds the training key set from Xtr, which cell 16 has already reduced to one hundred and twenty thousand rows, and slices the test side at colon fifty thousand of one hundred and forty-seven thousand, six hundred and thirty-five. Section 13 discloses the fifty-thousand window and calls the figure a sampled estimate; it does not disclose the training-side truncation.

SCIENTIFIC CONSEQUENCE

Overlap measured against fewer training rows is smaller by construction, because fewer rows are available to match. Section 13 nonetheless reads the 0.002 as a property of the data - train-test exact-row overlap 0.002 scores A, neither check flags a problem. That is a claim about the corpus derived from a measurement of one pipeline configuration, and the trust grade that the conclusion leans on rests on the conflation.

REQUIRED REPAIR

Measure overlap against the entire training partition, on a random sample of held-out rows, and report it with an interval.

DEFENSIBLE REPLY

The whole-corpus duplicate rate is independently printed at 0.003, which bounds how large the true contamination could be, so on this particular corpus the truncation cannot have hidden anything material.

You never say what makes a transaction fraudulent

EVIDENCE

Section 3's provenance table gives the label row as: isFraud, zero slash one, roughly three point five percent fraud. Cell 11 sets y equal to isFraud as integer. Nothing anywhere in cells 0 through 26 describes the labelling process.

SCIENTIFIC CONSEQUENCE

In card fraud generally, a zero in the fraud column means not reported as fraud, which is not the same as verified legitimate. Fraud surfaces through chargebacks and customer complaints, which arrive well after the transaction, so the recent tail of any transaction file tends to contain fraud still labelled as legitimate. That has two consequences the notebook cannot address because it never raises the issue: your negative class is contaminated at an unknown rate, and the contamination is time-dependent, which interacts directly with the temporal split you did not run. A doctoral fraud claim requires the provenance of its label, and yours is a column name.

REQUIRED REPAIR

State the labelling process, the reporting lag, and the resulting censoring at the end of the observation window. Then either exclude the censored tail or report the sensitivity of the headline to it.

DEFENSIBLE REPLY

I accept that a provenance table that gives the label as a column name and a rate is not provenance. But your consequence — that the negative class is time-dependently contaminated — is asserted from what card fraud usually looks like, not from anything about this dataset that you have put in front of me. Bring the publisher's labelling statement and this becomes unanswerable; without it, it is a gap in my record rather than a demonstrated defect in my label.

ORDERED CORRECTION

Cite the competition host's published labelling rule for isFraud, which propagates the fraud label across transactions linked by card, email or billing address. If that is on the record, this fault becomes FATAL and also converts fault 2 from a good-practice charge into a demonstrated label leak through retained feature columns. Do not assert it without the citation.

Nought point one seven seven out of how many?

EVIDENCE

Cell 18 prints the five recalls - W 0.177, H 0.455, S 0.47, R 0.636, C 0.677 - and the loop above filters with if count less than five, continue, without printing any count.

SCIENTIFIC CONSEQUENCE

Section 9 says the weakest stratum, not the aggregate, is the operational result, and then presents that result without the denominator needed to read it. Recall of 0.455 on the H stratum may rest on a few dozen positives or on several thousand, and the binomial interval differs by an order of magnitude between those cases. The notebook makes the strongest claim available about per-stratum failure and withholds the number that would let a reader test it.

REQUIRED REPAIR

Print each product code with its held-out positive count and a binomial confidence interval on its recall.

DEFENSIBLE REPLY

The denominators are two thousand two hundred and fifty for W, one thousand nine hundred and eighty-seven for C, three hundred and ninety-one for H, three hundred and fifty-seven for R and one hundred and eighty-one for S. The number my section 9 calls the operational result rests on two thousand two hundred and fifty positives and is precise to within about one and a half points. I should have printed them, and the omission changes no reading of the result.

ORDERED CORRECTION

Drop 'the binomial interval differs by an order of magnitude between those cases' — it does not here; the range across the five strata is roughly plus or minus 0.016 to plus or minus 0.073. Keep the charge as a presentation defect.

A personal re-upload, and no checksum

EVIDENCE

Section 3 gives the source as Kaggle, l-n-a-s-i-r-i-zero-zero-seven slash ieecis-fraud-detection, described as a Vesta data mirror. Cell 11 downloads that dataset and reads train underscore transaction dot csv with no hash comparison.

SCIENTIFIC CONSEQUENCE

Every number in this notebook is a measurement of whatever that account uploaded. You cite Vesta Corporation, 2019, in the abstract for a file you obtained from a third party, and you have no evidence the two are the same object. Row counts matching a remembered figure is not verification. On a corpus where you are making claims about real production fraud, the identity of the file is part of the claim.

REQUIRED REPAIR

Checksum against the official IEEE-CIS competition distribution, or pull from it directly. Print the hash in the provenance table.

DEFENSIBLE REPLY

Cell 11 asserts a row floor and prints five hundred and ninety thousand five hundred and forty transactions, three hundred and ninety-one features and a fraud rate of three and a half percent, all three of which match the published IEEE-CIS training transaction file. That is three independent checks, not a remembered figure. A checksum would be strictly better and I should have printed one; the absence of one is not the absence of evidence.

ORDERED CORRECTION

Change 'you have no evidence the two are the same object' to 'you have three coincident statistics and no cryptographic identity', and keep the hash requirement.

Your literature review says tuning wins, and you did none

EVIDENCE

Cell 16 hardcodes eighty estimators at depth six for XGBoost, eighty for LightGBM and sixty trees for the random forest. Cell 24 runs three folds on Xtr dot iloc, colon forty thousand. Cell 3's caveats table describes the competition leaders as using heavy feature engineering plus time-aware validation.

SCIENTIFIC CONSEQUENCE

You recorded, in your own literature review, that performance on this exact corpus is driven by feature engineering and validation design, and then submitted four untuned defaults on a random split. Unlike the near-separable network corpora elsewhere in this series, this problem is not saturated - your headline is 0.919380, not 1.000000 - so search would actually move the number, and the reader has no way to know by how much. Selection and assessment have been collapsed into one unsearched pass.

REQUIRED REPAIR

Nested cross-validation with an inner search over depth, learning rate and estimator count. Report the outer-loop estimate as the headline.

DEFENSIBLE REPLY

I never claimed eighty estimators at depth six was tuned, and a fixed-configuration comparison is not a tuning claim. The error you are reaching for is more specific and worse: I chose the winner on the test set and then quoted the test set. Nested cross-validation would fix that; so would a validation split, at a fraction of the cost.

ORDERED CORRECTION

Recast as selection-on-test with no validation partition. Keep the observation that this corpus is unsaturated, which makes the missing search consequential in a way it is not elsewhere in the series.

19

MAJOR

DOCUMENTED

VERIFIED

SINGLE SEED FOR THE HEADLINE

One draw decides a seven-thousandth margin

EVIDENCE

Cell 24 prints seed equals zero.
Cell 16 draws one split at that seed and cells 18, 20, 22 and 24 reuse it.

SCIENTIFIC CONSEQUENCE

The gap between the top two models is nought point nought zero seven five and the printed cross-validation spread is 0.0114, so a second draw could plausibly reorder them. And the per-product recalls - 0.177, 0.455, 0.47 - are the notebook's headline operational claims, reported from a single realisation with no spread.

REQUIRED REPAIR

Ten seeds. Report mean and standard deviation for the headline AUC, for each per-product recall, and for the identity of the winning model.

DEFENSIBLE REPLY

The three-fold cross-validation in cell 24 already supplies a spread of plus or minus 0.0114 for the winning model, which is a legitimate if partial substitute for a seed sweep.

20

CRITICAL

MEASURED

VERIFIED

THE SELF-ASSIGNED TRUST GRADE IS NOT A MEASUREMENT

An A that certifies the file and says nothing about the claim

EVIDENCE

Cell 20 assigns the letter by hardcoded thresholds - F at 0.999, D at 0.99, C at 0.95, B at 0.85 - and prints data trust grade colon A. Section 13 reads: neither check flags a problem, so nothing here explains the score away.

SCIENTIFIC CONSEQUENCE

Here the A is deserved, and that is precisely the point I want to make. The single-feature AUC really is 0.6882 and the exact-row overlap really is 0.002. And none of that touches the three defects that decide this notebook: a random split on time-ordered data, no card grouping, and a feature space nobody can interpret. So an A-grade file coexists with a claim that does not survive contact with the deployment protocol. The letter measures two failure modes that were not operating and is then read as clearing the result. That is the structural fault - the grade is orthogonal to the finding.

REQUIRED REPAIR

Delete the letter. Report the two quantities with intervals, and add the checks that match the real threats here: the degradation under a temporal split, the degradation under card grouping, and the named-versus-anonymised feature partition.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 147,635 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result. This notebook is where the charge bites hardest: its printed duplicate rate is essentially zero, so de-duplication changed almost nothing about the corpus — and the score moved anyway. That movement can only be the split.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

Closing. So the ruling, and I want to be precise about it. Of the four notebooks I am reading tonight, this is the best one - the honest headline, the low single-feature AUC, the real cross-validation spread, and section 9's insistence that the weakest stratum is the result. And it fails on the one line of code that would have cost you nothing: you sorted nothing, you dropped the clock as an identifier, and you let the model read tomorrow's fraud before it was asked about today's. Sort by `TransactionDT`, group by card, and bring me the number that comes out. I expect it to be considerably worse than 0.919380, and I expect it to be publishable.

System-Log Anomaly Detection on HDFS (Loghub)

575,061 HDFS block sessions, each a count vector over 28 varying log-event templates, labelled Success or Fail, with an anomaly rate of 2.93 percent.

HEADLINE

RandomForest reaches a held-out ROC-AUC of 0.999983 and an accuracy of 0.999833 on HDFS console-log anomaly detection.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

9.5

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 1 Decision quality	D5 1 Reproducibility	D6 1 Honesty / scope

- 20 live faults
- 5 critical (FATAL)
- 15 major (CONTESTED)
- 3 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me start with what this notebook did that most of the field does not. It measured its own train-test contamination, found it at 0.999, and printed an F on itself in the same cell. It then went further and actually re-ran the experiment with the duplicates removed, and it printed the number that came back. Section 3 refuses credit it has not earned, saying plainly that no log parser runs in this pipeline and that Drain is a method reference, not an attribution. And section 12 tells you, unprompted, that the cross-validation number describes a forty-thousand-row slice of a subsample and not the corpus. That is more honesty than I see in most submitted papers.

EXAMINER RULING

Ninety-nine point nine percent of your held-out set is a verbatim copy of a training row, so the number you put in your abstract is not a detection score, it is the hit rate of a lookup table.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The held-out set is the training set

EVIDENCE

The validity-audit cell, cell 20, prints the exact-duplicate row rate for the whole corpus at 0.999 and then prints, on its own line, train-test exact-row contamination equals 0.999. In words: ninety-nine point nine percent of the held-out rows the audit checked are byte-identical to a row the model was trained on. The independent audit probe measured the same thing on a randomly drawn window and got 0.9989, so this is a property of the corpus and not an artifact of which window was sampled.

SCIENTIFIC CONSEQUENCE

A held-out score answers one question: what does the model do on data it has not seen. At a contamination of 0.999, the model has seen essentially all of it. The reported 0.999983 is therefore a measurement of memorization, and it would be produced just as well by a hash table keyed on the twenty-eight count values. Every downstream number inherits the defect. The per-type recalls, the false-positive rate, and the confusion matrix are all computed on the same contaminated held-out set, so not one of them is an out-of-sample quantity.

REQUIRED REPAIR

De-duplicate the corpus to distinct count vectors before the split, not after, and assign every copy of a vector to one side of the split only. Then report the held-out score on genuinely unseen vectors. That is the number this notebook owed and did not print.

The corrected corpus has a few hundred rows

EVIDENCE

Do the arithmetic with me from the printed figures. Cell 20 prints a whole-corpus duplicate rate of 0.999. One minus 0.999 is 0.001. Times 575,061 rows is about five hundred and seventy-five distinct count vectors. Because 0.999 is printed to three decimals it could be anywhere from 0.9985 to 0.9995, so the true figure lies somewhere between roughly two hundred and ninety and eight hundred and sixty. On any rounding of the number this notebook printed, the de-duplicated corpus is under a thousand rows.

SCIENTIFIC CONSEQUENCE

The de-duplicated ablation is the only corrective run in the notebook, and it was performed on a few hundred rows split three-to-one. That means it fitted on roughly four hundred vectors and scored on roughly a hundred and fifty. A score of 0.9836 estimated on that many rows carries an enormous sampling spread, and none is reported. So the notebook's one honest number is also its least precise, and the reader is given no way to know that. Worse, the ablation table labels this row de-duplicated, one hundred percent rows removed, which is a rounding of 0.999 and reads to a student as though nothing was left.

REQUIRED REPAIR

Print the row count of the de-duplicated corpus beside the ablation score, attach a bootstrap confidence interval to the 0.9836, and if the effective sample size cannot support the claim, say that the corpus does not contain enough distinct sessions to support a supervised benchmark at all.

Forty percent of anomaly types were never scored

EVIDENCE

The results cell contains the line, if count is less than five, continue. The conclusion concedes the mechanism and then says, and I quote, it never prints how many types that removes. The independent audit probe ran the count for us. Of twenty anomaly-type groups, eight fell below the floor and were skipped. That is forty percent of the anomaly taxonomy, and the number appears nowhere in the notebook.

SCIENTIFIC CONSEQUENCE

The section-13 conclusion nominates per-type recall as the metric that matters, because with three percent anomalies accuracy is inflated. It then reports six recalls, the worst of which is 0.995, and it is those six that a reader carries away. But the eight unscored types are exactly the rare ones, and rare is where a log anomaly detector earns its keep. A coverage claim built on the twelve easy types and silent about the eight hard ones is not a coverage claim. The notebook says the printed recalls cannot be read as full coverage, which is correct, and then prints no number that would let anyone bound the gap.

REQUIRED REPAIR

Print every anomaly type with its held-out positive count next to its recall, and where the count is too small to score, print the word unscorable with the count. Never let a class leave the table without a line.

You plotted the right curve and refused to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and plots it in the third panel. Nowhere in this notebook is average precision computed, printed, or used to rank a model. That pattern holds across the whole series: every one of the thirty-six notebooks draws this curve and not one computes its area.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to the absolute count of false positives when negatives dominate. At a 2.93 percent positive rate, an enormous change in the analyst's workload moves ROC-AUC in the fifth decimal, which is exactly the regime this headline lives in. Average precision would have shown that movement. You built the instrument, put it on the page, and then ranked your models on the metric that cannot see the thing you said mattered.

REQUIRED REPAIR

Report average precision, that is, the area under that curve you already plotted, with a bootstrap interval, and report precision at a fixed operationally chosen recall. Rank the four learners on that.

Not one interval on any reported number

EVIDENCE

The ledger reports the single-feature AUC at 0.6896, the contamination at 0.999, the false-positive rate at 0.0001, and six per-type recalls, none with an interval. The only dispersion figure in the notebook is the cross-validation standard deviation of plus or minus 0.0006, and that is computed on a forty-thousand-row head slice of the training subsample, not on the reported held-out estimate.

SCIENTIFIC CONSEQUENCE

Consider the recall of 0.995 that the conclusion calls the thinnest measured detection. The held-out set contains 4,210 positives in total, which you can check yourself: 143,766 held out minus the 139,556 benign in the false-positive line gives 4,210. Those 4,210 are spread across twelve scored types, and the notebook prints six of those twelve, with no denominator for any of them. Look at what the printed values themselves tell you: a recall of 0.995 cannot come from a small group, it needs on the order of two hundred positives to be expressible at all. So the six recalls the notebook chose to show are precisely the ones whose values imply large denominators, and the six it did not show, along with the eight types the floor removed entirely, are where the small counts must live. No reader can tell which is which, because not one positive count appears beside a recall anywhere in this notebook.

REQUIRED REPAIR

Attach a bootstrap confidence interval to the held-out AUC and a Wilson binomial interval to every per-type recall, and print the positive count that each interval is based on.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

First label wins, and nobody counted the conflicts

EVIDENCE

Cell 22 constructs the de-duplicated corpus as X dot drop underscore duplicates, then indexes the labels by the surviving index. The conclusion discloses this in the words, the de-duplication variant keeps the first label when a feature vector appears twice. With a duplicate rate of 0.999 across twenty-eight small integer count columns, the number of vectors carrying both a Success and a Fail label is the single most informative statistic about this task, and it is not printed.

SCIENTIFIC CONSEQUENCE

If a count vector appears as both normal and anomalous, then no classifier operating on counts alone can separate those two cases, ever. That disagreement rate is a hard ceiling on achievable accuracy, a Bayes error floor, and it is the number that would tell you whether this representation can support the task. Deleting it by keeping the first label does not remove the ambiguity from the world, it removes it from your evaluation, and it makes the corrected score look better than the corrected protocol deserves.

REQUIRED REPAIR

Group the corpus by the feature tuple, count the groups whose labels are not unanimous, print that rate as the noise floor, and report the corrected score against that ceiling rather than against one.

DEFENSIBLE REPLY

You are right that I did not print the disagreement rate, and I should have. But you called it 'the single most informative statistic about this task' and predicated the charge on it being large. It is not. Nine of my 589 distinct vectors are label-ambiguous. They cover 550 rows, one tenth of one percent of the corpus, and the minority label inside them accounts for 36 rows. That is an irreducible error floor of 6.3 times ten to the minus five. My reported error rate is 1.67 times ten to the minus four. The floor you say I deleted sits below the error I actually make, so it cannot be the ceiling that explains my number, and 'keeping the first label' moved roughly two rows in a 148-row corrective test set. The thing that limits my corrected 0.9836 is the sample size you correctly attack in charge 4, not the label conflicts you attack here.

ORDERED CORRECTION

Strike 'the single most informative statistic about this task' and 'it makes the corrected score look better than the corrected protocol deserves' — both are refuted by the computed value. Keep the reporting demand: the disagreement rate is 9/589 groups and should be printed as the noise floor even though it is small, because the reader cannot know it is small without it.

Atlas proof

docs/course/security-ml-notebooks/nb29-hdfs-log-anomaly.ipynb · Code cell 22 and markdown cell 25; zero-based manifest cell IDs: 22=d0017f8a, 25=80147cc0

Three hundred thousand rows were never used

EVIDENCE

Cell 16 prints, materialized 575,061 rows, trained on 120,000 as a stratified subsample, held-out 143,766. Add the two numbers the model actually saw: 120,000 plus 143,766 is 263,766. Subtract from 575,061 and 311,295 rows were loaded, parsed, and then dropped on the floor.

SCIENTIFIC CONSEQUENCE

The five hundred and seventy-five thousand figure describes the download, not the study. It appears in the title line, in the abstract, and in the subtitle, where it functions as a claim about statistical power that the experiment does not possess. The notebook does disclose the subsample in the section-8 markdown, which I credit, but disclosure in the methods does not license the number in the abstract.

REQUIRED REPAIR

State the effective sample size in the abstract, or run a learning curve from ten thousand rows to the full corpus and show the score saturating, which would justify the cap on evidence rather than on assertion.

DEFENSIBLE REPLY

The 311,295 you subtracted are all on the training side. My test set is a stratified 25 percent of the whole corpus — 143,766 sessions carrying 4,210 positives — so the 575,061 does back the statistical power of every held-out estimate I report. What I capped is how much the models were allowed to learn from, and that cap is printed in the same print statement as the corpus size: 'materialized 575,061 rows | trained on 120,000 (stratified subsample) | held-out 143,766'. There is no sentence anywhere in this notebook that claims to have trained on 575,061 sessions.

ORDERED CORRECTION

Re-aim at the training cap alone: the claim that these learners 'saturate long before' 120,000 rows is asserted in a code comment and never demonstrated, so the cap rests on assertion. Drop the 'statistical power the experiment does not possess' framing, which the held-out size refutes.

Six decimals on a number that survives none of them

EVIDENCE

The ranking table in cell 16 prints 0.999983 for RandomForest and 0.999913 for XGBoost, and the conclusion carries the ablation as 0.999983 going to 0.999979, a difference of four parts in a million. Meanwhile the notebook's own de-duplication moves the same quantity to 0.983574, and the notebook itself says tiny differences are re-split noise.

SCIENTIFIC CONSEQUENCE

Reporting to a resolution the estimator does not support invites the reader to treat differences that are pure noise as findings, and the conclusion does exactly that when it calls a four-part-in-a-million move barely moves and draws an inference from it. The code comment in cell 16 says six decimal places, a one point zero zero zero zero zero zero is a red flag, not a win, which shows the author understood the risk and printed the digits anyway.

REQUIRED REPAIR

Report to the resolution the estimator supports, which given a re-split spread of that size is two decimal places at most, and state the spread beside it.

DEFENSIBLE REPLY

I print six decimals on purpose, and the comment on that line says why: at six places a 1.000000 announces itself as a red flag rather than hiding behind 1.00. That is a diagnostic device, not a precision claim, and my ledger says in terms that tiny differences between ablation rows are re-split noise. So you cannot charge me with inviting the reader to read noise as finding on the strength of the decimals alone.

ORDERED CORRECTION

Re-write to the contradiction, which is real and which the student cannot answer: the ledger says 'Each ablation re-splits and refits, so tiny differences are re-split noise', and in the same paragraph reasons from 0.999983 -> 0.999979 to the substantive conclusion 'So the separability is multi-feature.' Charge the inference, not the digits.

Your winner is not the most accurate model in your own table

EVIDENCE

Cell 16 prints RandomForest at accuracy 0.999833 and ROC-AUC 0.999983, and LightGBM at accuracy 0.999847 and ROC-AUC 0.999831. So LightGBM is more accurate and RandomForest has the higher area. The code sorts by ROC-AUC and takes the top row, and prints, best model, RandomForest. No McNemar test, no paired comparison, no interval appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

The separation between the top three learners is in the fourth to sixth decimal place on a single split. Which model wins is decided by the arbitrary choice of ranking column, and the notebook does not say why ROC-AUC is that column rather than accuracy, nor that the choice changes the answer. A winner selected by an unstated tie-break on a difference smaller than the re-split noise is not a finding. Everything downstream, the confusion matrix, the importances, the per-type recalls, is computed for whichever model that arbitrary sort happened to put first.

REQUIRED REPAIR

Run McNemar's test on the paired held-out predictions of each model pair, report the p-values, and if the models are statistically indistinguishable, say so and report the family of tied models rather than crowning one.

DEFENSIBLE REPLY

The sort column is not unstated — the line reads 'winner by ROC-AUC (excl. baseline)' in the code you are reading from. I print the entire table, all four learners and the baseline, so no reader is shown a winner without the alternatives. And the gap you found is 0.000014 in accuracy on 143,766 rows, which is two rows. You are right that I ran no McNemar test and I should have. But every downstream artefact you list — the confusion matrix, the importances, the per-type recalls — would be visually identical under LightGBM, because the two models agree on 143,764 of 143,766 held-out rows.

ORDERED CORRECTION

Drop 'unstated tie-break' — the sort key is written in the code. Keep the durable half: no paired significance test is run between any pair of models anywhere in the notebook, so 'best model' is a sort result, not a finding.

The operating point was inherited, not chosen

EVIDENCE

Cell 18 sets predictions as probability greater than 0.5 and prints, operational false-positive rate at 0.5. The printed threshold appears in the label but no cell selects it, justifies it, or sweeps it. There is no cost matrix and no threshold selection anywhere in the file.

SCIENTIFIC CONSEQUENCE

Section 1 states the operational problem precisely: operators cannot read eleven million log lines, so the model must flag the three percent worth investigating without drowning them in false positives. That is a statement about asymmetric costs. The moment the costs are asymmetric, the operating point is the result, and 0.5 is the one value that encodes the assumption that a missed anomaly and a false alarm cost the same amount. Nobody in reliability engineering believes that.

REQUIRED REPAIR

State the cost of a missed anomalous block and the cost of an analyst-hour, build the two-by-two cost matrix, and select the threshold that minimizes expected cost on a validation fold. Then report the operating point you chose and why.

DEFENSIBLE REPLY

Ask me for the cost of a missed HDFS block failure and the cost of an analyst-hour on the cluster Xu and colleagues instrumented in 2009, and I cannot answer you honestly, because that cluster's economics are not in this dataset and never were. A threshold chosen by minimising expected cost against numbers I would have to invent is worse than a threshold I label plainly as the default, which is what '@0.5' in every printed line does. What I owed you is the threshold-free trade curve quantified — which is the same average-precision remedy you already charged me with under charge 8, so this is one omission counted twice.

ORDERED CORRECTION

Either merge into fault 8, or replace the cost-matrix demand with one that is satisfiable on a public corpus: sweep the threshold and report precision at a stated recall (say 0.90) alongside the alert count that recall implies. Do not ask for a cost matrix the corpus cannot supply.

A false-positive rate with no denominator in the world

EVIDENCE

Cell 18 prints, operational false-positive rate at 0.5 equals 0.0001, nineteen benign flagged of 139,556. The literature review cites Sommer and Paxson on the closed-world critique. Nowhere does the notebook multiply that rate by a session arrival rate to produce alerts per analyst per day.

SCIENTIFIC CONSEQUENCE

A rate of nineteen in 139,556 is meaningless to an operator until it is expressed in sessions per hour on a real cluster. Section 1 grounds the whole motivation in analyst workload, so the notebook has committed itself to the operational frame and then declines to complete the arithmetic that frame requires. And the rate itself is measured on a held-out set that is 99.9 percent training rows, so even the fraction is not an out-of-sample quantity.

REQUIRED REPAIR

State the block-session arrival rate of the cluster, multiply by the false-positive rate, and report alerts per analyst per day beside the recall. If the arrival rate is unknown, say the operational claim cannot be made.

DEFENSIBLE REPLY

On the printed numbers precision is actually strong here: with 4,210 held-out positives and recall above 0.99, roughly four thousand two hundred true alerts sit beside nineteen false ones, so the alert stream is dominated by real anomalies and the workload argument is not obviously damaging to this notebook.

Hand-set constants, and cross-validation on a slice of a subsample

EVIDENCE

Cell 16 instantiates the models with fixed constructor arguments and no search object. Cell 24 sets `cvX` and `cvy` to `X-train dot iloc up to forty thousand`, and cross-validates on that. The notebook's own section-12 markdown says the folds are not the corpus and that the figure describes a slice of a subsample, two steps removed from the 575,061 sessions loaded. There is no `GridSearchCV` or `RandomizedSearchCV` anywhere in the file, and none in the other thirty-five notebooks either.

SCIENTIFIC CONSEQUENCE

Without an inner selection loop there is no basis for calling any of these four the best learner for this task, only the best of four default configurations on one split. And the cross-validation number, 0.9996 plus or minus 0.0006, is reported in the conclusion's validity ledger where a reader will take it as a stability guarantee for the headline. It is not. It is fold-to-fold variation on a forty-thousand-row slice that was never de-duplicated, so duplicate rows straddle those three folds too, which the notebook admits and does not measure.

REQUIRED REPAIR

Nested cross-validation. An inner loop selects hyperparameters, an outer loop estimates performance, and both run on the grouped split rather than on a head slice.

DEFENSIBLE REPLY

You say a reader will take my 0.9996 as a stability guarantee for the headline. My ledger says, in the same sentence that reports it: those folds run on the head slice `Xtr.iloc[:40_000]` of the training subsample, not on the corpus, that slice is not de-duplicated, and the figure measures split stability, not freedom from contamination. I also print the one-line check a student can run to see the duplicates straddle the folds. There is no reading of that paragraph on which the CV number is offered as validation of the headline.

ORDERED CORRECTION

Cut the validity-ledger clause, which the ledger's own text contradicts. Keep and lead with the model-selection half: four fixed constructor configurations, no `GridSearchCV` or `RandomizedSearchCV` anywhere in the file, so 'best learner' means best of four defaults on one split. Per CORRECTION 5, note the std here is 0.0006 and not a formatting artifact, so the exact-zero objection does not arise.

Log anomaly detection is deployed forward in time

EVIDENCE

Cell 16 uses a stratified random split and nothing else. No temporal or time-ordered split appears anywhere in this notebook or in any of the thirty-six in the series. The conclusion's scope paragraph names the gap in its own words: the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

Nobody deploys a log anomaly detector by sampling a quarter of history at random and holding it out. You train on what has happened and you score what happens next, and the interesting failure mode, which is drift as the cluster's software and workload change, is invisible to a random split by construction. Xu and colleagues' own framing in 2009 is a system that runs forward over a live console stream. A random split answers a question nobody in this field asks.

REQUIRED REPAIR

Order the block sessions by first appearance in the raw log, train on the earlier portion, test on the later portion, and report the drop against the random-split figure.

DEFENSIBLE REPLY

The shipped Event occurrence matrix has no time column at all, only BlockId, Label, Type and the event counts, so a temporal split would require going back to the raw log and re-deriving per-block first-appearance timestamps, which is a good deal more than the twenty lines a GroupShuffleSplit would have cost.

Seed stability was checked for the audit, not for the claim

EVIDENCE

Cell 16 splits once with the notebook's fixed random state and reports the resulting numbers. Cell 24 prints seed equals 0. The eight-seed stability evidence in the audit record concerns the single-feature AUC, which ranged from 0.6764 to 0.6929 across seeds, and nothing in the notebook repeats the headline fit across seeds.

SCIENTIFIC CONSEQUENCE

The notebook is willing to spend eight seeds establishing that a diagnostic number is stable, and zero seeds establishing that the number in the abstract is stable. That is the wrong way round. The ablation table's own footnote says each ablation re-splits and refits so tiny differences are re-split noise, which is an admission that split-to-split variation exists and was never characterized for the reported result.

REQUIRED REPAIR

Refit and rescore the headline across at least ten seeds and report the mean and standard deviation, so that the four-part-in-a-million ablation differences can be read against the actual noise floor.

DEFENSIBLE REPLY

Show me the eight seeds. Cell 20 draws one sample at `random_state=1`. Cell 24 prints `seed=0`. There is no seed loop in this notebook, for the diagnostic or for the headline. The eight-seed range from 0.6764 to 0.6929 that you quoted is from your own audit probe, not from my file, so the asymmetry you are charging me with — careful about the diagnostic, careless about the claim — is not a thing I did. I spent zero seeds on both, and that is a different and simpler charge.

ORDERED CORRECTION

Restate as the plain version: the headline is one fit on one split at one seed, and no seed sweep exists anywhere in the notebook for any quantity. Delete the eight-seed comparison. Note the residual then substantially duplicates fault 9's no-interval charge, and per CORRECTION 6 no instability may be asserted without a measured spread to set against a boundary.

You named the evasion and did not measure it

EVIDENCE

Section 3 states that the features are event-template counts, so the task captures volume and frequency anomalies but is blind to pure ordering anomalies. The conclusion repeats it: counts ignore event order, so ordering-only anomalies are invisible to this representation. No cell perturbs an input, and no worst-case recall under any perturbation budget is reported.

SCIENTIFIC CONSEQUENCE

The notebook has written down the recipe for defeating itself. Any fault or intrusion that produces the same multiset of events in a different order scores identically to normal, and the detector's recall against that class is not low, it is exactly zero by construction. That is not a limitation, it is a complete evasion, and the notebook reports a recall of 0.995 or better for every scored type without ever placing that zero beside it.

REQUIRED REPAIR

Construct order-permuted variants of the anomalous sessions that preserve the count vector, score them, and report the worst-case recall under that stated perturbation. It will be zero, and that zero belongs in the abstract.

DEFENSIBLE REPLY

Read your own instruction back: construct order-permuted variants that preserve the count vector, then score them. A permutation that preserves the count vector produces the same twenty-eight integers, so my model returns the same probability to the last bit. There is no experiment there — the result is fixed by arithmetic before any model is fitted, which is exactly why I wrote it as an analytic statement in section 3 and again in section 13 rather than as a cell. The quantity that is measurable is how often a count vector actually carries both labels, and it is nine vectors out of 589, 550 rows, 36 of them minority-label. That is the empirical size of the blind spot in this corpus: six thousandths of one percent.

ORDERED CORRECTION

Two changes. First, replace the vacuous permutation experiment with the executable one: group by the count tuple, count the groups that are not label-unanimous, and report that as the representational ceiling. Second — and this is the stronger attack the charge currently misses — the same Kaggle archive the loader downloads also ships `Event_traces.csv`, whose `Features` column is the full ORDERED event sequence per block. The input `DeepLog` needs was in the folder, and the notebook read the count matrix beside it. That is a mirror-selection charge with teeth, and it is better than the evasion framing.

Gini importance is not an explanation

EVIDENCE

Cell 18 reads best dot feature underscore importances underscore for tree models and plots the top twelve as a horizontal bar chart titled top importances. No permutation importance is computed on held-out data, and no SHAP values are computed, anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward features with more distinct values and it is computed on training data, which on a corpus that is 99.9 percent duplicated between train and test means it is describing the memorization, not the concept. So the one panel in the notebook that purports to explain the model is measuring the artifact. A reader who takes E9 or E20 as the operative signal is being told something the method cannot support.

REQUIRED REPAIR

Permutation importance computed on held-out data under the grouped split, or SHAP values on a held-out sample, with the impurity chart removed or explicitly labelled as a training-set diagnostic.

DEFENSIBLE REPLY

Everything you say about impurity importance is true and I do not contest it. But point me at the sentence it supports. The only feature claim in my conclusion is that the strongest single feature is E9 at AUC 0.6896, and that number comes from cell 20, where I score each column alone against the label with `roc_auc_score` — a held-out-independent, cardinality-independent computation that has nothing to do with Gini. The bar chart is a picture with no inference hanging off it. Removing it changes not one conclusion in this notebook.

ORDERED CORRECTION

Say plainly that the panel supports no claim, and charge the weaker true thing: an unlabelled training-set diagnostic is presented in the results section beside held-out figures, which invites a reader to treat it as one. Delete 'the one panel in the notebook that purports to explain the model', which the text does not support.

The number cannot be placed beside any prior result

EVIDENCE

The section-2 review cites Xu and colleagues 2009 for the dataset and PCA-based detection, Du and colleagues 2017 for DeepLog, and Zhu and colleagues 2023 for Loghub. The related-approaches table explicitly labels its rows as drawn from the wider literature and not measurements reproduced on this exact corpus. No cell reproduces any of them.

SCIENTIFIC CONSEQUENCE

HDFS is called the classic benchmark in this notebook's own abstract, and the entire point of a benchmark is that results sit on a common axis. Reporting 0.999983 under a bespoke protocol with a 0.999 contamination places it on no axis at all. A reader cannot tell whether this beats PCA, matches DeepLog, or is simply measuring the split, and the notebook offers nothing that would let them tell.

REQUIRED REPAIR

Run the PCA detector of Xu and colleagues on the same corpus and the same protocol as a comparator, and report both numbers under the same split, so the reader can see what the modelling actually bought.

DEFENSIBLE REPLY

I will grant that I placed no number beside a prior result, and my related-approaches table says so in its own header. But look at what your remedy buys. You want Xu and colleagues' PCA detector run on the same corpus and the same protocol. The same protocol is the one with 0.999 train-test contamination. Both numbers would then be memorisation scores, and a reader comparing them would learn which method memorises better. The comparator only becomes informative after charge 1 is fixed, which makes this charge downstream of charge 1 rather than independent of it.

ORDERED CORRECTION

Order it explicitly after fault 1: demand the comparator under the corrected grouped protocol, not 'the same split'. As written the remedy reproduces the defect it is meant to expose.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 143,766 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

De-duplication removed most of my corpus here, so the movement you see is dominated by a real effect, not by split noise. The criticism still applies — I never measured the nuisance term — but it is at its weakest on this notebook, not its strongest.

Closing. This notebook is the most self-aware submission in the set, and that is precisely why it fails the hardest. It found the defect, measured it, graded itself F, ran the corrective experiment, and printed a number three orders of magnitude worse in residual error, and then it put the flattering figure in the abstract anyway. Naming a fatal limitation is not a defense at this level, it is a confession, and what you have confessed to is knowing. Come back with a grouped split over distinct count vectors, the disagreement rate as your noise floor, and every anomaly type in the recall table with its positive count printed beside it.

Industrial-IoT Intrusion Detection on WUSTL-IIoT-2021

1,194,464 network flows from a real water-storage-tank SCADA testbed at Washington University, 39 features, 7.28 percent attack, and the attack class is roughly ninety percent DoS.

HEADLINE

XGBoost reaches a held-out ROC-AUC of 1.000000 and an accuracy of 0.999963 on industrial-IoT intrusion detection, with a three-fold cross-validated ROC-AUC of 1.0000 plus or minus 0.0000.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self C

D1

1

External validity

D2

1

Statistical inference

D3

2

Construct validity

D4

0

Decision quality

D5

2

Reproducibility

D6

2

Honesty / scope

21 live faults

7 critical (FATAL)

14 major (CONTESTED)

2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook does several things properly and I want them on the record. It drops the endpoint identifiers by explicit rule rather than by hope, naming StartTime, LastTime, source and destination address, both ports and both address identifiers in a DROP list, so the model is not reading who talked to whom. Section 3 corrects a circulating claim about this dataset rather than repeating it, noting that the released file spans about seven hours on one day and that the fifty-three hour figure belongs to the collection campaign. And section 1 identifies the safety-critical cases up front as command injection and Backdoor rather than the DoS flood that carries the aggregate. That is the right instinct about where the difficulty lives.

EXAMINER RULING

You wrote in your own code comment that a ROC-AUC of 1.000000 is a red flag and not a win, and then you printed 1.000000 as the headline, in the ranking table, in the cross-validation line, and in all three rows of your ablation.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Your own comment calls this a red flag

EVIDENCE

In cell 16, the line that rounds the area under the curve carries the comment, six decimal places, a 1.000000 is a red flag, not a win. The table it produces prints 1.000000 for XGBoost and 1.000000 for LightGBM. Cell 24 prints a three-fold cross-validated ROC-AUC of 1.0000 plus or minus 0.0000. And all three rows of the cell-22 ablation come back at 1.0, headline, de-duplicated, and shortcut feature dropped alike.

SCIENTIFIC CONSEQUENCE

The author encoded the correct inference in the source, in advance, and then declined to draw it. A metric that returns exactly one under every perturbation you apply is not reporting a strong model, it is reporting that it has run out of resolution and can no longer distinguish anything. At that point the ablation table stops being evidence, because a measurement that cannot move cannot fail. The section-11 markdown even says this, that a score which barely moves is not vindication and means the distributions barely overlap, which is its own generation artifact. Every element of the correct reading is in this document, and the abstract still leads with 1.000000.

REQUIRED REPAIR

Switch to a metric with headroom left at this separation, namely average precision, and report it with a bootstrap interval. Then re-run the ablation against that metric, so that the ablation is capable of detecting an effect if one exists.

The safety-critical recalls have no sample size

EVIDENCE

Cell 18 prints, worst per-family recalls, Backdoor 0.96, command injection 0.971, DoS 1.0, Reconnaissance 1.0. The code that produces that line prints the recall dictionary and nothing else, no counts. Section 1 tells us what the counts roughly are: command injection and Backdoor are, in the notebook's own words, a few hundred flows each. The held-out set is a quarter of the corpus, so a few hundred becomes a few dozen.

SCIENTIFIC CONSEQUENCE

The conclusion says the weakest group sits at 0.960, which is where detection is thinnest, and the whole safety argument in section 1 rests on those two families. A recall of 0.96 computed on a few dozen positives has a binomial interval tens of points wide, and 0.96 on twenty-five positives means twenty-four out of twenty-five, where a single additional miss would print 0.92. None of that is visible, because the denominator is not on the page. I credit that nothing was suppressed here: the audit probe confirms all four families cleared the five-positive floor and none was skipped. That makes the missing denominators worse, not better, because there was nothing to hide.

REQUIRED REPAIR

Print the held-out positive count beside every per-family recall and attach a Wilson binomial interval to each. The two safety-critical families are the entire argument of section 1 and they are reported without the one number that makes a recall interpretable.

A false-positive rate that was never turned into alarms

EVIDENCE

Cell 18 prints, operational false-positive rate at 0.5 equals 0.0000, five benign flagged of 276,862. Section 1 says a false positive can trip a physical process. No cell multiplies that rate by a flow arrival rate, and no cost model appears anywhere in the file.

SCIENTIFIC CONSEQUENCE

Let me do the arithmetic the notebook owed, using only its own printed numbers, and let me be explicit that this is an illustration rather than a measurement. Five false positives in 276,862 benign flows is a rate of about eighteen per million. The notebook says the file holds 1,194,464 flows across about seven hours, which is roughly forty-seven flows a second, and a day has eighty-six thousand four hundred seconds, so about four million flows a day. Four million times eighteen per million is on the order of seventy false trips a day. Now, a production plant does not run at this testbed's flow rate, and that is precisely the point. Only the author knows the deployment rate, which is exactly why the author, and not the reader, owed this calculation. Reporting 0.0000 and stopping conceals whether the detector is admissible at any volume at all.

REQUIRED REPAIR

State the flow arrival rate of the target environment, multiply by the false-positive rate, and report false trips per day beside the recall. Then state the cost of a false trip and the cost of a missed command injection and choose the threshold that minimizes expected cost.

The group key was in your DROP list

EVIDENCE

Cell 11 defines DROP as Target, Traffic, y, family, StartTime, LastTime, SrcAddr, DstAddr, Sport, Dport, sIpId and dIpId. So SrcAddr, the source address, was in the dataframe. Cell 16 then calls `train_test_split` with `test_size 0.25` and stratify on the label, and nothing else. There is no `GroupKFold` and no `GroupShuffleSplit` in this notebook, and none in any of the thirty-six in the series.

SCIENTIFIC CONSEQUENCE

You proved the availability of the group key by removing it. Dropping SrcAddr as a feature is correct, and I credit it, but dropping it as a feature and using it as a grouping variable are different operations and only the first was performed. On a testbed with a handful of hosts, a random split places flows from the same host, in the same session, minutes apart, on both sides of the boundary. The model then matches a neighbour rather than learning a concept, and a held-out score computed that way estimates nothing about a host the model has not seen.

REQUIRED REPAIR

`GroupShuffleSplit` with groups set to SrcAddr, so that every flow originating from a given host lands on one side of the split. Report the drop against the random-split figure. That is roughly twenty lines and the column was already in the dataframe.

DEFENSIBLE REPLY

You are right that SrcAddr was in my hand and right that dropping a column as a feature is not the same as using it as a group key. But count the groups before you cost the fix at twenty lines. SrcAddr has fourteen distinct values. One of them, 192.168.0.20, carries 1,090,574 of my 1,194,464 flows and 27 attacks. Another, 209.240.235.92, carries 46,615 flows and every one of them is an attack. Four source addresses account for 99.7 percent of all attack traffic. A `GroupShuffleSplit` over that structure does not hold out hosts, it holds out labels: whichever side 192.168.0.20 lands on has essentially all the benign traffic, and whichever side 209.240.235.92 lands on has a third of the attacks and none of the rest. There is no assignment of fourteen groups to a 75/25 split that leaves both sides with all four families and a workable class balance. The grouping you asked for cannot be run here, and the reason it cannot is a fact about this corpus that neither of us should skip past.

ORDERED CORRECTION

Replace the remedy with one the corpus admits. Two work: leave-one-attack-source-out over the four hosts that emit attacks, which asks the genuinely interesting question of whether the detector generalises to an unseen attacker; or leave-one-episode-out over the four attack windows. Delete 'roughly twenty lines' — the fix is not twenty lines and the reason why is itself a finding.

Seven hours of one day, split at random

EVIDENCE

Cell 11 lists StartTime and LastTime in the DROP list, so both timestamp columns exist in the file. Section 3 says, in the notebook's own words, that the released CSV spans about seven hours on the nineteenth of August 2019, established by those very columns. Cell 16 splits at random. No temporal or time-ordered split appears anywhere in this notebook.

SCIENTIFIC CONSEQUENCE

The availability of the temporal split is proved by its own removal, exactly as with the host identifier. Seven hours is short enough that a forward split is easy to construct and long enough that it would be informative: train on the first five hours, test on the last two. What a random split cannot see is whether the detector still works on traffic that arrives after the traffic it learned from, which is the only condition under which it would ever be deployed. An intrusion detector evaluated by interleaving past and future is not being evaluated as an intrusion detector.

REQUIRED REPAIR

Sort by StartTime, train on the earlier portion, test on the later portion, and report the drop against the random-split figure. Both columns were already in the dataframe.

DEFENSIBLE REPLY

Take your remedy literally and run it. My capture starts at 09:46:03 and ends at 16:48:11 on the nineteenth of August. Sort by StartTime, train on the first five hours, test on the last two, exactly as you said. Your test set is 348,992 flows and every single one of them is normal. The last attack in this corpus is a Backdoor flow at 14:43:50. ROC-AUC is undefined on a test set with no positives; so is recall. Any forward cut at or past the seventy-percent mark gives me the same empty answer. The reason is that the four families are four disjoint episodes: Reconnaissance from 11:04 to 11:29, DoS from 12:25 to 13:07, command injection from 13:40 to 14:04, Backdoor from 14:17 to 14:43. This testbed was not run forward through mixed traffic. It was run as four scheduled experiments and then left idle for two hours.

ORDERED CORRECTION

Re-mechanise to the real defect, which the episode structure makes far sharper than the temporal charge: because each family is one contiguous window 25 to 42 minutes long, a random split cuts through the middle of a single attack episode and places consecutive flows of the same attack on both sides. That is not 'past and future interleaved', it is the same attack scored against itself. Demand leave-one-episode-out. Keep the credit already given for the seven-hour correction, which I verified at 7.036 hours on 2019-08-19.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

One column at 0.9879, and the question left open

EVIDENCE

Cell 20 prints, best single-feature AUC equals 0.9879, feature DIntPkt, followed by its own note that a near-1.0 single-feature AUC means this feature is near-sufficient, a shortcut, which may be legitimate signal or an artifact, and that it is not the same as target leakage. That note is exactly right and I want to be precise about it: near-sufficiency is not target leakage, and I am not alleging leakage. The fault is that having posed the disjunction, the notebook resolves it in neither direction.

SCIENTIFIC CONSEQUENCE

One column carrying nearly the whole task is the central construct-validity question of this notebook, and answering it is what separates a finding from a coincidence. The single ablation that was run removes that column and the score stays at 1.000000, which the conclusion reads as the separability is multi-feature. But that reading is the weaker of the two available: if many columns each independently solve the task, the attack and benign distributions barely overlap, which is a property of how this testbed was operated rather than of how attacks behave. The notebook writes that alternative reading down in section 11 and then does not measure the overlap.

REQUIRED REPAIR

Measure the class-conditional overlap directly, for instance by reporting the Bayes error under a kernel density estimate or the fraction of the feature space where both classes have mass, and separately report the score on a feature subset chosen to exclude rate and volume statistics.

DEFENSIBLE REPLY

You charge me in the next breath, under fault 7, with a positive class that is ninety percent denial of service — and I measured it at 89.99 percent, so you are right. Now read the feature you say I left unexplained. DIntPkt is destination inter-packet time. A flood is, definitionally, packets arriving with collapsed inter-arrival gaps. A single column that reaches 0.9879 by measuring exactly the thing a flood does to a link is not an unresolved disjunction between signal and artifact; it is the signal, and your own fault 7 says so. What I owe you is the score with DoS excluded, which is your fault 7's remedy, not a kernel density estimate of class overlap.

ORDERED CORRECTION

Merge into fault 7. The two charges ask the same question — does anything survive when the flood is removed — and fault 7's remedy answers both. If kept separate, state that DIntPkt's near-sufficiency is explained by the DoS majority and demand the single-feature AUC recomputed with DoS rows excluded, which is the version that is actually open.

The headline is a DoS flood

EVIDENCE

Section 3 states plainly that the attack class is about ninety percent DoS, so aggregate accuracy is base-rate inflated. The held-out set contains 21,754 attack flows, which you can check as 298,616 held out minus the 276,862 benign in the false-positive line. If ninety percent of those are DoS, then roughly twenty thousand of the twenty-one thousand seven hundred and fifty-four positives that produce the 1.000000 are floods.

SCIENTIFIC CONSEQUENCE

A ROC-AUC of one on a class that is overwhelmingly a flood is a statement about arithmetic, not about detection. Flooding changes packet counts and inter-arrival statistics by orders of magnitude, and a decision stump would find it. The claim that matters for an operational-technology deployment is whether the two quiet, targeted families are caught, and that claim is carried by two numbers with no denominators printed beside them. So the headline measures the easy ninety percent and the interesting ten percent is reported without the sample size that would let anyone judge it.

REQUIRED REPAIR

Report the held-out score with the DoS family excluded entirely, so the number reflects the families section 1 says are the real test, and report it with an interval.

DEFENSIBLE REPLY

The per-family table already shows command injection at 0.971 and Backdoor at 0.96, so the non-DoS families are in fact being recovered at high rates, and excluding DoS would not obviously produce a materially different picture.

One thousandth of a point from a different letter

EVIDENCE

Cell 20 assigns the single-feature sub-grade by hard thresholds written in the cell itself: F at 0.999 and above, D at 0.99 and above, C at 0.95 and above. The printed value is 0.9879, so the grade is C. The audit probe re-ran that measurement across eight seeds and got a range from 0.9879 to 0.9891. The highest seed sits nine ten-thousandths below the 0.99 cutoff, and even the lowest sits only twenty-one ten-thousandths below it. Every one of the eight seeds falls inside a one-hundredth-wide band around the boundary.

SCIENTIFIC CONSEQUENCE

A grade produced by rounding a quantity across a threshold that the quantity's own sampling spread straddles is not a measurement, it is a coin landing on its edge. The conclusion reports data trust grade C and tells the reader that the single-feature check drives the grade. Had the seed been chosen differently the same sentence would report D, and the accompanying prose would read as a warning rather than as reassurance. The letter is doing rhetorical work that the number cannot support.

REQUIRED REPAIR

Report the single-feature AUC with a bootstrap confidence interval and delete the letter. If a categorical summary is wanted, it must be assigned on the interval, and it must read as indeterminate when the interval straddles a cutoff, which here it does.

DEFENSIBLE REPLY

You wrote that the highest seed sits nine ten-thousandths below the 0.99 cutoff and the lowest sits twenty-one ten-thousandths below it, and then you concluded that had the seed been chosen differently the same sentence would report D. Those two statements cannot both be true. Every seed in your own range is below the cutoff. I ran the eight myself: 0.9878, 0.9879, 0.9891, 0.9885, 0.9880, 0.9879, 0.9883, 0.9885. Eight of eight return C. The spread is 0.0013 and the distance from my printed point estimate to the boundary is 0.0021, so the spread is smaller than the distance, and citing it as instability shows the opposite of what you claimed.

ORDERED CORRECTION

Per CORRECTION 6, drop the counterfactual entirely and fall back to the method form: the four cutoffs at 0.999, 0.99, 0.95 and 0.85 are author-chosen with no stated basis, and a letter assigned to a point estimate with no interval is not a measurement regardless of which letter it is. Note that this residual then duplicates fault 11's no-interval charge and only one of the two should carry it.

Contamination counts byte-identical rows and nothing else

EVIDENCE

Cell 20 builds a set of rounded training rows and checks membership of held-out rows in it, which is an exact-match test after rounding to six decimals. It reports 0.019. No nearest-neighbour distance is computed across the split boundary anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

On a testbed where a small set of hosts exchange the same protocol traffic for seven hours, consecutive flows from one host differ in a byte count or a millisecond and are otherwise identical. Those are not caught by an exact-match test, and they carry essentially the same information as a duplicate would. A contamination figure of 0.019 therefore certifies the absence of the one leakage mode the notebook happened to test for, and says nothing about the mode that the data generation process actually produces.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from each held-out row to the training set, in the scaled feature space, and report the fraction of held-out rows whose nearest training neighbour lies below a stated distance. That is the near-duplicate rate, and it is the number that governs.

DEFENSIBLE REPLY

The exact-duplicate rate is only 0.029 across the whole corpus and de-duplication leaves the score at 1.000000, so the duplicate structure demonstrably is not what props this result up, and there is no positive evidence that a near-duplicate measure would tell a different story.

You plotted the right curve and refused to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and plots it in the third panel of the figure. Average precision is computed nowhere in this notebook, and nowhere in the other thirty-five in the series. The ROC-AUC column meanwhile reads 1.000000 for two of the four learners.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to the count of false positives when negatives dominate, and here it has run out of resolution entirely, printing exactly one for two different models. Average precision would still separate them, and it is the metric that speaks to the analyst's experience, which is what section 1 says the notebook cares about. You built the instrument, put it on the page, and then ranked your models on the one metric that has nothing left to say.

REQUIRED REPAIR

Compute the area under the curve you already plotted, report it with a bootstrap interval, and rank the four learners on it. Also report precision at a fixed operationally chosen recall.

A winner chosen from a printed tie

EVIDENCE

Cell 16 prints XGBoost with ROC-AUC 1.000000 and LightGBM with ROC-AUC 1.000000. The code sorts the table by roc_auc descending and takes the first non-baseline row, then prints, best model, XGBoost. Note also that RandomForest, ranked third, has a higher accuracy at 0.999940 than LightGBM at 0.999913. No McNemar test appears anywhere in this notebook or in the series.

SCIENTIFIC CONSEQUENCE

Ranking on a column where two entries are equal to six decimal places means the winner is determined by whatever order pandas happened to preserve. Everything downstream is then computed for that arbitrarily chosen model: the confusion matrix, the importances, the per-family recalls, the false-positive rate, and the entire ablation table. The identity of the reported model is an artifact of a tie-break the notebook never mentions.

REQUIRED REPAIR

Run McNemar's test on the paired held-out predictions for each pair of models and report the p-values. Where models are statistically indistinguishable, report the tied set and say the data cannot separate them.

The operating point is a library default in a plant-trip setting

EVIDENCE

Cell 18 sets predictions as probability greater than 0.5 and labels the printed rate at 0.5. Section 1 says a false positive can trip a physical process and a missed intrusion can damage equipment. No threshold sweep, no cost matrix and no expected-cost selection appears anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

You have stated in prose that the costs are asymmetric and then chosen the one threshold that assumes they are equal. In operational technology the operating point is the result, because it decides whether the plant stops. Reporting recall and false-positive rate at 0.5 tells an engineer nothing about the point they would actually deploy at, and the notebook gives them no curve from which to choose one.

REQUIRED REPAIR

State the cost of a false trip and the cost of a missed command injection, build the cost matrix, sweep the threshold on a validation fold and select by minimum expected cost. Report the chosen operating point and its recall and alarm volume.

DEFENSIBLE REPLY

The asymmetry is stated in my section 1 and I stand by it. But you want the cost of a false trip on a water-storage tank and the cost of a missed command injection on it, and neither number exists in this dataset or in the paper that released it. A threshold minimising expected cost against two figures I would have to invent is a worse artifact than a default I label as a default, which is what '@0.5' does on every threshold-dependent line I print. And you asked me for the same cost matrix under fault 5. That is one omission billed twice.

ORDERED CORRECTION

Either merge with fault 5 or restate as something the corpus can satisfy: sweep the threshold and report Backdoor and CommInj recall together with the benign alarm count at each of three stated operating points, so an engineer can pick one. Do not ask for costs the release does not carry.

15

CRITICAL

DOCUMENTED

VERIFIED

THE CORPUS IS NOT THE EXPERIMENT

Three quarters of a million flows never entered the study

EVIDENCE

Cell 16 prints, materialized 1,194,464 rows, trained on 120,000 as a stratified subsample, held-out 298,616. One hundred and twenty thousand plus 298,616 is 418,616. Subtract that from 1,194,464 and 775,848 rows never entered either training or evaluation.

SCIENTIFIC CONSEQUENCE

The 1.19 million figure appears in the title, the subtitle and the abstract, where it functions as a claim about statistical power the experiment does not have. The section-8 markdown does disclose the cap, which I credit, but a disclosure in the methods does not license the figure in the abstract. It particularly matters here because the rare families are the whole argument, and a stratified subsample stratifies on the binary label, not on family, so the number of command-injection and Backdoor flows the model trained on is never printed.

REQUIRED REPAIR

State the effective sample size in the abstract, print the per-family counts in the training subsample, and run a learning curve to show the score saturating if you wish to justify the cap on evidence.

16

CRITICAL

UNTESTED

VERIFIED

NO MODEL SELECTION

Defaults throughout, and cross-validation on a slice

EVIDENCE

Cell 16 instantiates four models with fixed constructor arguments and no search object. Cell 24 sets cvX and cvy to X-train dot iloc up to forty thousand and cross-validates on that slice. No GridSearchCV or RandomizedSearchCV appears in this notebook or in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

Without an inner selection loop, calling XGBoost the best learner for industrial-IoT flow classification is unsupported. It is the best of four default configurations on one arbitrary split, decided by a tie-break. And the cross-validation figure sits in the conclusion's validity ledger where a reader will take it as a stability guarantee for the headline, when it is fold variation on a slice two steps removed from the corpus, computed on a metric that has saturated.

REQUIRED REPAIR

Nested cross-validation with an inner selection loop and an outer estimation loop, both running under the grouped and time-ordered split rather than on a head slice.

Eight seeds for the diagnostic, one for the claim

EVIDENCE

Cell 16 splits once with the notebook's fixed random state. Cell 24 prints seed equals 0. The eight-seed evidence in the audit record concerns the single-feature AUC, which ranged from 0.9879 to 0.9891, and no cell repeats the headline fit across seeds.

SCIENTIFIC CONSEQUENCE

The notebook is willing to spend eight seeds establishing that a diagnostic is stable and none establishing that the number in the abstract is. That is exactly backwards, and it matters more here than usual because the ablation's entire argument rests on differences that the conclusion itself attributes to re-split noise. Without a characterization of that noise, no ablation row can be read as evidence for or against anything.

REQUIRED REPAIR

Refit and rescore the headline across at least ten seeds, report the mean and standard deviation, and read the ablation differences against that spread.

DEFENSIBLE REPLY

Find the seed loop. Cell 20 draws one sample at `random_state=1`. Cell 24 prints seed equals 0. There is no repetition over seeds anywhere in this file, for the diagnostic or for the headline. The range from 0.9879 to 0.9891 you quoted is your probe's output, not mine, so the charge that I was careful about the diagnostic and careless about the claim describes something I never did. I spent no seeds on either, which is a simpler and shorter accusation.

ORDERED CORRECTION

Restate plainly: one split, one seed, no repetition anywhere. Delete the eight-seed comparison. Per CORRECTION 6 no instability may be asserted without a measured spread set against a decision boundary, and the residual then overlaps fault 11.

No adversary appears anywhere in a security detector

EVIDENCE

No cell in this notebook perturbs a feature vector, applies an evasion budget, or reports a worst-case recall under adaptation. The literature review cites Sommer and Paxson on the closed-world critique, which is the canonical statement of precisely this gap, and then no experiment follows from it.

SCIENTIFIC CONSEQUENCE

The features here are flow summary statistics: counts, durations, timing and jitter. An attacker who slows a command-injection campaign, pads packets, or spaces frames changes those statistics at essentially zero cost and without changing the attack's effect on the process. The reported recall is therefore the recall against an adversary who is not trying, and the number a plant operator needs, the recall against one who is, is not measured. Citing the closed-world critique and then running a closed-world experiment is the fault the citation describes.

REQUIRED REPAIR

Define a perturbation budget in the units of these features, for instance a bounded change to inter-packet timing and packet sizing that preserves the attack semantics, apply it to the held-out attack flows, and report worst-case recall under it.

DEFENSIBLE REPLY

For the family that carries ninety percent of my positives your evasion removes the attack. A denial of service works by volume and rate; an attacker who pads packets and spaces frames until DIntPkt looks benign has stopped denying service. You cannot evade a volumetric detector on a volumetric attack without giving up the attack. Where you are right is the other ten percent: command injection and Backdoor are already low-rate and already look like normal traffic in these features, and against those the adaptation you describe is cheap and I never measured it. Charge me for the 118 held-out flows where the argument holds, not for the 19,566 where it does not.

ORDERED CORRECTION

Restrict to CommInj and Backdoor and say why: those two families are already low-rate, so a semantics-preserving perturbation of timing and sizing is nearly free for them, whereas the same perturbation destroys a DoS. Then the charge is narrow, correct, and lands on the exact 118 held-out flows the notebook says are the real test.

Gini importance is not an explanation

EVIDENCE

Cell 18 reads best dot feature underscore importances underscore for tree models and plots the top twelve. No permutation importance on held-out data and no SHAP values appear anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward features with more distinct values and is computed on the training data, so it describes how the trees were built rather than what the model relies on out of sample. Here it is also the only account of mechanism in the whole notebook, so the sole explanation offered for a saturated score is produced by a method that cannot support an explanatory claim. Notice as well that the model's top feature by importance, which the audit probe identifies as sTtl, is a different column from the DIntPkt the notebook ablated, and the notebook never reconciles those two rankings.

REQUIRED REPAIR

Permutation importance computed on the held-out set under the grouped split, or SHAP values on a held-out sample, with the impurity chart either removed or explicitly labelled as a training-set artifact.

DEFENSIBLE REPLY

Everything you say about Gini importance is true. Now show me the claim it supports. My conclusion's only feature statement is that the strongest single feature is DIntPkt at AUC 0.9879, and that comes from cell 20, where I score each column alone against the label with roc_auc_score — a computation that has nothing to do with impurity and no cardinality bias. The bar chart supports no sentence in this notebook. And your top feature is not sTtl. I refitted with my own seed and my own pipeline: DstJitAct at 0.524, then sTtl at 0.239, then DIntPkt at 0.203.

ORDERED CORRECTION

Correct sTtl to DstJitAct, or drop the specific column claim, since the importances are only ever rendered as a chart image and never printed as text — there is no stored output to quote. Keep the weaker true charge: an unlabelled training-set diagnostic is placed in the results section beside held-out figures. Delete 'it is also the only account of mechanism in the whole notebook'; the single-feature audit is a second and better one.

No prior result was reproduced or run alongside

EVIDENCE

The section-2 review cites Zolanvari and colleagues 2021 for this dataset and a separate 2019 study for the machine-learning intrusion-detection motivation, and it cites Morris and Gao on industrial control system attack datasets. The related-approaches table explicitly labels its rows as drawn from the wider literature and not measurements reproduced on this exact corpus, and adds that the 2019 study used a different corpus and protocol from this run. No cell reproduces any published protocol.

SCIENTIFIC CONSEQUENCE

Reporting 1.000000 under a bespoke random split means the number cannot be compared with anything. A reader cannot tell whether this matches the dataset authors' own published results, exceeds them, or is simply reporting the split, because the notebook supplies no anchor. The honest reading of an unanchored 1.000000 on a testbed corpus is that the protocol is not discriminative, and only a comparator run under the same protocol could rule that out.

REQUIRED REPAIR

Run at least one published comparator, ideally the dataset authors' own reported configuration, under the same split as your models, and report both numbers side by side.

DEFENSIBLE REPLY

I concede that no comparator is run and that my related-approaches table says so in its own header. But your remedy is to run Zolanvari's configuration under the same split as my models. The same split is the one you have spent four charges telling me is invalid. Both numbers would then be produced by the protocol you rejected, and a reader comparing them would learn which method exploits episode adjacency better. This charge only becomes informative after faults 1, 3 and 9 are fixed, which makes it downstream of them rather than independent.

ORDERED CORRECTION

Order it explicitly after the split charges: demand the comparator under leave-one-episode-out, not 'the same split'. As written the requirement reproduces the protocol the rest of the list condemns.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 298,616 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. The strongest sentence in this notebook is a code comment, and it says that a ROC-AUC of one is a red flag and not a win. Everything after that comment ignores it: the ranking table, the cross-validation line, all three ablation rows, and the abstract. You had the source address and both timestamps in the dataframe, which means the grouped split and the forward-in-time split were each about twenty lines away, and you dropped both columns instead. Bring me the number under a split grouped on source host, with average precision rather than a saturated area, and with a positive count printed beside every one of your four family recalls.

Cyber-Physical Intrusion Detection on the HAI ICS Testbed

1,323,608 one-second snapshots from a hardware-in-the-loop industrial control testbed with a gas turbine, a boiler and a water-treatment process, 60 varying sensor and actuator signals, 0.68 percent of seconds labelled attack.

HEADLINE

XGBoost reaches a held-out ROC-AUC of 0.998755 and an accuracy of 0.998788 on per-second cyber-physical intrusion detection, with a three-fold cross-validated ROC-AUC of 0.9816.

DEFENSE
F
Selected-model estimand

DOCTORAL
9
/30 · REJECT

PRAXIS RUBRIC
10
/10 · self A

D1 1 External validity	D2 1 Statistical inference	D3 3 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 2 Honesty / scope

- 22 live faults
- 10 critical (FATAL)
- 12 major (CONTESTED)
- 1 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be generous here, because this is the cleanest notebook in the set and it deserves saying so plainly. Contamination is 0.007, the duplicate rate is 0.033, the strongest single feature reaches only 0.7465 on its own, and the trust grade is a genuine A on both sub-checks rather than a rescued one. It prints its per-stage positive counts in the loader cell, which none of its siblings do, and it constructs the multi-stage label by counting the flags rather than testing them in priority order, so a genuine multi-stage attack cannot be misfiled under whichever stage happened to be tested first. That is a careful piece of work that most authors get wrong. And the per-stage recalls it reports are honestly low, from 0.633 to 0.877, rather than the suspicious row of ones you see elsewhere. Everything I am about to say is said against a notebook that earned the right to be taken seriously.

EXAMINER RULING

Your code comment names the release's own attack-free training partition on one line, and the next line concatenates it into the test files and splits the result at random, which destroys the only protocol under which this corpus was designed to be evaluated.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

You named the official partition, then concatenated it away

EVIDENCE

Read cell 11 with me. The header comment says, and I quote, we use the clean comma-delimited hai-21.03 release, train equals attack-free, test equals labelled attacks. Four lines below, the code globs every CSV under hai-21.03, and the next statement is pandas dot concat over that list into a single dataframe. Cell 16 then splits that concatenation with train_test_split at test size 0.25, stratified on the label.

SCIENTIFIC CONSEQUENCE

This is not an oversight and it is not a gap in knowledge. The correct protocol is written in English in the same cell, one screen above the line that destroys it, and the release was constructed by its authors specifically so that this experiment would be run the right way round. Concatenating the partition and re-splitting at random converts a designed generalization test into an in-distribution memorization test. And it has a specific consequence you can derive: because the training files are attack-free, every one of the 8,947 attack seconds necessarily comes from the test files, so the random split scatters those 8,947 seconds across both halves of a boundary the dataset authors drew deliberately.

REQUIRED REPAIR

Load the train files as the training set and the test files as the held-out set, exactly as the release defines them, and report that score. It is a change of two lines in the loader. If you also want a random-split number for comparison, report it second and label it as the in-distribution ceiling.

Contamination cannot see the second next door

EVIDENCE

Cell 20 rounds the training matrix to six decimals, builds a set of tuples from it, and counts how many held-out rows appear in that set. It reports 0.007, and the audit probe confirms 0.0067 on the first window and 0.0057 on a random window. The data are one-second samples of pressures, flows, valve positions and temperatures on a physical plant, as the abstract states.

SCIENTIFIC CONSEQUENCE

This is the decisive fault for any time-series notebook and it is why a clean contamination number here is not reassurance. A physical process does not move much in one second. Two consecutive seconds are near-identical across sixty analogue channels, differing in the third or fourth significant figure of a few sensors. A random split places second N in training and second N plus one in the held-out set constantly, and the model then predicts the held-out second by effectively interpolating its neighbour. An exact-match test at six decimals cannot detect any of that, by construction, because the neighbouring row is not identical, it is merely almost identical. So a contamination of 0.007 certifies the absence of the leakage mode that does not occur here and is silent about the one that does. The A grade on that sub-check is therefore not evidence of a clean split, and the conclusion's sentence that neither check flags a problem, so nothing here explains the score away, does not follow.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from each held-out row to the training set in the standardized feature space, and report the fraction of held-out seconds whose nearest training neighbour lies within a stated distance. That is the number that governs on this corpus, and it will not be 0.007.

Nine false trips a day on a running plant

EVIDENCE

Cell 18 prints, operational false-positive rate at 0.5 equals 0.0001, thirty-four benign flagged of 328,665. The abstract says the testbed logs process signals once a second, and section 1 says the honest challenge is catching manipulations without false alarms that would trip a running plant.

SCIENTIFIC CONSEQUENCE

Do this arithmetic with me, because it uses only the notebook's own numbers and the length of a day. Thirty-four false alarms in 328,665 benign seconds is one false alarm about every nine thousand seven hundred seconds. A day contains eighty-six thousand four hundred seconds. Divide, and this detector raises roughly nine false alarms every twenty-four hours on a plant where the notebook says a false alarm can halt the process. That is a shutdown before lunch, every day, forever. And someone will say the deployed system would debounce, would require several consecutive alarms before tripping. Perhaps, but then the debounced rate is the result, and it is not reported, nor is the recall that survives debouncing. The notebook reports 0.0001 and stops at exactly the point where the number becomes interpretable.

REQUIRED REPAIR

Multiply the false-positive rate by the one-hertz sampling rate, report false trips per day, and state the debounce or persistence policy along with the recall that survives it. Then state the cost of a false trip and the cost of a missed manipulation and choose the threshold by minimum expected cost.

The worst recall in the notebook rests on about forty-five events

EVIDENCE

Cell 18 prints, worst per-family recalls, stage unflagged 0.633, stage P3 0.782, stage P2 0.8, stage P1 0.845, stage multi 0.877, and prints no counts beside them. I credit that cell 11 does print the corpus-wide counts, which is more than its siblings do: stage P1 6,158, stage P2 1,110, stage multi 1,091, stage P3 409, stage unflagged 179. And the quarter arithmetic is confirmed by the notebook's own false-positive line: 330,902 held out minus 328,665 benign gives 2,237 held-out positives, and 8,947 times a quarter is 2,236.75. So the held-out set holds on the order of forty-five unflagged-attack seconds and on the order of a hundred stage-P3 seconds.

SCIENTIFIC CONSEQUENCE

A recall of 0.633 on roughly forty-five positives means about twenty-eight caught and seventeen missed, and the binomial interval on that is roughly twenty-five percentage points wide, which is robust to the exact denominator being forty or fifty. The conclusion states that the weakest group sits at 0.633, which is where detection is thinnest, as though 0.633 were a located quantity. It is not. It is a point estimate on a few dozen events with no interval attached, and the reader is given no way to see that. The same holds for stage P3 at 0.782 on roughly a hundred events. These five numbers carry the notebook's entire safety argument and not one of them is accompanied by the sample size in the cell where it is printed.

REQUIRED REPAIR

Print the held-out positive count beside every per-stage recall in cell 18 and attach a Wilson binomial interval to each. Where the interval is wider than the differences between stages, say that the stages cannot be ranked.

The time column was in the file and was dropped

EVIDENCE

Cell 11 defines DROP as time, attack, attack_P1, attack_P2, attack_P3, y and family. So the time column exists in the dataframe and its removal is the proof of its availability. Cell 16 splits at random. No temporal or time-ordered split appears anywhere in this notebook or in any of the thirty-six in the series.

SCIENTIFIC CONSEQUENCE

An industrial control anomaly detector is deployed forward in time and in no other way. It is fitted on the plant's past and it scores the plant's next second. The failure mode that matters, drift as setpoints change, as equipment ages, as operating regimes rotate, is invisible to a random split by construction, because a random split lets the model see the future of every period it is asked about. The notebook's scope paragraph names the gap in its own words, saying the split is random, not temporal or entity-grouped. Naming it is not running it, and the column needed to run it was in the dataframe and was deleted.

REQUIRED REPAIR

Sort by the time column, train on the earlier portion, test on the later portion, and report the drop against the random-split figure. Report both, and put the temporal number in the abstract.

DEFENSIBLE REPLY

Run your remedy. Sort my 1,323,608 rows by time, train on the earlier portion, test on the later. At a quarter held out your test set is 330,902 rows with zero attacks. At a third, zero. At half, zero. The last attack second in this corpus sits at 46.4 percent of the timeline, because the two largest training captures, train2 from 31 July and train3 from 4 to 10 August, come after every test capture, and they are attack-free by design. There is no temporal number to put in my abstract. And your own charge 1 tells me to restore the release's partition, which means training on 11-13 July, 31 July to 3 August and 4-10 August, and testing on 7-8 July, 9-11 July, 13-14 July, 28 July and 30-31 July. That is training on the future and testing on the past. Your two remedies cannot both be run.

ORDERED CORRECTION

Keep the tier-1 observation that 'time' is first in the DROP list, per CORRECTION 3, but redirect it: the column was needed to build episode identifiers, which is fault 6's remedy and is executable, not to build a forward split, which is not. Then fold the residual into fault 6 rather than carrying it as an independent charge.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Attack episodes are the unit, and they were never grouped

EVIDENCE

Cell 16 uses `train_test_split` with stratification on the binary label only. No `GroupKFold` and no `GroupShuffleSplit` appears anywhere. Cell 11 prints per-stage counts but nothing anywhere in the notebook characterizes the temporal structure of the positives: no run lengths, no episode count, no scenario identifiers.

SCIENTIFIC CONSEQUENCE

Notice that the notebook cannot tell you, and does not tell you, whether its 8,947 attack seconds are eight thousand nine hundred and forty-seven independent events or a few dozen scenario runs of a few minutes each. That omission is itself the fault, because the protocol it owed depends entirely on which is true, and it never established which. If the positives occur in contiguous runs, and a hardware-in-the-loop testbed executing attack scenarios is the paradigm case of runs rather than isolated seconds, then a random split cuts through the middle of every episode and places seconds from the same attack, moments apart, on both sides. The model then recognizes the episode rather than the manipulation, and the reported per-stage recalls measure episode recall on episodes it has already seen.

REQUIRED REPAIR

Identify contiguous runs of the attack flag, assign each run an episode identifier, characterize and print the run-length distribution, and split with `GroupShuffleSplit` on that episode identifier so no episode straddles the boundary. Report per-stage recall on wholly unseen episodes.

The contest baseline is cited and never run

EVIDENCE

Section 2 cites Shin and colleagues 2021, which documents this exact release, along with a table row for HAIcon 2020 baselines described as time-series anomaly detectors designed for point-adjusted anomaly scoring, and Kravchik and Shabtai 2018 for one-dimensional convolutional anomaly detection on process data. The table explicitly labels its rows as drawn from the wider literature and not measurements reproduced on this exact corpus. No cell reproduces any of them.

SCIENTIFIC CONSEQUENCE

This dataset was released with a scoring contest, which means the community agreed on an evaluation protocol for it, and that protocol uses point-adjusted anomaly scoring on the designed partition rather than per-second classification on a random split. Reporting 0.998755 under a bespoke protocol places it on no shared axis. A reader cannot tell whether this beats a HAIcon entry, matches the residual approach of Kravchik and Shabtai, or merely reflects the split, and the notebook offers nothing that would let them tell. The section-2 table even records that the contest metric and this notebook's metric are different objects, so the incomparability was known.

REQUIRED REPAIR

Run at least one comparator under its own published protocol on the designed partition, ideally the next-step residual model of Kravchik and Shabtai that section 2 names as the honest approach, and report both numbers under the same protocol.

DEFENSIBLE REPLY

I accept that I ran no comparator. But note what your remedy requires: a comparator under its own published protocol on the designed partition. I do not have the designed partition — I concatenated it away, which is your charge 1. So this is not a second failure, it is the same failure seen from the other end. Restore the partition and the comparator becomes runnable; leave the partition destroyed and no comparator is meaningful. And my section 2 says in its own words that the HAIcon metric is point-adjusted anomaly scoring rather than per-second classification, so I did not claim comparability and did not smuggle one in.

ORDERED CORRECTION

Order it explicitly after fault 1 and say so in the text, so the student cannot collapse the two. Credit the section-2 table for recording that the contest metric and this notebook's metric are different objects; that is the correct disclosure and it is rarer than it should be.

Every positive comes from one set of files

EVIDENCE

The cell-11 comment states that in hai-21.03 the train portion is attack-free and the test portion carries the labelled attacks. The loader concatenates both into one frame. It follows arithmetically that all 8,947 positives sit in rows that came from the test files, and none in rows that came from the train files. The notebook never adds a file-provenance indicator, never tests whether provenance is predictive, and never reports the class balance by source file.

SCIENTIFIC CONSEQUENCE

If the two collection periods differ at all in setpoints, ambient conditions, sensor calibration or operating regime, then a model can achieve part of its separation by inferring which period a row came from, and label information rides along with that inference for free. This is not target leakage and I am not alleging that. It is confounding between the label and the file of origin, introduced by the concatenation, and it is entirely unmeasured. The control costs one column and one fit.

REQUIRED REPAIR

Add a binary source-file indicator, fit a classifier to predict that indicator from the sixty features alone, and report its held-out AUC. If the periods are distinguishable, the headline is confounded and the designed partition must be restored.

The stealthy attacker is the threat model and is never tested

EVIDENCE

No cell perturbs an input, states a perturbation budget, or reports a worst-case recall under adaptation. Section 2 cites Sommer and Paxson on the closed-world critique, and section 1 says the honest challenge is catching subtle stage-specific manipulations.

SCIENTIFIC CONSEQUENCE

In industrial control security the entire threat model is an adversary who moves the process slowly enough to look like drift. That is the definition of a stealthy process attack, and it is the reason residual and sequence models exist for this domain. The reported recalls, from 0.633 to 0.877, are recalls against the specific scenarios this testbed happened to execute, and they say nothing about an attacker who observes the detector's sensitivity and manipulates within it. Section 1 poses that adversary in words and then no experiment addresses it.

REQUIRED REPAIR

Define a perturbation budget in engineering units on the manipulated channels, apply the smallest perturbation that still achieves the physical effect, and report worst-case per-stage recall under that budget.

DEFENSIBLE REPLY

Tell me how to build the input. My row is 60 simultaneous readings from a gas turbine, a boiler and a water-treatment loop that are coupled by the physics of the plant — pressures follow valve positions, flows follow pressures, temperatures follow both with lag. If I perturb the manipulated channels by your budget and leave the rest, I have written down a plant state that cannot exist, and any recall I report against it is a number about an impossible world. If I perturb everything consistently, I need the plant model, and hai-21.03 ships process logs, not a simulator. The honest version of your charge is the one my section 3 already makes: a per-second classifier is the wrong instrument for a slow manipulation, and a residual model over time is the right one. That is a modelling limit, not an experiment I withheld.

ORDERED CORRECTION

Restate as an evaluation-design charge rather than an experiment charge: the reported recalls are recalls against the 50 scenarios this testbed happened to execute, the notebook never says so, and the conclusion presents 0.633 to 0.877 as properties of the detector rather than of that scenario set. That version is unanswerable and does not require a plant model.

You plotted the right curve and refused to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and plots it in the third panel. Average precision is computed nowhere in this notebook, and nowhere in the other thirty-five in the series. The conclusion itself says accuracy is meaningless at under one percent attacks.

SCIENTIFIC CONSEQUENCE

The notebook correctly rejects accuracy at a 0.68 percent positive rate and then adopts ROC-AUC, which suffers from a related insensitivity: it is indifferent to the absolute count of false positives when negatives dominate. Nine false trips a day and ninety false trips a day differ enormously to an operator and differ in the fourth decimal of ROC-AUC. Average precision registers that difference and is the metric the conclusion's own reasoning demands. You built the instrument, put it on the page, and ranked four learners on the metric that cannot see the thing you said mattered.

REQUIRED REPAIR

Compute the area under the curve you already plotted, report it with a bootstrap interval, and rank the four learners on it. Report precision at a fixed operationally chosen recall as well.

The operating point is a library default

EVIDENCE

Cell 18 sets predictions as probability greater than 0.5 and labels the rate at 0.5. Section 1 says a false alarm can trip a running plant and the conclusion repeats that a false trip can halt a physical plant. No threshold sweep, no cost matrix and no expected-cost selection appears anywhere.

SCIENTIFIC CONSEQUENCE

You have written down twice that the costs are asymmetric and then adopted the single threshold that assumes they are equal. At 0.68 percent positives with a plant-trip consequence on one side and an undetected process manipulation on the other, the operating point is not a detail of the result, it is the result. The five per-stage recalls and the nine-alarms-a-day figure are both properties of a threshold nobody chose.

REQUIRED REPAIR

State the cost of a false trip and the cost of a missed stage manipulation, build the two-by-two cost matrix, sweep the threshold on a validation fold and select by minimum expected cost. Report the chosen point, its per-stage recalls and its daily alarm volume.

DEFENSIBLE REPLY

I wrote the asymmetry down and I stand by it. But you want the cost of a false trip on a hardware-in-the-loop turbine-boiler-water rig and the cost of a missed stage manipulation on it, and neither figure exists in hai-21.03 or in Shin's papers. Selecting a threshold that minimises expected cost against two invented numbers is a fabrication with more decimal places than the default I label as a default. And you asked me for this same cost matrix under charge 4.

ORDERED CORRECTION

Merge with fault 4, or restate to something the corpus supports: report the five per-stage recalls and the daily alarm count at three stated thresholds, so a plant engineer can pick an operating point without anyone inventing a cost.

Four learners ranked, none compared

EVIDENCE

Cell 16 prints XGBoost at 0.998755, LightGBM at 0.997606, RandomForest at 0.992782 and LogisticRegression at 0.895681 on ROC-AUC. The code sorts by that column and prints, best model, XGBoost. No McNemar test, no paired bootstrap and no interval appears in this notebook or in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

The gap between the top two is about eleven ten-thousandths of an area, estimated on a held-out set containing 2,237 positives. Whether that gap is real is a question a paired test answers in a few lines, and it was not asked. Everything downstream is computed for the model that sort happened to put first: the confusion matrix, the importances, all five per-stage recalls, the false-positive rate and every row of the ablation table. If XGBoost and LightGBM are statistically indistinguishable, then the reported per-stage recalls are one arbitrary draw from a set of equally defensible answers.

REQUIRED REPAIR

Run McNemar's test on the paired held-out predictions for each model pair, report the p-values, and where models cannot be separated, report the tied set rather than crowning one.

DEFENSIBLE REPLY

I ran no McNemar test and I should have. But look at the column you say the winner was drawn from arbitrarily: 0.998755, 0.997606, 0.992782, 0.895681. That is a monotone ordering with a hundred-point spread between top and bottom, not two entries printed identically to six places. Whichever of my top two you crown, the confusion matrix and the per-stage recalls come from a model at 0.998 or 0.9976, and no conclusion in this notebook turns on which. The tie-break artifact you are describing is a real problem elsewhere; it is not what happened here.

ORDERED CORRECTION

Drop 'the model that sort happened to put first' — nothing here happened to be first. Keep the true residual: the top gap of 0.001149 is estimated on 2,237 positives with an approximate standard error around 0.0007, so it is roughly 1.6 standard errors and the notebook asserts a winner without saying so.

An ablation read at the fourth decimal

EVIDENCE

The cell-22 ablation prints headline 0.998755, de-duplicated 0.998193 and shortcut feature dropped 0.998901. So removing the strongest single feature makes the score go up, by 0.000146. The conclusion reads that movement as, the ablation refutes a single-feature story, dropping that feature barely moves the AUC. The conclusion also states that each ablation re-splits and refits, so tiny differences are re-split noise, without ever measuring how large that noise is.

SCIENTIFIC CONSEQUENCE

You cannot simultaneously say that a difference is re-split noise and use its smallness as evidence for a conclusion. Either the noise floor is known, in which case print it and read the ablation against it, or it is not, in which case the ablation supports no inference in either direction. Reporting six decimals invites the reader to treat a rise of a hundred and forty-six parts in a million as informative, and the conclusion does exactly that.

REQUIRED REPAIR

Repeat every ablation configuration across at least ten seeds, report the mean and standard deviation of each, and state the ablation result as a difference with an interval rather than as a pair of six-decimal point estimates.

DEFENSIBLE REPLY

Two things. My best single feature scores 0.7465 and my headline is 0.998755. No single column can produce that, so the separability is multi-feature by arithmetic, before any ablation is run. The ablation is corroboration, not the warrant, and you can strike it without touching the conclusion. Second, on the decimals: my held-out set has 2,237 positives and 328,665 negatives, which is 735 million label pairs, so ROC-AUC on it has a quantum of 1.4 times ten to the minus nine. Six decimal places is not more resolution than my estimator supports; it is three orders of magnitude less.

ORDERED CORRECTION

Drop the decimals framing and the 'refutes a single-feature story' target, both of which are answerable. The surviving true point is narrow: the notebook says tiny differences are re-split noise and never measures that noise, so the ablation table is uninterpretable in either direction. That residual duplicates fault 16 and only one of the two should carry it.

14

CRITICAL

DOCUMENTED

VERIFIED

THE CORPUS IS NOT THE EXPERIMENT

Eight hundred and seventy thousand seconds never entered the study

EVIDENCE

Cell 16 prints, materialized 1,323,608 rows, trained on 120,000 as a stratified subsample, held-out 330,902. One hundred and twenty thousand plus 330,902 is 450,902. Subtract from 1,323,608 and 872,706 rows never entered training or evaluation.

SCIENTIFIC CONSEQUENCE

The 1.32 million figure appears in the title, the subtitle and the abstract, where it functions as a claim about statistical power the experiment does not possess. It matters more here than usual because the positives are so scarce. The subsample is stratified on the binary label, not on stage, so the number of stage-P3 and unflagged attack seconds the model actually trained on is never printed, and those are precisely the groups whose recalls the conclusion leads with.

REQUIRED REPAIR

State the effective sample size in the abstract, print the per-stage counts inside the training subsample, and run a learning curve if you wish to justify the hundred-and-twenty-thousand cap on evidence rather than on assertion.

15

CRITICAL

UNTESTED

VERIFIED

NO MODEL SELECTION

Defaults throughout, and cross-validation on a head slice

EVIDENCE

Cell 16 instantiates four models with fixed constructor arguments and no search object. Cell 24 sets cvX and cvy to X-train dot iloc up to forty thousand and cross-validates on that slice, printing 0.9816 plus or minus 0.0014. No GridSearchCV or RandomizedSearchCV appears anywhere in the notebook or in the series.

SCIENTIFIC CONSEQUENCE

Without an inner selection loop, XGBoost is not the best learner for cyber-physical intrusion detection, it is the best of four default configurations on one split. And note the discrepancy the notebook never explains: the held-out ROC-AUC is 0.998755 and the cross-validated figure on the head slice is 0.9816, a gap of about seventeen thousandths. Those two numbers disagree by more than every ablation difference in the notebook combined, and the conclusion reports both without reconciling them.

REQUIRED REPAIR

Nested cross-validation, with an inner loop selecting hyperparameters and an outer loop estimating performance, run on the designed partition rather than on a head slice. Then explain the gap between the cross-validated and held-out estimates.

Eight seeds for the audit, one for the claim

EVIDENCE

Cell 16 splits once with the notebook's fixed random state, and cell 24 prints seed equals 0. The eight-seed evidence in the audit record concerns the single-feature AUC, which ranged from 0.7401 to 0.7840 across seeds, a spread of 0.0439. No cell repeats the headline fit across seeds.

SCIENTIFIC CONSEQUENCE

The diagnostic that got eight seeds moved by four hundred and thirty-nine ten-thousandths across them. The headline, which got one seed, is being read at the fourth and fifth decimal. If a univariate AUC on sixty thousand rows swings that much with the seed, the assumption that the reported six-decimal held-out figure is seed-stable is an assumption, not a result, and every ablation comparison in cell 22 depends on it.

REQUIRED REPAIR

Refit and rescore the headline across at least ten seeds, report the mean and standard deviation, and re-read the ablation table against that spread.

DEFENSIBLE REPLY

Point at the loop. Cell 20 samples once at `random_state=1` and gets 0.7465, which is the number on the page. Cell 24 prints seed equals 0. There is no seed sweep anywhere in this file. The range from 0.7401 to 0.7840 is your probe's output. And even taken on its own terms the inference does not carry: that range comes from redrawing which 60,000 of my 1.32 million rows are sampled, whereas my held-out AUC is computed on a fixed 330,902 rows. Those are different estimators with different variance sources, and the spread of one does not bound the spread of the other.

ORDERED CORRECTION

Restate plainly: one split, one seed, no repetition anywhere in the notebook, and the ablation table is therefore read against an unmeasured noise floor. Delete the eight-seed comparison and the inference from the diagnostic's spread to the headline's, which does not follow. Per CORRECTION 6, no instability may be asserted without a measured spread set against a decision boundary.

The worst number belongs to a residual bucket

EVIDENCE

Cell 11 constructs stage unflagged as the default branch of a select over the three per-stage flags, that is, rows where the global attack column is set but no stage flag is. It prints a count of 179. Section 3 describes the group as marked as an attack by the global attack column while carrying no per-stage flag. Cell 18 then reports its recall as 0.633, the worst number in the notebook, and the conclusion leads the recall list with it.

SCIENTIFIC CONSEQUENCE

A residual bucket is not a class. Those 179 rows could be transitions at the boundary of an attack window, they could be annotation errors, they could be a stage that was attacked but whose flag was not recorded, and the notebook offers no evidence for any of these. Reporting a recall of 0.633 for a group of unknown semantics, and placing it first in the conclusion's list as the point where detection is thinnest, attributes a detection failure to the model when it may be an artifact of the annotation. A doctoral claim cannot rest on a class whose membership rule is only that no other rule applied.

REQUIRED REPAIR

Characterize those 179 rows: report their position relative to the flagged attack windows, whether they are contiguous with flagged seconds, and how many distinct runs they form. Then either assign them to a stage, treat them as unlabelled and exclude them, or report them as an annotation-quality finding rather than as a recall.

Gini importance is not an explanation

EVIDENCE

Cell 18 reads best dot feature underscore importances underscore for tree models and plots the top twelve. No permutation importance on held-out data and no SHAP values appear anywhere in the notebook. Note also that the model's top feature by that importance, which the audit probe identifies as P1_LCV01Z, is a different column from the P1_PCV02Z the notebook chose to ablate, and the notebook never reconciles the two rankings.

SCIENTIFIC CONSEQUENCE

Impurity importance is biased toward features with more distinct values, which on sixty analogue channels of differing resolution is a live concern, and it is computed on training data, so it describes tree construction rather than out-of-sample reliance. In a cyber-physical setting the identity of the informative channels is not a footnote, it is the engineering finding, because it tells a plant engineer which sensors to instrument. Producing that finding with a method that cannot support it is worse than producing none.

REQUIRED REPAIR

Permutation importance on held-out data under the designed partition, or SHAP values on a held-out sample, with the impurity chart removed or explicitly labelled a training-set diagnostic.

DEFENSIBLE REPLY

Show me the engineering finding you say I produced with the wrong method. My ledger names one channel, P1_PCV02Z, at AUC 0.7465, and that comes from cell 20, where I score each column alone against the label — no impurity, no cardinality bias, no training-set dependence. The bar chart supports no sentence in this notebook. And your P1_LCV01Z is not quotable against me: importances in this notebook are only ever drawn, never printed, so there is no stored output for either of us to cite.

ORDERED CORRECTION

Drop the P1_LCV01Z claim or mark it as a re-run result rather than a quotation, since the notebook stores no textual importance output. Keep the weaker true charge: an unlabelled training-set diagnostic sits in the results section beside held-out figures. Delete 'the identity of the informative channels is the engineering finding' unless you can point at a sentence in the notebook making that finding; there is none.

The sibling corpus is cited and sitting in the same repository

EVIDENCE

Section 2 cites Goh and colleagues 2016, the SWaT water-treatment testbed dataset, and cites Kravchik and Shabtai for a detector built on SWaT process data. This series contains a SWaT notebook and a dedicated cross-distribution transfer notebook, so the capability exists in the same codebase. No cell in this notebook trains on one corpus and tests on another.

SCIENTIFIC CONSEQUENCE

The only claim that would matter operationally is that a process-anomaly detector transfers to a plant it was not fitted on, and every ingredient for testing that is present in the same repository and was not combined. Without it, the reported figure describes this testbed under this random split and generalizes to nothing, which is the closed-world result that Sommer and Paxson, cited in the same section, warn against.

REQUIRED REPAIR

Train on HAI and test on SWaT, and then the reverse, over whatever common process abstraction can be defined, and report both directions.

DEFENSIBLE REPLY

The transfer notebook in this series works because its four corpora are NetFlow version 2 and share a 43-field schema — that shared schema is the whole reason a model fitted on one can be scored on another. HAI gives me 60 signals from a GE gas turbine, an Emerson boiler and a FESTO water rig. SWaT gives 51 signals from a six-stage water treatment plant. There is no column in common, no unit in common, and no process in common except that both involve water somewhere. 'Whatever common process abstraction can be defined' is doing all the work in your requirement, and defining one is a research contribution, not a missing cell.

ORDERED CORRECTION

Re-mechanise to the weaker true claim CORRECTION 4 prescribes: no cross-corpus evaluation of any kind was attempted, and the notebook's reported figure therefore describes this testbed under this split and is not shown to generalise to any other plant. Delete the claim that the ingredients are present in the repository.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 330,902 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. This is the cleanest notebook in my set and it still fails, which is the most useful lesson available today. A contamination of 0.007 and a trust grade of A did not save it, because the leakage on a one-second process time series is adjacency, and an exact-duplicate test cannot see adjacency by construction. Meanwhile the release you loaded ships an attack-free training partition, your own comment says so, and the next line concatenates it away. Restore the designed partition, group on attack episodes, and bring me the daily false-trip count instead of a four-decimal rate.

In-Vehicle CAN-Bus Intrusion Detection (Car-Hacking / HCRL)

3,931,517 CAN frames captured from one Hyundai YF Sonata with four kinds of message-injection attack, described by ten columns: the arbitration ID, the data-length code and eight payload bytes.

HEADLINE

RandomForest reaches a held-out ROC-AUC of 1.000000 and an accuracy of 0.999995 on in-vehicle CAN-bus intrusion detection, recovering all four attack families at a recall of exactly one.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

10

/10 · self F

D1 1 External validity	D2 1 Statistical inference	D3 2 Construct validity	D4 1 Decision quality	D5 1 Reproducibility	D6 1 Honesty / scope

- 20 live faults
- 7 critical (FATAL)
- 13 major (CONTESTED)
- 3 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook is unusually candid and several of its moves are correct. It states that its 59.3 percent attack rate is a bounded-sample rate and not the on-bus base rate, so it never sells the prior as natural. It handles the label flag carefully: where a short frame pushes the R or T character into a byte position, the hex converter maps both characters to zero, so the flag itself cannot separate the classes, and that is a genuine piece of care most authors would miss. It names the hard case in domain terms rather than boilerplate, saying that masquerade and replay attacks reusing valid identifiers and plausible payloads need inter-frame timing and a sequence model, with Song and Lee cited as the direction not taken. And it reads its own row of perfect recalls correctly, saying that uniformity suggests the corpus is easy to separate, not that the detector is strong. That last sentence is the right instinct and it is why I am going to hold this notebook to a high standard.

EXAMINER RULING

Eighty-one percent of your held-out frames are verbatim copies of training frames, and that is not an accident of sampling, it is what a CAN bus is, which is exactly why a random split over frames was never an admissible protocol for this medium.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

A CAN bus is periodic, so a random split cannot work

EVIDENCE

Cell 20 prints an exact-duplicate row rate across the whole corpus of 0.853 and a train-test exact-row contamination of 0.811, with contamination grade F and overall data-trust grade F. The audit probe measured 0.8109 on the first window and 0.8079 on a random window, so the figure is a property of the corpus and not of the window chosen.

SCIENTIFIC CONSEQUENCE

Understand the mechanism, because it is the most interesting thing in this notebook and it is what makes the fault structural rather than incidental. A CAN bus is periodic by design. Engine control units broadcast the same arbitration identifier carrying the same or nearly the same payload at a fixed interval, tens or hundreds of times a second, for as long as the vehicle runs. The feature vector here is exactly the arbitration identifier, the data-length code and the eight payload bytes, with the timestamp removed. So identical rows are not noise in this corpus, they are the signal the bus is designed to produce. A duplicate rate of 0.853 is therefore the expected value, not a surprise, and it follows immediately that a uniformly random split over frames will place copies of the same broadcast on both sides of the boundary at a rate of roughly eighty percent. The held-out score is then dominated by rows the model has already memorized, and a hash table over ten small integers would produce the same 1.000000. This is not a case where the audit found a flaw in the data. The audit found the medium, and the protocol was still built on top of it.

REQUIRED REPAIR

Split by time window or by injection episode, never at random over frames. Take a contiguous block of the capture as the training period and a later, disjoint block as the held-out period, and report the score across that boundary. On a periodic bus, a random split is not a weaker protocol, it is not a protocol.

De-duplication is the wrong control, which is why it changed nothing

EVIDENCE

Cell 22 removes exact duplicates, which the label reports as eighty-five percent of rows, refits, and prints 1.000000. The conclusion then says, and I quote, de-duplication does not lower the score, so duplicate rows are not what props it up, and the overlap above is a caveat about the random split, not proof the score is fabricated. Note the scale: eighty-five percent removed from 3,931,517 rows still leaves on the order of five hundred and seventy-eight thousand distinct vectors, so this ablation did not run short of data.

SCIENTIFIC CONSEQUENCE

Here is why the ablation cannot address the objection. Each attack file is one capture in which one kind of injection was performed over some interval. De-duplicating collapses each repeated frame to a single instance, but it does not change which capture those instances came from, and the subsequent split is still uniformly random over the surviving rows. So the distinct payload tuples produced during one injection episode are still scattered across both sides of the boundary. The model learns the specific set of identifier and payload combinations that this episode emitted, and is then tested on other members of that same set. That is episode-level memorization, and de-duplication is orthogonal to it. The fact that the score stays at exactly 1.000000 after removing eighty-five percent of the corpus should have been read as a warning that the ablation lacks the power to detect anything, not as a clearance. The correct control was never run, and the conclusion treats the absence of a result from an irrelevant control as a positive finding.

REQUIRED REPAIR

GroupShuffleSplit with the group set to the injection episode, that is, the source capture and the contiguous interval within it, so that no episode contributes frames to both sides. Report the held-out score across unseen episodes. That is the number that would test the claim the conclusion makes.

Atlas proof

docs/course/security-ml-notebooks/nb32-car-hacking-can.ipynb · cell 22 _retrain_auc and cell 25 ablation interpretation; zero-based manifest cell IDs: 22=b4a7fb33, 25=e626e14c

Your class balance is an integer literal in cell 11

EVIDENCE

Follow the chain with me, it is three numbers and you can check every one from the PDF. Cell 11 loops over the four attack files, keeps every attack frame, and takes normal frames as a sample of at most four hundred thousand per file, a literal written as 400 underscore 000. Four files at four hundred thousand each is 1,600,000 benign frames. The loader printed 3,931,517 rows in total, so the attack count is 3,931,517 minus 1,600,000, which is 2,331,517. Divide that by 3,931,517 and you get 0.5930, which is exactly the attack rate the loader printed. Cell 16 then prints a majority-class baseline of 0.5930, which is that same quotient. And the split takes a quarter, stratified, so the benign rows in the held-out set number 1,600,000 divided by four, which is four hundred thousand exactly. Now look at cell 18: three benign flagged of 400,000. Not four hundred thousand and thirty-two. Exactly four hundred thousand, because the author typed it.

SCIENTIFIC CONSEQUENCE

Every headline quantity in this notebook is therefore a function of one constant in the loader. Change 400,000 to 100,000 and the attack rate, the baseline, the accuracy and the false-positive denominator all move, while nothing about the vehicle, the bus or the attack has changed. I credit the notebook for saying the rate is a bounded-sample rate and not the on-bus base rate, but that disclosure does not repair the arithmetic that follows from it. The accuracy of 0.999995 is quoted against a baseline of 0.5930, and that comparison reads as an enormous margin only because the sampler manufactured a nearly balanced problem out of a bus on which injected frames are, in reality, a small minority of traffic outside a flood. The operational false-positive rate inherits the same defect, since its denominator is a design decision.

REQUIRED REPAIR

Report every operational quantity at the true on-bus base rate. Keep the balanced sample for fitting if you like, then reweight the evaluation to the deployment prior, and report precision and alarms per second at that prior. State the true attack proportion in each source capture so the reweighting is checkable.

The timestamp was read, named, and thrown away

EVIDENCE

Cell 11 names the eleven columns as ts, can_id, dlc and the eight payload bytes, and then sets DROP to ts, y and family, with the comment, drop the timestamp, identifier. Cell 16 splits at random. Section 3 states that the notebook classifies each frame independently, dropping the timestamp, so the model cannot use inter-frame timing, which it calls the strongest CAN-bus intrusion-detection signal for flooding attacks.

SCIENTIFIC CONSEQUENCE

The timestamp is doing double duty here and both duties were declined. As a feature it carries the inter-frame timing that the notebook itself calls the strongest available signal, and it was dropped, which is a defensible scoping choice. But as a split key it is the only thing that makes an admissible protocol possible on a periodic bus, and it was dropped for that purpose too, silently, without the notebook ever noting that the same column it removed as a feature was the column that would have fixed its F grade. The availability of the temporal split is proved by its own removal. Train on the first portion of each capture and test on the last portion, and the eighty-one percent contamination falls, because a frame broadcast in minute one is no longer sitting in the held-out set alongside its own copy from minute two.

REQUIRED REPAIR

Keep ts out of the feature matrix, which is correct, but use it as the split key. Sort each capture by timestamp, train on an early contiguous window and test on a later disjoint window, and report the score and the contamination under that protocol.

You named the attack that defeats you and did not run it

EVIDENCE

The conclusion states that the genuinely hard case, masquerade and replay attacks that reuse valid identifiers and plausible payloads, needs inter-frame timing and a sequence model over the frame stream, citing Song and colleagues 2020 and Lee and colleagues 2017. Section 1 says the defensive question is whether a frame-level classifier catches injection, especially spoofing, which mimics legitimate identifiers. No cell constructs such a frame, applies any perturbation budget, or reports a worst-case recall.

SCIENTIFIC CONSEQUENCE

A replay attack on this bus emits a frame that is byte-identical to a legitimate frame. On the ten features used here, that frame is indistinguishable from normal by construction, so the detector's recall against replay is not low, it is exactly zero, and no experiment is needed to know it. The notebook reports a recall of exactly one for all four families it did test and never places that structural zero beside them. The reader is shown four ones and told in a separate paragraph that a fifth case exists and is hard. Those belong in the same table.

REQUIRED REPAIR

Construct replayed and masquerade frames by resampling legitimate frames from the same capture, inject them into the held-out set, and report the worst-case recall. It will be zero or near it, and that number belongs in the abstract next to the 1.000000.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

One car, one capture, one repertoire

EVIDENCE

The abstract states that this is real in-vehicle CAN-bus traffic from a Hyundai YF Sonata. The feature list in cell 11 is can_id, dlc and b0 through b7, ten columns, all read directly off the frame. Both sides of the split are drawn from the same four captures, so no cell in this notebook evaluates on data from a different vehicle, a different drive or a different bus.

SCIENTIFIC CONSEQUENCE

A model fitted on these ten columns is learning which identifier and payload combinations occurred in this capture and which did not. That repertoire is specific to one model of car, one wiring configuration, one firmware revision, one driving session. Nothing here measures whether it transfers, and nothing here can, because there is no second capture on the other side of the split. So the claim in the title, in-vehicle CAN-bus intrusion detection, is supported by evidence about one Sonata on one afternoon. The notebook does scope this in prose, saying to treat it as a benchmark number and not a deployment estimate, and I credit that, but a scoping sentence in section thirteen does not change what the title and the abstract assert.

REQUIRED REPAIR

Hold out an entire capture, or better an entire vehicle, and report the score on it. If no second vehicle is available in this release, say so in the abstract and restrict the claim to within-capture separability, which is what was actually measured.

DEFENSIBLE REPLY

The abstract says this is traffic from a Hyundai YF Sonata, one car, in plain words, so the scope is declared rather than hidden. And look at what your remedy would measure. My four captures are DoS, fuzzy, gear and RPM — one attack type each. Hold out a capture and you have held out an attack family, so a low score tells you nothing about whether the detector transfers to another drive; it tells you the model cannot recognise an injection type it has never seen, which is a different and much harder question that no frame-level classifier claims to answer. The experiment you are asking for cannot distinguish the two effects, because in this release capture and family are the same variable.

ORDERED CORRECTION

Keep only the reporting half, which is unanswerable: the claim must be restricted in the abstract to within-capture separability on one vehicle. Drop the leave-one-capture-out remedy, or state explicitly that it is confounded with attack family and would need a second normal capture from the same car to be interpretable.

You graded yourself F and then argued it away

EVIDENCE

Cell 20 prints, data trust grade F, and the section-13 conclusion answers it directly: the audit flags grade-F contamination because CAN frames repeat heavily, so a random split leaks, but the ablation shows the score is not an artifact of it, de-duplicating removes about eighty-five percent of rows and the AUC stays about 1.0, the separability is real, the grade F is a warning that a random split over repetitive frames is the wrong protocol, not that the score is fake.

SCIENTIFIC CONSEQUENCE

Read that passage carefully and you will see it concede the fatal point and then decline to act on it. It says, in the notebook's own words, that a random split over repetitive frames is the wrong protocol. That sentence is correct and it is the end of the matter, because the reported 1.000000 was produced by that wrong protocol. Whether the separability is also real under a right protocol is an open question that this notebook does not answer, since it never runs one. And the evidence marshalled for the rebuttal, the de-duplication ablation, addresses a different failure mode entirely, as I set out a moment ago. So the structure is: identify the correct protocol, state that yours is not it, decline to run the correct one, and keep the headline. At doctoral level, naming a fatal limitation is not a defense, it is a confession.

REQUIRED REPAIR

Run the episode-grouped or time-ordered split, and put that number in the abstract. If it holds at 1.000000, you have a genuine and interesting finding about how separable injection is on this bus. If it does not, you have a different paper. Either way you have an answer, which you currently do not.

DEFENSIBLE REPLY

My route was wrong and I accept it: de-duplication is not the control for what you are alleging, and I should not have written 'the separability is real' on the strength of it. What I would ask is that you charge it once. Charge 2 says the ablation is the wrong control and has no power. This charge says I used that ablation to argue away the F. Those are the same three sentences of my conclusion read twice.

ORDERED CORRECTION

Merge into fault 2, which owns the ablation's lack of power and now has a proof of it. If kept separate, restrict this charge to the rhetorical move alone and lead with the internal contradiction: section 11 says in advance that a score which barely moves 'is not vindication', and the ledger then writes 'The separability is real.' Do not argue that the claim is false — argue that the notebook could not have known either way.

Four families, four ones, no counts

EVIDENCE

Cell 18 prints, worst per-family recalls, fuzzy 1.0, rpm 1.0, gear 1.0, dos 1.0, and prints no counts. The code that produces the line emits the recall dictionary only. I credit that nothing was suppressed here: the audit probe confirms four groups with none skipped by the five-positive floor, so every family did appear.

SCIENTIFIC CONSEQUENCE

A diagnostic whose four cells all read one has no discriminative power. It cannot distinguish the hypothesis that the detector is excellent from the hypothesis that eighty-one percent of the held-out frames were already in training, and the second hypothesis is the one the audit in cell 20 supports. The conclusion partly sees this, saying the uniformity suggests the corpus is easy to separate rather than that the detector is strong, which is the right reading. But it then reports the row of ones as a result anyway, and without denominators no reader can compute an interval or check whether the smaller families were scored on enough events to mean anything. Worse, the notebook's own section on masquerade attacks tells us the four families present are all payload-anomalous injections, so the recall table is four measurements of the same easy case.

REQUIRED REPAIR

Print the held-out positive count beside every family recall, attach Wilson intervals, and report per-family recall under the episode-grouped split where the numbers will actually vary. A table of ones is a symptom to be explained, not a result to be reported.

DEFENSIBLE REPLY

Print the counts and see what changes. My held-out families carry 163,640, 148,705, 147,269 and 123,266 positives. A binomial interval on a recall of 1.0 at that sample size is narrower than the fifth decimal place. Your stated reason for wanting the denominators is that the table 'contains no information capable of distinguishing a strong detector from a memorizing one', and that is true — but it is true because 81 percent of those rows are copies of training rows, which is your charge 1, not because I omitted a number in the hundred thousands.

ORDERED CORRECTION

Keep the fault's own closing line and throw the rest away: 'A table of ones is a symptom to be explained, not a result to be reported.' Then explain it, which is now possible exactly: of the four held-out families, DoS, gear and RPM are 100 percent verbatim copies of training rows (147,269, 148,705 and 163,640 rows, none novel), while fuzzy is 0 percent copied (all 123,266 novel, because fuzzing emits random payloads). The four ones are therefore three lookups and one payload-anomaly detection, which is a mechanism, not a sample-size problem. As written the charge attributes the emptiness of the table to absent denominators in the hundred thousands.

You plotted the right curve and refused to read it

EVIDENCE

Cell 18 calls precision underscore recall underscore curve and plots it in the third panel of the first figure. Average precision is computed nowhere in this notebook, nor in any of the other thirty-five in the series. Cell 16 prints ROC-AUC 1.000000 for RandomForest and 1.000000 for XGBoost, and the code comment beside the rounding says, a 1.000000 is a red flag, not a win.

SCIENTIFIC CONSEQUENCE

A metric that returns exactly one for two different learners, and returns exactly one again under every ablation you apply, has stopped measuring. At that point it cannot fail, so it cannot provide evidence, and the entire ablation table in cell 22 is three readings from a saturated instrument.

Average precision at the deployment prior would still move. And the notebook wrote down the rule that a 1.000000 is a red flag, in the source, and then reported it four times as a result.

REQUIRED REPAIR

Compute the area under the curve you already plotted, at the deployment class prior rather than the sampler's prior, report it with a bootstrap interval, and re-run every ablation against that metric so the ablations are capable of registering an effect.

DEFENSIBLE REPLY

My held-out set has 582,880 positives and 400,000 negatives, which is 233 billion label pairs, so ROC-AUC on it resolves to four parts in a trillion. The actual value is 0.9999999974: about 611 discordant pairs. The metric did not stop measuring; the round-to-six-places in my source stopped printing. And your remedy is to compute average precision at the deployment prior — which is charge 3's remedy, and charge 3 is the one I cannot answer. Bill me once.

ORDERED CORRECTION

Delete 'has stopped measuring' and 'the entire ablation table is three readings from a saturated instrument'; substitute the true version, which is that the ablation is read at a printed resolution six orders of magnitude coarser than the estimator's. Then merge the prior-reweighting demand into fault 3. Note for the record that CORRECTION 2 was correctly observed here — the ROC-AUC-insensitivity line is nowhere in this charge, and on a 0.5930 attack-majority corpus asserting it would have inverted.

A winner chosen from a printed tie

EVIDENCE

Cell 16 prints RandomForest at ROC-AUC 1.000000 and XGBoost at ROC-AUC 1.000000. The code sorts the table by roc_auc descending, takes the first non-baseline row, and prints, best model, RandomForest. No McNemar test appears in this notebook or in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

Ranking on a column where the top two entries are equal to six decimal places means the winner is settled by whatever order the dictionary happened to preserve. Everything downstream then belongs to that arbitrary choice: the confusion matrix, the importances, all four family recalls, the false-positive rate, and every row of the ablation table. The reported model is an artifact of a tie-break the notebook never mentions.

REQUIRED REPAIR

Run McNemar's test on the paired held-out predictions for each model pair and report the p-values. Where two models are indistinguishable, report the tied set rather than crowning one.

The operating point is a library default on a moving vehicle

EVIDENCE

Cell 18 sets predictions as probability greater than 0.5 and labels the printed rate at 0.5. Section 1 asks whether the classifier catches injection without flagging normal traffic that would disrupt the vehicle. No threshold sweep, no cost matrix, no expected-cost selection and no alarms-per-second figure appears in the file.

SCIENTIFIC CONSEQUENCE

The notebook states in prose that the two error types have different consequences, one being a missed injection into a vehicle's control network and the other being a spurious intervention on a moving car, and then adopts the single threshold that assumes they cost the same. Furthermore, the false-positive rate of three in four hundred thousand cannot be turned into an alarm volume, because the notebook never states the frame rate of the bus. So even a reader who wanted to assess admissibility has no denominator to work with, and the notebook gives them none.

REQUIRED REPAIR

State the bus frame rate, convert the false-positive rate into false alarms per second and per hour of driving, state the cost of each error type, and select the threshold by minimum expected cost. Report the chosen operating point.

DEFENSIBLE REPLY

Half of this I owe you and half of it nobody can pay. The frame rate is in my file: the DoS capture runs 3,665,771 frames over 2,833 seconds, about 1,294 frames a second, so three false positives in 400,000 benign frames is on the order of thirty spurious alarms an hour of driving, and I should have printed that. But you also want the cost of a false alarm on a car in motion and the cost of a missed injection, and neither number is in HCRL's release or in Song's paper. A threshold chosen to minimise expected cost against two figures I invent is a fabrication with a decimal point.

ORDERED CORRECTION

Split it. The alarm-volume half deserves its own charge and should be led with, since this notebook has no B6 entry and it is the only one of the four where the arrival rate is directly measurable from a retained column — roughly 1,294 frames per second, giving about thirty false alarms per hour of driving at the printed rate. Then drop the cost-matrix half or restate it as a threshold sweep reported at three stated operating points.

Two point eight million frames were never used

EVIDENCE

Cell 16 prints, materialized 3,931,517 rows, trained on 120,000 as a stratified subsample, held-out 982,880. One hundred and twenty thousand plus 982,880 is 1,102,880. Subtract from 3,931,517 and 2,828,637 rows were loaded, converted from hexadecimal, and then discarded.

SCIENTIFIC CONSEQUENCE

The subtitle promises real CAN frames, at least one million, and the section-3 table opens with sixteen point six million. The model saw one hundred and twenty thousand. Those figures are doing work in the reader's mind about statistical power that the experiment does not possess, and it matters especially here because the whole question is whether the model has memorized a repertoire: a hundred-and-twenty-thousand-row training set on a bus with a small identifier repertoire is more than enough to memorize it and tells us nothing about how much data a genuine detector would need.

REQUIRED REPAIR

State the effective sample size in the abstract, and run a learning curve from ten thousand rows upward. On this corpus I expect it to saturate almost immediately, and that saturation point is itself the interesting measurement about how small the repertoire is.

DEFENSIBLE REPLY

The 2.8 million you subtracted are all training-side. My held-out set is 982,880 frames with 582,880 positives, a quarter of everything I materialised, so the corpus does back the power of every estimate I report. And the 16.6 million to 3.9 million reduction is not a discard I hid — it is the benign cap, printed in my provenance table as 'every attack kept + normal bounded', which is the same literal your charge 3 correctly attacks. You are billing me twice for one constant, and charge 3 is the version that lands.

ORDERED CORRECTION

Merge into fault 3, which owns the subsampling constant and states its consequences precisely. If kept separate, drop the 'statistical power the experiment does not possess' framing and keep only the learning-curve demand, which the fault itself notes would be interesting because it would measure how small the frame repertoire is.

Defaults throughout, and cross-validation on a head slice

EVIDENCE

Cell 16 instantiates four models with fixed constructor arguments and no search object. Cell 24 sets `cvX` and `cvy` to `X-train dot iloc` up to forty thousand and cross-validates on that. No `GridSearchCV` or `RandomizedSearchCV` appears in this notebook or in any of the thirty-six.

SCIENTIFIC CONSEQUENCE

Without an inner selection loop there is no basis for calling `RandomForest` the best learner for CAN intrusion detection, only the best of four default configurations on one split, chosen by a tie-break. And the cross-validated figure is worse than uninformative here, because the forty-thousand-row slice it runs on was never de-duplicated, so with an eighty-five percent duplicate rate the same frames straddle all three folds. That cross-validation number measures memorization consistency, and it is reported in the conclusion's validity ledger where a reader will take it as corroboration.

REQUIRED REPAIR

Nested cross-validation with an inner selection loop and an outer estimation loop, run under the episode-grouped or time-ordered split, on a de-duplicated training set.

Eight seeds for the diagnostic, one for the claim

EVIDENCE

Cell 16 splits once with the notebook's fixed random state, and cell 24 prints seed equals 0. The eight-seed evidence in the audit record concerns the single-feature AUC, which ranged from 0.7066 to 0.7121. No cell repeats the headline fit across seeds. Note also that the benign subsample in cell 11 is drawn with random state zero, so the composition of the corpus itself is one draw that is never varied.

SCIENTIFIC CONSEQUENCE

There are two unrepeatable random choices stacked here, not one. The corpus was assembled by a single sampling draw, and then the model was fitted on a single split of it. Neither is varied, so the notebook cannot say how much of its result depends on which four hundred thousand benign frames per file happened to be drawn. The conclusion attributes ablation differences to re-split noise without ever measuring that noise, which requires exactly the repetition that was not performed.

REQUIRED REPAIR

Vary both random choices. Re-draw the benign subsample across at least ten seeds and refit the headline across at least ten splits, and report the mean and standard deviation of the held-out score across that grid.

DEFENSIBLE REPLY

There is no seed loop in this file. Cell 20 draws one sample, cell 24 prints seed zero. The range from 0.7066 to 0.7121 is your probe, not my output, so the title 'eight seeds for the diagnostic' describes something I never ran. Your second sentence is a different matter and I have no answer to it.

ORDERED CORRECTION

Invert the charge. Lead with the corpus-composition point, which is unique to this notebook among the four and is unanswerable: the 400,000 benign frames per file are one draw at `random_state=0`, so the class prior, the baseline, the denominators and the held-out score all rest on a single unvaried sample, and no cell varies it. Delete the eight-seed framing entirely.

Gini importance on ten memorized columns

EVIDENCE

Cell 18 reads best dot feature underscore importances underscore for tree models and plots the top twelve, which on ten features is all of them. No permutation importance on held-out data and no SHAP values appear anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

Impurity importance is computed on the training data and is biased toward features with more distinct values, and here the arbitration identifier has far more distinct values than a single payload byte, so the ranking is partly an artifact of cardinality before any modelling question is asked. More seriously, with eighty-one percent contamination the trees are describing a memorized lookup, so the importance chart is an account of how that lookup was indexed rather than of what distinguishes an injected frame. Note also the tension the notebook never addresses: the single-feature audit crowns b3 while the audit probe's model-importance ranking puts b3 first and can_id second, and dropping both together still leaves 0.9951, which means no small set of columns explains anything.

REQUIRED REPAIR

Permutation importance computed on held-out data under the episode-grouped split, or SHAP values on a held-out sample, with the impurity chart removed or explicitly labelled a training-set diagnostic.

DEFENSIBLE REPLY

Show me the sentence the chart supports. My ledger names one field, b3, at 0.7113, and that comes from cell 20, where I score each column alone against the label with roc_auc_score. No impurity, no cardinality bias, no training-set dependence. The bar chart carries no inference in this notebook, and with ten features 'the top twelve' is simply all of them in order, which is a display, not a finding.

ORDERED CORRECTION

Keep only the narrow true point: an unlabelled training-set diagnostic sits in the results section beside held-out figures. Delete any claim that it is offered as an account of which frame fields carry the detection; no sentence in the notebook makes that claim.

19

MAJOR UNTESTED VERIFIED NOT COMPARABLE TO THE LITERATURE

The dataset authors' own detector is cited and never run

EVIDENCE

Section 2 cites Song, Woo and Kim 2020, who published this dataset and a deep convolutional detector on it, and Lee and colleagues 2017 on request-and-response time intervals. The conclusion names both as the direction the notebook did not take. No cell reproduces either.

SCIENTIFIC CONSEQUENCE

This corpus was released alongside a published detector by its own authors, which means a comparable number exists in the literature and could have been placed beside this one. Reporting 1.000000 under a random split with eighty-one percent contamination and no comparator leaves the reader unable to tell whether the modelling contributed anything at all beyond the protocol.

REQUIRED REPAIR

Reproduce the published detector on the same split as your four learners and report both under the same protocol, so the reader can see what the modelling bought.

DEFENSIBLE REPLY

Song and colleagues classify frame sequences with a convolutional network over a time window, which requires the temporal representation this notebook explicitly scoped out in section 3, so reproducing it is a different experiment on a different input representation rather than a comparator that was simply omitted.

20

MAJOR DOCUMENTED VERIFIED THIRD-PARTY MIRROR PROVENANCE

The data came from a re-upload, unverified

EVIDENCE

Cell 11 downloads from the Kaggle dataset pranavjha24 slash car-hacking-dataset, while section 2 credits the Hacking and Countermeasure Research Lab and Song, Woo and Kim 2020. The section-3 table says approximately sixteen point six million frames, and no cell verifies that figure against the upstream release, nor computes any file digest.

SCIENTIFIC CONSEQUENCE

Every number in this notebook is downstream of a file the notebook never verified came from the group it credits. On a corpus whose defining property is heavy exact duplication, a mirror that concatenated, re-ordered or partially de-duplicated the original files would change the duplicate rate, the class balance and the score, and nothing here would detect it. The provenance table asserts a row count for the upstream release and the loader never checks it.

REQUIRED REPAIR

Reconcile against the authoritative HCRL distribution: verify the per-file row counts and record a SHA-256 digest of each downloaded file in the notebook output, so the corpus a reader reproduces is provably the corpus you scored.

DEFENSIBLE REPLY

The notebook names the mirror explicitly rather than passing it off as the original, credits Song and colleagues correctly, and the four per-attack file structure it reads matches the documented HCRL release, so there is no positive indication that the mirror is altered.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 982,880 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. This notebook produced the most technically interesting finding in my set and then declined to follow it. It measured an eighty-one percent train-test overlap, graded itself F, and correctly wrote down that a random split over repetitive frames is the wrong protocol. That sentence is true, and once you have written it the reported 1.000000 is no longer available to you, because it was produced by the protocol you just called wrong. The de-duplication ablation cannot rescue it, because de-duplication is not the control for episode leakage, and the timestamp that would have given you the right control was in the file, in a column you named, and you dropped it. Bring me the score across a time boundary, with the label-conflict rate printed, and every metric restated at the true on-bus prior rather than at four hundred thousand.

Cyber-Physical Intrusion Detection on the SWaT Water-Treatment Testbed

One-point-four-four million one-second sensor and actuator snapshots from the SUTD Secure Water Treatment testbed, forty-six signals, three point seven nine percent of seconds labelled attack.

HEADLINE

RandomForest, held-out ROC-AUC 0.999979 on the 360,430-row held-out split, trained on a 120,000-row stratified subsample.

DEFENSE

F

Selected-model estimand

DOCTORAL

8

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self B

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 4 critical (FATAL)
- 19 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Let me say first what this notebook does better than most of the thirty-six. It tells the reader, in plain language and before any result, that the data came from a mirror that should not exist, and it calls that the reader's first finding rather than a footnote. It prints its majority-class baseline of 0.9621 and blocks the usual mistake of comparing an accuracy to a 0.5 ROC baseline. It runs a real ablation instead of narrating one, and when the score survives dropping the top feature it refuses to call that vindication, saying instead that multi-feature separability reflects how the corpus was generated. That is a better instinct than I see in most submitted dissertations.

EXAMINER RULING

You reported six significant figures of detection performance on a corpus your own section three tells the reader nobody was lawfully permitted to publish, and you split a one-hertz physical process at random, so what you measured is how similar one second of a water plant is to the second next to it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

1

CRITICAL

DOCUMENTED

VERIFIED

THIRD-PARTY MIRROR PROVENANCE

ORAL TOP 5

Headline computed on a corpus you say cannot exist

EVIDENCE

Section three, cell four, says it in your own words. Quote: what the loader below actually reads is a third-party Kaggle mirror, and the stored outputs in this notebook were computed from it. Under the terms above, nobody is permitted to have published that mirror. Then, three lines later: a corpus whose provenance you cannot verify is a corpus whose results you cannot verify. Cell eleven downloads vishala28 slash swat-dataset-secure-water-treatment-system. Cell sixteen prints a held-out ROC-AUC of 0.999979.

SCIENTIFIC CONSEQUENCE

You stated the sufficient condition for your own results being unverifiable. You then established that the condition holds. And you published anyway, to six significant figures. There is no inferential path from an unverifiable corpus to a verified result, and you wrote the premise yourself.

REQUIRED REPAIR

Obtain the release from iTrust under the academic request process, recompute, and report the delta. If the request cannot be granted in time, the provenance finding is the paper and the AUC is not reportable.

Atlas proof

docs/course/security-ml-notebooks/nb33-swat-ics.ipynb · cell 4, Dataset provenance & honesty caveats; cell 11, loader; zero-based manifest cell IDs: 4=f475a431, 11=d531943d

4

CRITICAL

UNTESTED

VERIFIED

NO TEMPORAL HOLDOUT ON TIMESTAMPED DATA

ORAL TOP 5

Timestamp present, explicitly dropped, split declined

EVIDENCE

Cell eleven, the loader, reads: DROP equals label, y, family, Timestamp, lowercase timestamp. Your own section-three invariant table even records the header quirk that leading spaces survive on space-Timestamp. The section-thirteen conclusion then says, quote, the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

Availability is proven by removal. You cannot drop a column you do not have. The single experiment that separates detecting an attack from memorising a run was sitting in the dataframe, cost about twenty lines, and was declined. And the declining is written into the conclusion as though writing it down were the remedy. At doctoral level, naming the fatal limitation is not a defence. It is a confession.

REQUIRED REPAIR

Train on the earlier portion of the eleven-day run and test on the later portion, with the attack campaign falling entirely in the test window, and report the drop from 0.999979.

Sixteen false trips a day on a water plant

EVIDENCE

Cell eighteen prints: operational false-positive rate at 0.5 equals 0.0002, sixty-three benign flagged of three hundred forty-six thousand seven hundred seventy-five. Cell eleven establishes that each row is one second.

SCIENTIFIC CONSEQUENCE

Do the arithmetic in front of the room. Sixty-three divided by 346,775 is one false alarm every five thousand five hundred and four seconds. There are eighty-six thousand four hundred seconds in a day. That is between fifteen and sixteen spurious attack alarms every single day, forever, on a six-stage water treatment plant. You named the consequence, in the conclusion, in the same paragraph as the number, and did not multiply. A rate that reads as reassuringly small in scientific notation is an unusable operating point once you attach it to the clock the plant actually runs on.

REQUIRED REPAIR

Multiply the false-positive rate by the per-day sample count, state the alarms-per-day figure in the abstract, and set the threshold from an alert budget a plant operator will actually accept rather than from the library default.

Adjacent seconds of one plant, split at random

EVIDENCE

Cell eleven loads one million four hundred forty-one thousand seven hundred nineteen one-second samples. Cell twenty prints an exact-duplicate row rate of 0.356 and a train-test exact-row contamination of 0.062. Cell sixteen splits with `train_test_split`, test size a quarter, stratified on the label, with no time ordering.

SCIENTIFIC CONSEQUENCE

A water treatment plant's tank level, flow rate and pressure at second t and at second t plus one differ by sensor noise. Under a random split, second t sits in training and second t plus one sits in the held-out set. Your 0.062 counts only the pairs that are bit-for-bit identical. It is a floor on the leakage, not a measurement of it, and on a one-hertz process time series the near-duplicates outnumber the exact ones by a wide margin. The 0.999979 is therefore a measure of temporal adjacency, not of detection.

REQUIRED REPAIR

Compute the distribution of nearest-neighbour distances from each held-out row to the training set and report the fraction lying inside the sensor noise band. Then split by contiguous time blocks with a guard band wide enough to exceed the process autocorrelation time, and report the new score.

DEFENSIBLE REPLY

The near-duplicate leak is real and I do not contest it, but the mechanism you assign to the printed numbers is not the one operating. I measured `merged.csv`: 494,991 of its 1,441,719 rows are full-row verbatim duplicates, timestamp included — 34.3% of the corpus is the same second written twice. That, not sensor-noise adjacency, is what the 0.356 duplicate rate counts, and the notebook's 0.356 reproduces exactly. And the 0.062 is not an arbitrary floor: it is measured against the 120,000 rows the model was actually fitted on, which is the correct denominator for the 0.999979 you are attacking. The remedy you ask for — contiguous time blocks with a guard band — is the remedy fault 4 already demands, so charging it twice prices one defect twice.

ORDERED CORRECTION

Re-mechanise: 'Your 0.356 duplicate rate is not process autocorrelation. 494,991 rows of this mirror are byte-identical including the timestamp — the file duplicates itself. Two distinct leaks are therefore uncounted: verbatim file duplication, which you measured but misread, and one-second adjacency, which you did not measure at all.' Then merge the split remedy into fault 4.

A winner declared on sixteen parts per million

EVIDENCE

Cell sixteen's ranking table: RandomForest 0.999979, XGBoost 0.999963, LightGBM 0.999958. The gap between first and second is sixteen millionths. Cell sixteen then executes print, best model, best_name, and every downstream diagnostic in cells eighteen, twenty, twenty-two and twenty-four is conditioned on that winner.

SCIENTIFIC CONSEQUENCE

Nothing in the notebook estimates the sampling spread of a held-out AUC on 360,430 rows, so nothing establishes that sixteen parts per million is a difference rather than a coin flip. Your entire results section, your feature importances, your per-family recall, your false-positive rate and your ablation all inherit from a selection that has not been shown to be reproducible under a second draw.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two models, or overlapping bootstrap confidence intervals on the AUC difference, before any model is called best.

DEFENSIBLE REPLY

The missing paired test is a real omission and I will run it. But nothing in the notebook rests on the ordering. RandomForest 0.999979, XGBoost 0.999963 and LightGBM 0.999958 span $2.1e-5$; the ablation, the false-positive rate and the per-family recall would be indistinguishable under any of the three. And the notebook never claims RandomForest is superior — it prints the top row. If you want the sharper version of your own charge, take it from the same table: on the accuracy column XGBoost is first at 0.999134 against RandomForest's 0.999010. The ranking is metric-dependent, which is a better argument than sixteen parts per million.

ORDERED CORRECTION

Replace the 'sixteen millionths' framing with the metric-inversion one: 'Your ranking column and your accuracy column name different winners. RandomForest leads on ROC-AUC by $1.6e-5$; XGBoost leads on accuracy by $1.2e-4$. You sorted on one of them and conditioned the entire results section on the result.'

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

Section thirteen withdraws section thirteen

EVIDENCE

The first sentence of section thirteen reads: as on HAI, point-in-time process features separate normal from attack SWaT snapshots above the roughly ninety-six percent normal baseline. The last paragraph of the same section reads: scope, the split is random, not temporal or entity-grouped, every number above therefore measures in-distribution separability only.

SCIENTIFIC CONSEQUENCE

Those two sentences are in the same section, four hundred words apart, and they cannot both stand. The first is a claim about process features and SWaT snapshots in general. The second says that nothing in the section licenses any claim beyond the particular random partition drawn with seed zero. This is the master fault of the whole series in its purest form: you identified the limitation precisely, wrote it down in the correct technical vocabulary, and then left the phenomenon-level sentence at the top of the section and in the abstract untouched. The disclosure paragraph costs nothing and buys nothing. Naming a fatal limitation is not a defence at doctoral level. It is a confession that the defect was understood.

REQUIRED REPAIR

Either restate the opening sentence as a claim about this random split of this mirror of this testbed, or run the split that would license the general claim. The two sentences must be made consistent, and only one of them can be made consistent by editing.

DEFENSIBLE REPLY

The sentence you read as phenomenon-level says 'point-in-time process features separate normal from attack SWaT snapshots above the ~96%-normal baseline' — that is a claim about this corpus's separability, which is exactly what 'measures in-distribution separability only' permits, and the next sentence of the same paragraph already withdraws its practical force with 'but that number is meaningless at ~4% attacks'. The abstract you say carries the unamended claim carries no phenomenon-level claim at all: it states the corpus size, the 120,000-row subsample and the 360,430-row held-out split and stops. What I will concede is narrower: 'As on HAI' imports a two-testbed generalisation into the first three words of a section whose last paragraph says cross-testbed transfer was never measured.

ORDERED CORRECTION

Retarget the charge to the two words that actually overreach: 'As on HAI, ...' asserts a cross-testbed regularity in the same section that says SWaT-to-HAI transfer is unmeasured. Drop the claim that the abstract carries an unamended phenomenon-level sentence — it does not.

Point estimates only, from one seed

EVIDENCE

Cell ten sets `RANDOM_STATE` equals zero with the comment, single seed used everywhere. Cell twenty-four prints a three-fold cross-validated ROC-AUC of 0.9980 plus or minus 0.0006, but the code above it reads `cvX`, `cvy` equals `Xtr dot iloc`, colon forty thousand. That cross-validation runs inside the training subsample on forty thousand rows. It says nothing about the 360,430-row held-out figure that the abstract quotes.

SCIENTIFIC CONSEQUENCE

A plus-or-minus computed within training data on a third of the training rows is a stability statistic for that slice, not a confidence interval for the reported result. So the notebook reports 0.999979, 0.999981 and 0.999987 side by side in the ablation table with no way for the reader to know whether any of those differences exist. Without an interval, none of the comparisons in the notebook are decidable.

REQUIRED REPAIR

Bootstrap the held-out AUC over at least a thousand resamples and report the interval; report a binomial interval on each per-group recall; and repeat the headline across at least ten seeds with mean and standard deviation.

DEFENSIBLE REPLY

You say none of the comparisons in the notebook are decidable without an interval. Name one the notebook decides. The ablation rows differ by $2e-6$ and $8e-6$ and the conclusion reads them as no effect, which is the correct reading at that magnitude. The per-family table has one group, so there is nothing to order. The learner ranking spans $2.1e-5$ and nothing downstream turns on it. The defect you are describing is real but it is the one you charge as fault 10: the plus-or-minus 0.0006 is a 3-fold CV on 40,000 rows of the training subsample, carried into the validity ledger as 'Winning learner: RandomForest (3-fold CV ROC-AUC 0.9980)' with no statement that it is an in-training number on an already-selected model.

ORDERED CORRECTION

Merge the load-bearing half into fault 10 and keep only the method complaint here: six significant figures are reported with no interval anywhere, which is a resolution claim the design cannot support.

You drew the precision-recall curve and never read it

EVIDENCE

Cell eleven prints an attack rate of 0.0379. Cell eighteen imports `precision_recall_curve`, calls it, and plots the result in the third panel with the title Precision-Recall. Nowhere in the notebook is `average_precision_score` called, and nowhere is a PR area printed.

SCIENTIFIC CONSEQUENCE

ROC-AUC is insensitive to the absolute count of false positives when negatives dominate, which is exactly the regime here at one attack second in twenty-six. You had the correct diagnostic on screen. You rendered it into the PDF. And then you selected your model, wrote your abstract and graded your data on the metric that hides the thing the plant operator cares about. Plotting the right instrument and declining to read it is worse than not having it, because it gives the reader the impression the question was answered.

REQUIRED REPAIR

Compute average precision with a bootstrap interval, report precision at a fixed operationally meaningful recall, and rank models on that rather than on ROC-AUC.

DEFENSIBLE REPLY

This is the one notebook of the four where your textbook statement about ROC-AUC under imbalance is correctly oriented — attacks are 3.79%, so the positive class really is rare and I will not argue the metric theory with you. I will argue the consequence. Cell 18 prints 63 false positives of 346,775 benign; the ranking table prints accuracy 0.999010 on 360,430 rows, so errors total 357, false negatives 294, true positives 13,361. Precision is $13,361/13,424 = 0.9953$ and recall is 0.9785, which is the 0.978 the notebook already prints. The average precision you say hides the operator's problem is approximately 0.99. What actually harms the operator is fifteen alarms a day, and that is your fault 5, where it belongs.

ORDERED CORRECTION

Keep the charge but state its true content: 'You had precision and recall available in printed counts and reported neither, so a reader cannot place your result beside the SWaT literature.' That is fault 13, and it is where this should have gone.

The operating point is the library default

EVIDENCE

Cell eighteen: pred equals pb greater than 0.5. The false-positive rate in the same cell is reported at 0.5. There is no cost matrix anywhere in cells ten through twenty-six.

SCIENTIFIC CONSEQUENCE

For a cyber-physical plant, a missed process attack and a spurious shutdown are not comparable costs, and the conclusion says so, in the phrase a false trip can disrupt water treatment. When the costs are asymmetric the operating point is not a detail downstream of the result. It is the result. You inherited it from a default argument in scikit-learn and never revisited it.

REQUIRED REPAIR

State an explicit cost matrix for a missed attack and a false trip, choose the threshold that minimises expected cost, and report the score at that threshold rather than at one half.

DEFENSIBLE REPLY

Fault 5 already tells me to set the threshold from an alert budget rather than from the scikit-learn default. This charge asks for the same act with a cost matrix attached. It is one defect and you have priced it twice; charge the cost matrix as the remedy for fault 5 and the fault count is honest.

ORDERED CORRECTION

Fold into fault 5 as its required remedy, or keep it separate only if you also state the cost ratio you would apply — otherwise the two faults have one fix.

No search, and the cross-validation is inside training

EVIDENCE

Cell sixteen instantiates RandomForest with sixty trees, XGBoost with eighty estimators and depth six, and LightGBM with eighty estimators, all hand-set, with no grid, no random search and no tuning loop. Cell twenty-four clones the winning model and cross-validates it on Xtr dot iloc, colon forty thousand.

SCIENTIFIC CONSEQUENCE

The winner was chosen on the held-out set in cell sixteen and then cross-validated on training rows in cell twenty-four. That ordering makes the cross-validated number a description of a decision already taken, not an independent estimate of it. There is no inner loop that selects and no outer loop that estimates, so the reported figure carries the optimism of the selection it is supposed to check.

REQUIRED REPAIR

Nested cross-validation. An inner loop that searches hyperparameters and picks the learner, an outer loop that estimates generalisation on folds the inner loop never touched, and the outer estimate reported as the headline.

One draw, six decimals

EVIDENCE

Cell ten: `RANDOM_STATE` equals zero, with the inline comment single seed used everywhere. Every `train_test_split` in cells sixteen and twenty-two passes that same constant. Nowhere does a seed vary.

SCIENTIFIC CONSEQUENCE

Six decimal places assert a resolution that one draw cannot support. Worse, the ablation table in cell twenty-two invites the reader to compare 0.999979 against 0.999981 against 0.999987, and the conclusion characterises the gaps as re-split noise. That characterisation is an assertion about a variance the notebook never measured, because it never drew a second sample.

REQUIRED REPAIR

Repeat the full pipeline across at least ten seeds and report the mean and standard deviation of the held-out AUC, so that the word noise refers to something you measured.

DEFENSIBLE REPLY

Everything in this charge except the word 'noise' is fault 7. And the sharper version you had available you did not take: the shortcut-dropped row calls `train_test_split` with the same `n`, the same `y`, the same `test_size` and the same `random_state` as the headline, so `sklearn` returns the identical partition. Nothing is re-split there. That makes the conclusion's blanket sentence factually wrong for one of the two ablation rows, which is a better charge than 'one draw, six decimals' — and it is the charge you did bring against nb34 and nb35.

ORDERED CORRECTION

Re-mechanise to match the nb34/nb35 version of this fault: 'Your conclusion says each ablation re-splits and refits. The shortcut-dropped row does not. Same seed, same row count, same label vector returns the same partition, so 0.999979 to 0.999981 is the measured effect of dropping PIT503, not noise.'

A security detector never met an adversary

EVIDENCE

A search of the full source of this notebook for the words adversarial, evasion, evade or perturb returns nothing. The reference list in cell twenty-six names Goh, Mathur and Tippenhauer, Kravchik and Shabtai, and Sommer and Paxson. None of them is an adversarial machine learning result and none is applied.

SCIENTIFIC CONSEQUENCE

SWaT's entire threat model is an attacker who manipulates sensor readings and actuator commands on the process bus. Your detector's inputs are sensor readings and actuator commands on the process bus. An adversary who can move PIT503 to stage the attack can move PIT503 to hide it, and at zero additional cost, because it is the same capability. A security detector whose worst case is unmeasured has no reported worst case, and the worst case is the only number a defender can plan against.

REQUIRED REPAIR

Evasion under a stated perturbation budget on the physical signals, respecting process constraints, with worst-case recall reported alongside the average-case number.

DEFENSIBLE REPLY

There is no adversarial evaluation and I will not pretend otherwise. But the notebook claims in-distribution separability of SWaT snapshots and nothing else, and an unmade claim cannot be falsified. Its conclusion already names the two next steps it thinks matter — a temporal or residual model, and cross-testbed transfer — and neither is robustness. I accept the sharpest form of your point, which is specific to this corpus rather than to the series: SWaT's threat model is an attacker who writes sensor and actuator values, and my features are sensor and actuator values, so here the adversary controls the input directly and the gap is more on-point than it is in most of the series.

ORDERED CORRECTION

Keep the PIT503 argument, which is genuinely specific to this corpus, and drop the framing that a security detector without a worst case is fatally defective — that judgement applies unchanged to all thirty-six and therefore separates none of them.

No number that can sit beside a published SWaT result

EVIDENCE

Cell twenty-six cites Kravchik and Shabtai, 2018, who evaluate detectors on SWaT. The notebook reports a per-second binary held-out ROC-AUC of 0.999979 and a single per-group recall of 0.978. It reports no precision, no F-measure, no per-attack-scenario result, and no evaluation on the canonical normal-versus-attack partition.

SCIENTIFIC CONSEQUENCE

The SWaT literature evaluates sequence models over time windows and reports precision, recall and F-measure per attack scenario. There is no counterpart in that literature to a per-second binary ROC-AUC on a random split. So your number is not better than, worse than, or comparable to anything. It is unplaceable, which for a benchmark result is the same as being uninformative.

REQUIRED REPAIR

Report precision, recall and F-measure on the canonical normal-and-attack partition with per-scenario breakdown, so that a reader can put the result in a table next to Kravchik and Shabtai.

DEFENSIBLE REPLY

I never claim comparability. Section 2 prints the cited work under a heading that reads, in bold, that these are not measurements reproduced on this exact corpus, and the table row against Kravchik and Shabtai says a 1D-CNN is a temporal residual model and that point-in-time classifiers are weaker. So the notebook states in advance that its number does not enter that table and says which direction the gap runs. What survives of your charge is that I could have printed precision and F-measure at almost no cost from the counts already in cell 18 — 0.9953 and 0.9868 — and did not.

ORDERED CORRECTION

Recast as the cheap-omission charge: 'Precision 0.9953 and F-measure 0.9868 were arithmetic away from your printed confusion matrix. You printed neither, so nothing here can be placed beside Kravchik and Shabtai even though the numbers existed.'

The generalization test is one directory away

EVIDENCE

The section-thirteen conclusion says, quote, cross-tested transfer, SWaT to HAI, which neither notebook measures, is the real generalization question for cyber-physical IDS. Notebook thirty-six, in the same folder, fits one model per corpus and scores every model on every corpus in cell twenty, printing a full four-by-four transfer matrix.

SCIENTIFIC CONSEQUENCE

The capability is not hypothetical and it is not elsewhere. It is in the same codebase, on the same template, written by the same author. You have a second ICS testbed in notebook thirty-one and a working transfer harness in notebook thirty-six, and you produced neither cell. You identified the experiment that would settle your central claim, wrote down that it was the real question, and shipped the headline instead.

REQUIRED REPAIR

Fit on SWaT and score on HAI, and fit on HAI and score on SWaT, on the shared physical-signal representation, and report both off-diagonal numbers alongside the diagonal.

DEFENSIBLE REPLY

Notebook 36 can loop train-on-one, test-on-another because all four of its corpora were re-featured by Sarhan onto one 43-field NetFlow layout — its own loader comment says that is exactly what makes the comparison possible, and I verified the parquets carry identical 43-column schemas. SWaT is 51 instrument tags belonging to a six-stage water plant; HAI is a different tag set on a turbine, boiler and water process. There is no shared representation to loop over, and your own required clause quietly concedes this by asking for one — 'on the shared physical-signal representation' — that does not exist and would itself be the contribution. The notebook says cross-tested transfer is the real generalization question. It calls it a question, not a cell.

ORDERED CORRECTION

Re-mechanise to the weaker true claim: 'No cross-corpus evaluation of any kind was attempted anywhere in this notebook, and notebook 36 proves the series knows how to run one.' Drop 'the harness exists, one directory away' — it does not port to a corpus with no shared schema.

Atlas proof

docs/course/security-ml-notebooks/nb33-swat-ics.ipynb · cell 25; zero-based manifest cell IDs: 25=3195433a

Eleven days does not hold sixteen point seven days of seconds

EVIDENCE

The abstract in cell one says the plant ran for eleven days with staged cyber-attacks and that the notebook loads one million four hundred forty-one thousand seven hundred nineteen one-second samples. The same claim appears as a code comment in cell eleven. Section three's invariant table breaks that total into 1,387,098 normal rows and 54,621 attack rows.

SCIENTIFIC CONSEQUENCE

Eleven days is nine hundred fifty thousand four hundred seconds. You printed one million four hundred forty-one thousand seven hundred nineteen. That is sixteen point seven days of one-second samples, five point seven days more than the run you describe. Either the samples are not one second apart, or the run was not eleven days, or the mirror contains material from outside the run you cite. The notebook does not say which, and given fault one it cannot.

REQUIRED REPAIR

Reconcile the row count against the iTrust release duration and state which of the three explanations holds, or drop the eleven-day claim from the abstract.

DEFENSIBLE REPLY

You offered three explanations and the true one is a fourth. I measured the file. merged.csv has 1,441,719 rows and 946,728 distinct timestamps, and 494,991 rows are byte-identical to another row including the timestamp field. 946,728 seconds is 10.957 days. The samples are one second apart, the run was eleven days, and the mirror contains nothing from outside it — it simply writes 34% of the corpus down twice. The notebook already prints the symptom as an exact-duplicate row rate of 0.356, and I reproduce that figure exactly at 0.3557. Your own rebuttal note, that the normal workbook might be a longer baseline run, is also wrong.

ORDERED CORRECTION

Rewrite around the measurement, which is far more damaging than the arithmetic: 'Your file holds 946,728 distinct seconds — your eleven days exactly. The other 494,991 rows are the same seconds written twice. You printed 0.356 duplicates and read it as a data-quality nuisance rather than as evidence that the mirror is not the release, and your section-3 invariant table then instructs the reader to treat that corrupted count as the expected value.'

PIT503 records a plant setting, not an attacker

EVIDENCE

Cell twenty reports the strongest single feature as PIT503 at an alone-AUC of 0.9002. The probe's independent importance ranking names FIT503 and MV101. All three are stage-five and stage-one instrument tags specific to this six-stage layout.

SCIENTIFIC CONSEQUENCE

A pressure transmitter reading at stage five is a property of this testbed's pump sizing, pipe diameter and setpoints. A second water utility with different hydraulics produces different values for identical attacker behaviour. So a model built on raw tag values learns a plant, and the claim that it detects attacks does not follow.

REQUIRED REPAIR

Re-express the signals as physics-relative residuals against a process model, ablate the raw configuration-coupled tags, and report the score on residual features only.

DEFENSIBLE REPLY

Unlike a flow-meter artefact in a NetFlow corpus, a pressure transmitter reading is genuine physical evidence of the process state, and process attacks do show up in it, so calling it a configuration artefact overstates the case.

The right words about overlap, and no measurement of it

EVIDENCE

Cell twenty-two prints the headline at 0.999979, the de-duplicated variant at 0.999987 and the shortcut-dropped variant at 0.999981. The conclusion says, quote, the separability is multi-feature. That reflects how this corpus was generated, not one leaky column.

SCIENTIFIC CONSEQUENCE

The sentence is correct and it is unsupported. If many columns each separate the classes independently, the attack and normal distributions barely overlap, and the size of that overlap is the quantity that decides whether the problem is hard. You asserted the interpretation and printed no number for it, so the reader has your reading of the ablation and no way to check it.

REQUIRED REPAIR

Estimate the class overlap directly: nearest-opposite-class distance distribution, or a Bayes-error estimate, and report it as the quantity the multi-feature claim rests on.

DEFENSIBLE REPLY

The notebook is already reading the ablation in the harder direction rather than the flattering one, which is more than the ablation literature usually asks, and the overlap measurement is a separate research question rather than a missing step.

A per-family recall plot with one bar

EVIDENCE

Cell eleven sets family equals numpy dot where, y equals one, attack, normal. Cell eighteen prints worst per-family recalls, open brace, attack, 0.978, close brace, one entry. The panel is titled Per-family recall, worst first, red below 0.5, and shows a single bar.

SCIENTIFIC CONSEQUENCE

Across this series the per-family recall is the diagnostic that reveals whether rare and hard attack types are actually detected. On this notebook it has been defined into vacuity. The plot title promises a breakdown, the axis promises a comparison, and the content is one number the reader already had.

REQUIRED REPAIR

Obtain the per-scenario attack labels from the iTrust release, which documents thirty-six attack scenarios, and report recall per scenario. Where the labels are unavailable, print unscorable rather than substituting the label for itself.

DEFENSIBLE REPLY

The notebook discloses the collapse three separate times, in cell zero, cell four and cell seventeen, and never claims the single-entry dictionary carries information beyond overall attack recall.

A letter grade on a boundary you chose

EVIDENCE

Cell twenty computes the grade with fixed thresholds: F at 0.999, D at 0.99, C at 0.95, B at 0.85 for the single-feature AUC, and F at 0.5, D at 0.3, C at 0.15, B at 0.05 for contamination. The printed values are 0.9002 and 0.062. The contamination figure sits 0.012 above the B-to-A boundary at 0.05. The conclusion reports, quote, data-trust grade B, and calls it the worse of two independent sub-checks.

SCIENTIFIC CONSEQUENCE

The contamination estimate is one number from one split with no interval attached, and it sits close enough to a cutoff that a different draw could move the letter. A grade near a boundary on an uncertain quantity conveys less information than the quantity itself, and the letter is what the conclusion repeats.

REQUIRED REPAIR

Report contamination and single-feature AUC with confidence intervals and drop the letter, or state explicitly the sampling spread of each quantity relative to the nearest cutoff.

DEFENSIBLE REPLY

Read the grading code. $\text{grade} = \max(\_ga, \_gc)$, and here $\_ga$ is B from a single-feature AUC of 0.9002 and $\_gc$ is B from contamination of 0.062. Move contamination below 0.05 and $\_gc$ becomes A; $\max('B', 'A')$ is still B. The boundary proximity your charge is built on cannot move this letter, because the other sub-check independently pins it. And CORRECTION 6's arithmetic runs the same way on the half you did aim at: the eight-seed sweep spans 0.8953 to 0.9102, a width of 0.0149, against 0.0498 to the nearest cutoff — three times the headroom, with no flip in eight of eight.

ORDERED CORRECTION

Drop the instability claim and keep the durable method complaint: hard cutoffs applied to two interval-free point estimates are not a measurement, and the letter is what the conclusion repeats. Do not argue that a different draw moves it — on this notebook it demonstrably cannot.

One point four million loaded, one hundred twenty thousand trained

EVIDENCE

Cell sixteen prints: materialized 1,441,719 rows, trained on 120,000 stratified subsample, held out 360,430. The subtitle in cell one reads 120,000 training rows, 360,430 held out of 1,441,719 ICS samples. The cap is a hard constant in cell sixteen, if length of Xtr greater than 120,000.

SCIENTIFIC CONSEQUENCE

The scale claim describes the download, not the experiment. Eight percent of the corpus reached the learner, and there is no learning curve anywhere in the notebook showing that the score has saturated. So the reader is offered a million-row corpus as evidence for a number produced on a twelfth of it.

REQUIRED REPAIR

Plot a learning curve from ten thousand rows to the full corpus and show saturation, or quote the effective training size in the abstract rather than the corpus size.

DEFENSIBLE REPLY

Cell sixteen carries an explicit comment stating that the notebook does not train on all N, that every headline is a subsample number, and that saying otherwise would be the fabrication the course forbids, and it prints both figures on the same line.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 360,430 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. So here is where this leaves us. The most honest paragraph in this notebook is the one that tells the reader the data should not exist, and the most consequential number is one your own conclusion says would disrupt a water utility fifteen times a day. Everything between those two is a measurement of how similar one second of a plant is to the second beside it. Eight out of thirty. Reject. Bring me a temporal split, a per-scenario label set from iTrust, and the alarms-per-day figure, and this becomes a real result about a real plant.

In-Vehicle CAN Intrusion Detection on OTIDS (Second Vehicle)

Three point seven four million in-vehicle CAN bus frames from the HCRL OTIDS captures, eleven frame fields, fifty-nine point nine percent labelled attack.

HEADLINE

XGBoost, held-out ROC-AUC 0.954777 and accuracy 0.858807 on the 936,011-row held-out split, with a three-fold cross-validated ROC-AUC of 0.9521.

DEFENSE

F

Selected-model estimand

DOCTORAL

7

/30 · REJECT WITH PREJUDICE

PRAXIS RUBRIC

8

/10 · self F

D1 1 External validity	D2 2 Statistical inference	D3 1 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 1 Honesty / scope

- 23 live faults
- 6 critical (FATAL)
- 17 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

This notebook is the most candid of the four I was given, and I want that on the record before I take it apart. It self-assigns a grade of F, correctly, and says out loud that the number should be read as a benchmark figure and not a deployment estimate. It refuses to assert a file-to-attack mapping it cannot verify. And in section eleven it explains, unprompted, that its own de-duplicated ablation row resolves every label conflict toward attack and is therefore an upper bound on an easier task rather than a corrected headline. That is a level of self-incrimination I almost never see. It is also, in the end, exactly the problem.

EXAMINER RULING

Eighty-three percent of your held-out set is a byte-for-byte copy of a training row, your positive-class rate is a constant you typed into the loader, and the impersonation attack you built the whole notebook around is one you decline to say which capture contains, so there is no quantity here that survives contact with the data.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

A random split on a bus that repeats by design

EVIDENCE

Cell twenty prints an exact-duplicate row rate across the whole corpus of 0.903 and a train-test exact-row contamination of 0.830. Cell sixteen splits with `train_test_split`, test size a quarter, stratified on the label, shuffled. The notebook's own grade line reads: train-test exact-row overlap 0.830 scores F.

SCIENTIFIC CONSEQUENCE

A CAN bus is periodic by construction. An engine control unit emits the same arbitration identifier with a near-identical payload at a fixed interval, several hundred times a second, for the entire life of the vehicle. Exact duplicate frames are not a data-quality accident on this medium. They are the medium. So a random split is not a weak protocol here, it is a structurally invalid one: it guarantees that ninety percent of the corpus can appear on both sides. Ninety percent of the held-out set being memorised means the reported 0.954777 is a nearest-neighbour lookup score, not a detection score. There is no reading of a held-out metric that survives a test set which is five sixths copies of training rows.

REQUIRED REPAIR

Split by capture and by contiguous time block, never at random. Hold out an entire capture window that shares no frame with training, and report the score there. Deduplicate on the frame key before splitting, and report how many unique frames actually exist.

The attack rate is a number you typed

EVIDENCE

Cell eleven keeps every attack frame and then subsamples the normal frames with a ceiling of one million five hundred thousand, written as one underscore five zero zero underscore zero zero zero in the sample call. Cell eleven prints a loaded total of 3,744,041 frames and an attack rate of 0.5994. Do the subtraction on screen: 3,744,041 minus 1,500,000 is 2,244,041 attack frames. Divide 1,500,000 by 3,744,041 and you get 0.4006. One minus that is 0.5994, to four decimals, which is exactly the printed rate, and it matches the independent probe's positive rate of 0.59936 to five decimals.

SCIENTIFIC CONSEQUENCE

The baseline line in cell sixteen tells the reader that any model must beat 0.5994, not 0.5, to be interesting. That bar is not a fact about vehicle traffic. It is one minus one-point-five million over the loaded total, and the one-point-five million is a constant chosen to satisfy the series' one-million-row floor. Change that constant and every accuracy figure in the ranking table moves, including the margin the notebook reports over baseline. On a real CAN bus the injection rate is a small fraction of a percent, so the notebook has constructed a task in which attack is the majority class and then reported the accuracy of clearing a majority bar it manufactured.

REQUIRED REPAIR

Report the archive's true class balance, evaluate at a realistic injection prevalence, and state the baseline as a function of the sampling constant rather than as a property of the corpus. If the row floor forces subsampling, subsample both classes proportionally and say so in the abstract.

You deleted the signal the dataset exists for

EVIDENCE

Cell eleven's drop list reads: DROP equals TS, target, y, family. The reference list in cell twenty-six cites Lee, Jeong and Kim, 2017, whose title is OTIDS, a novel intrusion detection system for in-vehicle network by using remote frame. Section three of the notebook says it plainly: because we drop timing, the offset-ratio and time-interval signal OTIDS was built around is absent.

SCIENTIFIC CONSEQUENCE

An impersonation attack on CAN is a node sending a legitimate arbitration identifier with a plausible payload. By construction, its identifier field and its eight payload bytes look like the genuine node's. The only thing that separates them is when the frame arrives relative to the genuine node's period. You removed that column and then reported 0.954777 as CAN intrusion detection. Whatever the model is separating on the impersonation capture, it cannot be impersonation, because you deleted the evidence of it before the model was fitted.

REQUIRED REPAIR

Retain the timestamp, derive the inter-arrival interval and the remote-frame response offset ratio per arbitration identifier, and report per-attack recall with those features present. If timing is genuinely to be excluded, restrict the claim to the DoS and fuzzy captures and say the impersonation result is not attempted.

One alarm for every six legitimate frames

EVIDENCE

Cell eighteen prints: operational false-positive rate at 0.5 equals 0.1586, fifty-nine thousand four hundred seventy-two benign flagged of three hundred seventy-five thousand.

SCIENTIFIC CONSEQUENCE

Do the division out loud. Three hundred seventy-five thousand divided by fifty-nine thousand four hundred seventy-two is six point three. This detector raises an intrusion alarm on approximately one out of every six legitimate CAN frames. Applied to the corpus's one and a half million benign frames, that is two hundred thirty-seven thousand nine hundred false alarms. A CAN bus carries hundreds of frames per second in normal operation, so this is not a detector with an inconvenient alert volume. It is a detector that is alarming continuously. That figure sits in the conclusion two sentences from the ROC-AUC and is never converted into anything a vehicle engineer could act on.

REQUIRED REPAIR

State the false-alarm count per second of bus time at the reported threshold, set an alert budget compatible with an in-vehicle response, and select the threshold from that budget rather than from the default of one half.

The abstract names impersonation, section three refuses to

EVIDENCE

The abstract in cell one says each frame is labelled normal or one of three attack captures, open parenthesis, DoS, fuzzy, impersonation, close parenthesis, and calls impersonation the crucial addition that Car-Hacking lacks. Section three says: we label them by capture id, attack underscore capture underscore one, two, three, rather than assert a file-to-attack mapping we cannot verify. Cell eighteen prints worst per-family recalls, attack_capture_3 at 0.809, attack_capture_1 at 0.896, attack_capture_2 at 0.944.

SCIENTIFIC CONSEQUENCE

The abstract sold the room a reason to read this notebook rather than notebook thirty-two, and the reason was impersonation. Section three then withdraws the only thing that would let anyone check it. Your weakest capture sits at 0.809, and nobody in this room, including you, can say whether that is the impersonation attack, the fuzzy attack or the DoS. Two claims that cannot both be true appear in the same document, and the one in the abstract is the one that will be quoted.

REQUIRED REPAIR

Establish the file-to-attack mapping from the HCRL release documentation and cite it, then report recall per named attack type. If the mapping genuinely cannot be established, remove the impersonation claim from the abstract, because a capability you cannot locate in your data is not a capability you can advertise.

DEFENSIBLE REPLY

Read section 3's first clause, not only its second. It says 'the three attack captures map to OTIDS's DoS, fuzzy and impersonation attacks' and then declines to say which file is which. The abstract claims impersonation is present in the corpus; section 3 agrees it is present and refuses to assign it to a numbered file without a citable source. Those two statements can both be true and both are. What survives, and I accept it, is narrower than 'two claims that cannot both be true': the weakest capture at 0.809 cannot be named, so the notebook cannot say whether its worst result is the impersonation attack it was written to study.

ORDERED CORRECTION

Rewrite as the unnameable-result charge, which is true and sufficient: 'Your weakest capture sits at 0.809 and nobody in this room, including you, can say which attack it is. The notebook exists because of impersonation and cannot locate impersonation in its own results table.' Drop 'two claims that cannot both be true' — section 3 affirms the attack set and withdraws only the file mapping.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You wrote the one-line check and did not run it

EVIDENCE

Section eleven, cell twenty-one, says: a vector that appears in both an attack capture and the normal capture therefore survives as the attack row, its benign twins are deleted, not reconciled. It then says its size is unmeasured, and supplies the check verbatim: `print, len of X dot drop_duplicates, comma, len of X dot assign underscore y equals y dot drop_duplicates`. That line appears nowhere in any executed cell.

SCIENTIFIC CONSEQUENCE

The disagreement rate is the irreducible error floor of this task. If the same eleven-field frame appears in both a normal capture and an attack capture, no frame-level classifier can be right about both, and the size of that set bounds every accuracy number in the notebook from above. You identified the quantity, wrote the code, described what the two counts would tell you, and left it unrun. That is not an oversight. That is a decision, and it is the same decision as declining the temporal split.

REQUIRED REPAIR

Run the two-count comparison, report the conflicted-key count as a percentage, and quote the resulting Bayes error floor alongside the headline accuracy.

An ablation row you know is invalid, printed anyway

EVIDENCE

Cell twenty-two prints the ablation under the heading, how much of the headline survives once each artifact is removed: headline as-is 0.954777, de-duplicated with ninety percent of rows removed 0.996659, shortcut feature dropped 0.946496. Section eleven states that the de-duplicated AUC is an upper bound, not a corrected headline, and instructs the reader not to read it as the headline with contamination removed.

SCIENTIFIC CONSEQUENCE

A number placed in a table headed survives artifact removal will be read as the score with the artifact removed. You knew that, which is why you wrote three paragraphs telling the reader not to. Those paragraphs are in the prose. The 0.996659 is in the table, and it is the largest number on the page. The unbiased version, resolving each duplicate key by majority label or dropping conflicted keys outright, is named in section eleven and explicitly not run. Prose cannot un-print a table row.

REQUIRED REPAIR

Resolve conflicted keys by majority label or drop them, refit, and report that number in the table. Remove the invalid row or relabel it explicitly as an upper bound on a modified task.

DEFENSIBLE REPLY

The table row is labelled 'de-duplicated (90% rows removed)' — the disqualifying fact is printed inside the row itself, not only in prose. And it is not only section 11 that disowns it: the validity ledger a reader is directed to says 'De-duplication raises the AUC, to 0.996659. That is not evidence the headline is safe... The de-duplicated task is therefore easier, not cleaner.' No abstract, no ranking table and no conclusion quotes 0.996659 as a result. The thing you actually want run — resolve conflicted keys by majority label and refit — is what fault 6 already demands, so this is that fault a second time with the table row attached.

ORDERED CORRECTION

Fold into fault 6 as its consequence, or keep it only as a presentation charge: the row belongs under a heading that says 'upper bound on a modified task', not under 'how much of the headline survives'.

TS was in the file and is in the drop list

EVIDENCE

Cell eleven: DROP equals TS, target, y, family. Cell sixteen splits at random with random_state zero. The section-thirteen conclusion states: the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

You removed the column, which proves you had it. On a periodic bus, a time-ordered split is not merely a stricter protocol, it is the only protocol under which a held-out frame is genuinely unseen. Training on the first portion of each capture and testing on the last would have produced the honest number in about twenty lines, and the notebook instead writes down that it did not do this and reports 0.954777.

REQUIRED REPAIR

Order each capture by TS, train on the earlier portion, test on the later portion with a guard interval, and report the drop.

DEFENSIBLE REPLY

Fault 1 tells me to split by capture and by contiguous time block and to report the score there. This charge tells me to order each capture by TS and split earlier-from-later. That is the same experiment under a different name, and I only have to run it once. Charge it once.

ORDERED CORRECTION

Merge into fault 1. If you keep it separate, distinguish it: fault 1 is about contamination, this is about protocol, and you should say which of the two the 0.954777 is more sensitive to.

Six decimals, one seed, no interval

EVIDENCE

Cell ten sets `RANDOM_STATE` to zero with the comment single seed used everywhere. Cell twenty-four prints a three-fold cross-validated ROC-AUC of 0.9521 plus or minus 0.0008, computed on `cvX` equals `Xtr dot iloc, colon forty thousand`. Cell eighteen's per-capture recalls, 0.809, 0.896 and 0.944, carry no intervals.

SCIENTIFIC CONSEQUENCE

The three per-capture recalls are the notebook's honest lead numbers and the conclusion nominates the weakest of them, 0.809, as where detection is thinnest. Without an interval on each, the reader cannot tell whether 0.809 and 0.896 differ. And the cross-validation figure describes a slice of training data, not the 936,011-row held-out set the abstract quotes.

REQUIRED REPAIR

Bootstrap the held-out AUC and every per-capture recall, report intervals, and repeat across at least ten seeds.

DEFENSIBLE REPLY

You ask for a binomial confidence interval on each per-capture recall. Apply it. The held-out set carries 561,011 attack frames across three captures, so the thinnest capture still supplies on the order of 150,000 positives; the binomial standard error at a recall of 0.809 is about 0.001. The gaps between 0.809, 0.896 and 0.944 are 0.087 and 0.048 — forty to eighty standard errors. The instrument you prescribe confirms the ordering rather than questioning it. What I concede is the other half of your charge and it is real: the 0.9521 plus-or-minus 0.0008 is a 3-fold cross-validation on 40,000 rows of the training subsample and describes nothing about the 936,011-row held-out set the abstract quotes.

ORDERED CORRECTION

Drop the 'cannot tell whether 0.809 and 0.896 differ' claim — it is arithmetically false. Keep the CV-scope half, and if you want an interval charge that bites, make it the one your fault 1 implies: with 83% of held-out rows duplicated from training, the effective sample size behind every one of these recalls is unknown and far below n .

You plotted the right curve and took no area

EVIDENCE

Cell eighteen imports `precision_recall_curve`, calls it on the winning model's held-out probabilities, and plots it in the third panel titled Precision-Recall. The function `average_precision_score` appears nowhere in the notebook. Cell sixteen sorts the ranking table on `roc_auc` and prints best model, XGBoost.

SCIENTIFIC CONSEQUENCE

At a false-positive rate of 0.1586 the precision of this detector is the number that decides whether it is deployable, and it is sitting in a figure that nobody read. Rendering the correct diagnostic into the PDF and then selecting on a different metric leaves the reader with the impression that the question was addressed.

REQUIRED REPAIR

Compute average precision with an interval, report precision at a fixed recall, and rank on that.

DEFENSIBLE REPLY

Attacks are 59.94% of this corpus, so the classic imbalance argument for average precision does not apply here and you are right not to make it. But the number you say nobody read is recoverable from what is printed: 375,000 benign held out, 59,472 false positives, accuracy 0.858807 on 936,011 rows gives 488,332 true positives, so precision at threshold 0.5 is 0.8914 and recall is 0.8705. Average precision would be high, because the positive class is the majority — which is exactly why it is not the diagnostic that matters here. The one that matters is the false-positive rate, it is printed at 0.1586, and you have already charged it as fault 5.

ORDERED CORRECTION

Per CORRECTION 2, on an attack-majority corpus the defensible version is different: the PR panel is drawn with the majority class as positive, so it says nothing about the rare class, and no operating point for the rare class is ever established. State it that way or fold it into fault 5.

11

MAJOR

UNTESTED

DOWNGRADED

THRESHOLD FIXED AT 0.5, NO COST MODEL

A safety-critical operating point left at the default

EVIDENCE

Cell eighteen: pred equals pb greater than 0.5. The false-positive rate is reported at 0.5. No cost matrix appears anywhere in cells ten through twenty-six.

SCIENTIFIC CONSEQUENCE

The consequences of a missed CAN injection and of a spurious in-vehicle alarm are not on the same scale and are not symmetric, which is the entire reason automotive intrusion detection is a discipline. The threshold that trades between them is the design decision, and it was taken by a default argument.

REQUIRED REPAIR

State a cost matrix appropriate to a vehicle network, choose the threshold by minimum expected cost, and report the full operating curve rather than a single default point.

DEFENSIBLE REPLY

One fix, two charges. Fault 5 asks me to set an alert budget and pick the threshold from it; this asks me to write a cost matrix and pick the threshold from that. They are the same cell.

ORDERED CORRECTION

Fold into fault 5, keeping the cost-matrix requirement as the remedy.

12

CRITICAL

UNTESTED

VERIFIED

NO MODEL SELECTION

No search, and the cross-validation runs after the choice

EVIDENCE

Cell sixteen hand-sets XGBoost at eighty estimators and depth six, RandomForest at sixty trees, LightGBM at eighty estimators, with no grid or random search. Cell sixteen then prints best model, XGBoost, chosen on the held-out ROC-AUC. Cell twenty-four clones that winner and cross-validates it on the first forty thousand training rows.

SCIENTIFIC CONSEQUENCE

The selection happened on the held-out set and the check happened on the training set. That is backwards. The cross-validated 0.9521 therefore inherits the optimism of a choice it is supposed to validate, and no number in the notebook is an unbiased estimate of generalisation for a selected model.

REQUIRED REPAIR

Nested cross-validation, with an inner loop selecting the learner and its hyperparameters and an outer loop estimating on folds the inner loop never saw.

One draw, and re-split noise asserted without measuring it

EVIDENCE

Cell ten: `RANDOM_STATE` equals zero, single seed used everywhere. The section-thirteen conclusion says each ablation re-splits and refits, so tiny differences are re-split noise. In fact the shortcut-dropped row in cell twenty-two calls `train_test_split` with the same seed, the same row count and the same label vector as the headline, which returns the identical partition, so nothing is re-split there at all.

SCIENTIFIC CONSEQUENCE

The word noise is doing inferential work that no measurement supports. For the shortcut row the split is unchanged, so the difference between 0.954777 and 0.946496 is not noise but the actual effect of removing ID1. For the de-duplicated row the split genuinely differs, but by ninety percent of the corpus, which is far past anything a noise term describes. One blanket sentence is applied to two rows with entirely different causes.

REQUIRED REPAIR

Estimate the seed-to-seed spread by rerunning the pipeline across at least ten seeds, then use that measured spread to say which ablation differences are real.

DEFENSIBLE REPLY

Your identical-partition finding is right and I verified it — same `n`, same `y`, same `test_size`, same `random_state` returns the same `StratifiedShuffleSplit` partition, so the shortcut row is not re-split. But the sentence you are prosecuting says 'tiny differences are re-split noise', and the notebook does not call the ID1 effect tiny: the ledger reports it separately as '0.954777 to 0.946496' and reads it as a real multi-feature result. The sentence is loose, not false, on the row you care about. Strip the identical-partition observation out and what is left is 'one seed, no interval', which is fault 9.

ORDERED CORRECTION

Keep only the identical-partition finding and make it precise: 'You tell the reader each ablation re-splits. One of the two does not. Say which sentence covers which row, or delete the sentence.' Drop the seed half into fault 9.

The adversary here controls the input directly

EVIDENCE

A search of this notebook's full source for the words adversarial, evasion, evade and perturb returns nothing. The reference list in cell twenty-six cites Koscher and colleagues, 2010, on experimental automotive compromise, and applies none of it.

SCIENTIFIC CONSEQUENCE

On CAN there is no measurement layer between the attacker and the features. The attacker chooses the arbitration identifier and chooses all eight payload bytes. The feature vector is the attack. An adversary who reads your top-importance plot can construct frames that sit inside the benign region at literally no cost, because there is nothing else to construct. A detector on this medium without an evasion budget has reported only its best case.

REQUIRED REPAIR

Evasion under a stated frame-construction budget, with worst-case per-capture recall reported next to the average-case figure.

DEFENSIBLE REPLY

There is no evasion experiment and I will not pretend otherwise, and I concede your framing is sharper here than anywhere else in the series: on CAN there is no measurement layer, the attacker chooses the arbitration identifier and all eight payload bytes, so the feature vector is the attack. What I resist is the severity. The notebook claims per-frame separability on three captures and explicitly refuses to be read as a deployment estimate; adversarial robustness is a claim it never makes, and a claim never made cannot be falsified.

ORDERED CORRECTION

Keep the CAN-specific argument, which is genuinely the best version of this charge in the set, and drop the general 'a detector without a worst case has no worst case' framing that lands identically on all thirty-six.

The companion vehicle is in the same folder

EVIDENCE

The section-thirteen conclusion says cross-vehicle transfer between this dataset and nb32, in parentheses unmeasured here, is the other. The abstract calls this notebook a cross-dataset companion to nb32. Notebook thirty-six, cell twenty, fits one model per corpus and scores every model on every corpus, printing a four-by-four matrix.

SCIENTIFIC CONSEQUENCE

The abstract calls itself a cross-dataset companion and then measures nothing across datasets. Both corpora are CAN frames with an identifier and eight payload bytes, so the schemas already align. The harness exists, the second corpus exists, the notebook says the experiment matters, and it was not run. Naming it in the conclusion does not substitute for two extra cells.

REQUIRED REPAIR

Fit on OTIDS and score on Car-Hacking, and fit on Car-Hacking and score on OTIDS, on the shared identifier-and-payload representation, and report both directions.

DEFENSIBLE REPLY

Notebook 36's twenty-line loop works because its four parquets carry byte-identical 43-column schemas — I checked them, and its own loader comment says that is precisely what makes train-on-one, test-on-another possible. Now compare the two CAN corpora as they actually sit on disk. OTIDS is a headed CSV with TS, ID1, ID0, LEN and DLC0 through DLC7 — the arbitration identifier split across two columns, eight fixed payload columns, and a numeric target code. Car-Hacking is headerless, with a timestamp, a single hex identifier field, a DLC, then DLC-many payload bytes so the row width varies from record to record, and an R/T flag as the label. Different identifier encoding, variable row width against fixed, and a different label representation. That is a feature-alignment contribution, not a copied cell. What I do concede without qualification is that the abstract calls this a cross-dataset companion to nb32 and then measures nothing across datasets.

ORDERED CORRECTION

Lead with the abstract, not the harness: 'You call this a cross-dataset companion to notebook 32 and report no quantity computed across the two datasets.' That charge needs no schema claim and cannot be deflected.

You cite a timing method and report a timing-free number

EVIDENCE

Cell twenty-six cites Lee, Jeong and Kim, 2017, OTIDS, a novel intrusion detection system for in-vehicle network by using remote frame, and Song, Woo and Kim, 2020, in-vehicle network intrusion detection using deep convolutional networks. The notebook drops TS in cell eleven and reports a per-frame binary ROC-AUC of 0.954777 on a random split.

SCIENTIFIC CONSEQUENCE

The cited works evaluate on time-window sequences with per-attack precision and recall. There is no published in-vehicle result in that form against which a per-frame ROC-AUC on a shuffled ninety-percent-duplicated corpus can be compared. Your number cannot be placed above, below or beside any prior work, which means it does not enter the literature at all.

REQUIRED REPAIR

Report per-attack precision, recall and F-measure on the captures under the evaluation protocol used by Lee and colleagues, so a reader can place the result in a table.

DEFENSIBLE REPLY

The notebook never claims comparability to the OTIDS protocol, states plainly that timing is dropped and that the offset-ratio signal the paper was built around is therefore absent, and cites Lee and colleagues as the dataset source rather than as a benchmark it is competing with.

A different vehicle, on no stated authority

EVIDENCE

The abstract in cell one states it was captured on a different vehicle than the nb32 Car-Hacking set. Cell eleven's loader comment repeats it. No printed output, no probe value and no cited page in cell twenty-six supports it. The corpus is pulled from a Kaggle re-upload, bikashkundu slash can-hcrl-otids, not from the HCRL distribution.

SCIENTIFIC CONSEQUENCE

Different vehicle is what makes this a second-vehicle study rather than a second capture of the same one, and it is the whole justification for the notebook's title. It is asserted twice and evidenced nowhere, on a corpus obtained from a mirror.

REQUIRED REPAIR

Cite the HCRL release documentation for the vehicle model of each capture, or restate the claim as a second capture set from the same laboratory.

DEFENSIBLE REPLY

The HCRL laboratory does publish the vehicle used for each of its CAN releases, so the claim is checkable in the primary source even though the notebook does not carry the citation.

18

MAJOR

MEASURED

VERIFIED

MODEL RANKING WITHOUT A TEST

A winner named without a paired comparison

EVIDENCE

Cell sixteen: XGBoost 0.954777, LightGBM 0.952706, RandomForest 0.951774. The gap between first and second is 0.002071. The line print, best model, best_name selects on that ordering, and every diagnostic in cells eighteen, twenty, twenty-two and twenty-four is conditioned on the winner.

SCIENTIFIC CONSEQUENCE

Three learners within three thousandths of each other are declared ranked, and the whole results section descends from the top row. Nothing establishes that a second draw would produce the same ordering, and the notebook never draws a second sample.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of XGBoost and LightGBM, or overlapping bootstrap intervals on the AUC difference, before any model is named best.

DEFENSIBLE REPLY

On a held-out set of 936,011 rows the standard error of a ROC-AUC estimate is small enough that a gap of 0.002 is very likely a real ordering, so the missing test would probably have confirmed the winner rather than overturned it.

19

MAJOR

MEASURED

VERIFIED

SELF-ASSIGNED TRUST GRADE IS NOT A MEASUREMENT

An F that changes nothing downstream

EVIDENCE

Cell twenty prints: data trust grade F, worse of the two, F equals shortcut and slash or heavy contamination. The conclusion says on this A-best, F-worst scale, a D or F means the headline is optimistic, treat it as a benchmark number, not a deployment estimate. The abstract in cell one nonetheless leads with model comparison and validity audit and quotes no grade, and cell sixteen reports 0.954777 without qualification.

SCIENTIFIC CONSEQUENCE

The grade is a letter derived from two point estimates against cutoffs the notebook chose, and its only consequence is a sentence in section thirteen. It changed no number, retracted no claim, and did not reach the abstract. A grading scheme whose failing mark leaves the output unchanged is a rhetorical device, not an audit.

REQUIRED REPAIR

Report contamination with an interval and drop the letter, and let a failing check gate publication of the headline rather than annotate it.

DEFENSIBLE REPLY

The grade is at least an accurate and voluntarily disclosed self-assessment computed by executable code, which is more than the great majority of published intrusion-detection benchmarks provide.

Gini importance on eleven correlated frame fields

EVIDENCE

Cell eighteen reads `imp equals best dot feature_importances_` and plots the top twelve under the title `top importances slash absolute coefficient`. The independent probe ranked `LEN` as its top column while the notebook's own single-feature audit in cell twenty named `ID1`, and both are among eleven strongly correlated frame fields.

SCIENTIFIC CONSEQUENCE

The arbitration identifier is a high-cardinality field and the payload bytes are correlated with it, which is precisely the configuration in which Gini importance misallocates credit. The plot is presented in the results section as an account of what the model uses, and it is not evidence for that.

REQUIRED REPAIR

Permutation importance computed on the held-out set, or SHAP values, reported instead of impurity importance.

DEFENSIBLE REPLY

The notebook never claims the importance panel is a causal or complete explanation; it is one of four diagnostic panels and the ablation, not the importance plot, is what carries the argument.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 936,011 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

De-duplication removed most of my corpus here, so the movement you see is dominated by a real effect, not by split noise. The criticism still applies — I never measured the nuisance term — but it is at its weakest on this notebook, not its strongest.

Closing. Seven out of thirty. Reject with prejudice. And I want to be precise about why this one lands lowest of the four rather than the one with the least candour, because that will feel unfair. It lands lowest because the quantity it reports is an artifact of the file in the strictest sense the band describes. The positive rate is a constant from the loader, the held-out set is five sixths a copy of the training set, and the attack the notebook was written to study is one it declines to locate. Candour about all three is on the page, and candour is not a result. Run the leave-one-capture-out split, keep the timestamps, and cite the file mapping, and this becomes the second-vehicle study the title already promises.

Android Malware Detection & Category Characterization (CCCS-CIC-AndMal-2020)

Three hundred fifty-seven thousand eight hundred and five Android applications described by nine thousand four hundred and seventeen static features, fifty-four point six seven percent malware across fourteen categories plus benign.

HEADLINE

RandomForest, held-out ROC-AUC 0.997291 and accuracy 0.980716 on 89,452 held-out apps, with the de-duplicated figure of 0.984201 named in the conclusion as the honest number.

DEFENSE

F

Selected-model estimand

DOCTORAL

10

/30 · REJECT

PRAXIS RUBRIC

10

/10 · self F

D1 1 External validity	D2 2 Statistical inference	D3 2 Construct validity	D4 1 Decision quality	D5 2 Reproducibility	D6 2 Honesty / scope

- 23 live faults
- 4 critical (FATAL)
- 19 major (CONTESTED)
- 0 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

Two things here are genuinely better than the series average and I will say them first. This notebook does not stop at disclosing its contamination. It measures it at zero point five one four, grades itself F on that exact check, refits on a de-duplicated corpus with fifty-two percent of rows removed, and then names the corrected 0.984201 as the honest number in its own conclusion. That is a re-derivation under a stricter protocol, executed and adopted, and only one other notebook I was given does anything comparable. It also reports recall per malware category rather than one aggregate, and it puts the un-categorised bucket at 0.806 in front of the reader instead of burying it.

EXAMINER RULING

You told the room in section three that this dataset defines a hundred and ninety-one malware families and that you do not use them, and then you split the corpus at random, which puts members of the same family on both sides and turns your detector into a family recogniser with a ninety-nine point seven printed on it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

Nine thousand features and not one selection step

EVIDENCE

Cell eleven prints: loaded 357,805 Android apps by 9417 static features. Cell sixteen caps training at 120,000 rows and instantiates RandomForest with sixty trees, XGBoost with eighty estimators at depth six, LightGBM with eighty estimators, all hand-set, with no grid and no search. Cell sixteen then prints best model, RandomForest, selected on the held-out ROC-AUC. Cell twenty-four clones that winner and cross-validates it on cvX equals Xtr dot iloc, colon forty thousand.

SCIENTIFIC CONSEQUENCE

At roughly twelve training rows per feature, and with a forest whose default considers about ninety-seven of the nine thousand four hundred and seventeen columns at each split, the reported model is one arbitrary point in an enormous model space that was never searched. And the ordering is inverted: the winner was chosen on the held-out set and then checked against the training set. That makes the cross-validated 0.9951 a description of a decision already taken rather than an independent estimate of it, so the notebook contains no unbiased estimate of generalisation for the model it reports. In a nine-thousand-dimension regime, any selection performed outside a nested loop is optimistically biased by construction, and here there is no loop at all.

REQUIRED REPAIR

Nested cross-validation, with feature selection and hyperparameter search entirely inside the inner loop and the generalisation estimate taken from outer folds the inner loop never saw. Report the outer estimate as the headline.

Identical vectors in nine thousand dimensions are the same app

EVIDENCE

Cell twenty prints an exact-duplicate row rate across the corpus of 0.519 and a train-test exact-row contamination of 0.514. Cell eleven establishes the feature count at 9,417. The independent probe reproduced the contamination at 0.5141 on the first window and 0.5138 on a random window, so it is not a windowing artefact.

SCIENTIFIC CONSEQUENCE

Two applications agreeing exactly across nine thousand four hundred and seventeen static features do not do so by coincidence. That is the same binary, or a repackaging of it with the manifest untouched. So the 0.514 is not a data-hygiene statistic. It is a direct measurement that fifty-one point four percent of your held-out apps are already in training under a different row index. The conclusion reads this the wrong way, saying that de-duplication only mildly inflates the score so the signal is real and a random split is merely optimistic. Merely optimistic is the wrong description of a test set that is half copies. And de-duplication does not repair it, because near-identical repacks, which differ in one permission flag, survive the `drop_duplicates` call untouched and are never counted.

REQUIRED REPAIR

Report the nearest-neighbour Hamming distance distribution from each held-out app to the training set, not just the exact-match count, and split on the application hash or the family so that no binary and no repack of it appears on both sides.

One in sixty-seven legitimate apps blocked

EVIDENCE

Cell eighteen prints: operational false-positive rate at 0.5 equals 0.0150, six hundred and eight benign flagged of forty thousand five hundred forty-six. The conclusion frames the task as stopping exactly the apps a store most needs to stop.

SCIENTIFIC CONSEQUENCE

Forty thousand five hundred forty-six divided by six hundred and eight is sixty-six point seven. This model wrongly blocks one in every sixty-seven legitimate applications. Applied to the benign apps in this corpus alone, that is roughly two thousand four hundred developers whose software is rejected. And the arithmetic gets worse when you leave the corpus, because this corpus is fifty-four point six seven percent malware and no application store is anything like that. At a realistic store prevalence the false positives outnumber the true positives by a wide margin, and the notebook's chosen headline metric is precisely the one that is blind to that shift. The number is printed. The consequence is never computed.

REQUIRED REPAIR

Multiply the false-positive rate by a realistic store submission volume at a realistic malware prevalence, state the blocked-legitimate-apps figure and the resulting precision, and choose the threshold from a review-capacity budget.

One hundred ninety-one families, named and declined

EVIDENCE

Section three's provenance table says, in the label row: the dataset also defines 191 finer families, which this notebook does not use, every per-group number here is category-level. Cell sixteen splits with `train_test_split`, test size a quarter, stratified on the binary label. The section-thirteen conclusion says the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

A malware family is a lineage of repackaged variants of the same code. Under a random split, variants of one family land in training and its siblings land in the held-out set. The model then has only to recognise the family, which is a memorisation task, and every held-out number becomes a measure of how many families straddle the split rather than of whether malice is detectable. This is not an abstract risk on this corpus: fifty-one point nine percent of the rows share an identical feature vector with another row, which is what a family lineage looks like in feature space. You wrote the grouping variable down in section three, on the same page as the row count, and then chose a protocol that ignores it.

REQUIRED REPAIR

`GroupShuffleSplit` or `GroupKFold` on the family identifier, so that no family appears on both sides, with the drop from 0.997291 reported. Where the chosen feature file does not carry the family column, join it from the CCCS-CIC release that defines the hundred and ninety-one families.

DEFENSIBLE REPLY

The grouping prescription needs a key. I read the file: `cicandmal2020-static.parquet` has 9,505 columns, F0 through F9503 and Label, and nothing else — no family, no SHA-256, no package name, no app id. `GroupShuffleSplit` has no argument to take. Your required clause anticipates this and says join the families from the CCCS-CIC release, but a join needs a key on both sides and this side has none; the rows are anonymous ordered feature vectors. And the 191 figure itself is prose in section 3 that is never printed and cannot be verified against this parquet. What survives, and I accept it, is corpus selection rather than split design: I chose a mirror that had thrown the grouping key away, and I said so in the same table where I said the families exist.

ORDERED CORRECTION

Re-mechanise to Tier 2: 'You told the room the dataset defines 191 families, then chose a feature mirror that strips every column by which an app could be identified — no family, no hash, no package name. The defect is not that you declined `GroupKFold`. It is that you selected a corpus on which no independence check of any kind is possible, and reported six significant figures from it.'

TESSERACT is in your bibliography and not in your code

EVIDENCE

Cell twenty-six lists reference four: Pendlebury and colleagues, 2019, TESSERACT, eliminating experimental bias in malware classification across space and time. The section-thirteen conclusion says a time-aware evaluation, Pendlebury et al., 2019, is the deployment-realistic test for a mobile-malware scanner. Section three repeats it: a random split ignores temporal concept drift, the honest generalization test we flag but do not run.

SCIENTIFIC CONSEQUENCE

TESSERACT exists because a random split on malware over-reports by a large margin, and its central result is that the over-reporting is not a rounding error. You cited the paper that says your protocol is invalid, restated its conclusion twice in your own words, and then reported the number the paper was written to discredit. The loader in cell eleven drops only the label, the derived y and the family column, and names no date field, so the notebook does not even tell the reader which column a time-aware split would order on. Naming the fatal limitation is not a defence at this level. It is a confession that the experiment was understood and declined.

REQUIRED REPAIR

Run the TESSERACT protocol: order applications by first-seen date, train on an earlier window, test on subsequent windows, and report the area under the time-decay curve rather than a single shuffled split. Identify and publish the date field used.

DEFENSIBLE REPLY

You ask me to identify and publish the date field I would order on. There is none. The parquet holds 9,505 columns — F0 through F9503 and Label — with no timestamp, no first-seen date, and no hash or package name from which a date could be looked up, so the field is not merely absent but unrecoverable from this file. TESSERACT is a protocol over time-stamped samples; it cannot be run on a corpus that carries no time. Your own evidence notices this and reads it as an aggravating factor — 'the notebook does not even tell the reader which column a time-aware split would order on' — when it is the reason the protocol is impossible. What I concede is the choice that precedes it: I picked a dateless mirror for a threat everybody agrees drifts, and I cited the paper that says so.

ORDERED CORRECTION

Re-mechanise from 'declined' to 'foreclosed by corpus selection': 'You cite the paper that proves your protocol invalid, then choose a feature mirror with no date column, which forecloses the corrective experiment before you begin. The defect is upstream of the split.' That charge is true and survives the missing column; the current one does not.

Atlas proof

docs/course/security-ml-notebooks/nb35-android-malware.ipynb · Cells 2, 4, 16, and 25; zero-based manifest cell IDs: 2=9b2e5564, 4=83dc9e5a, 16=5fc1fde7, 25=6e2ac6cc

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

The honest number is one scalar; every diagnostic came from the other model

EVIDENCE

Cell twenty-two's ablation prints headline as-is 0.997291, de-duplicated with fifty-two percent of rows removed 0.984201, shortcut feature dropped 0.997303. The conclusion says de-duplication does matter here, it lowers the AUC to 0.984201, so the headline is inflated by repeated rows and 0.984201 is the honest number. But cell sixteen's ranking, cell eighteen's false-positive rate of 0.0150 and per-category recalls of 0.806 through 0.971, cell twenty's grade of F and cell twenty-four's cross-validated 0.9951 are all derived from the fit on the contaminated corpus.

SCIENTIFIC CONSEQUENCE

If 0.984201 is the honest number, then the honest false-positive rate, the honest per-category recalls and the honest grade are the ones the de-duplicated model produces, and none of them exists in this notebook. Correcting one scalar in prose while leaving every downstream diagnostic computed from the model you have just disowned is the master fault of this whole series in miniature. The correction was affordable: it is one refit and a re-run of cells eighteen and twenty.

REQUIRED REPAIR

Refit on the de-duplicated corpus and regenerate the entire results section from that model: ranking, importances, per-category recalls, false-positive rate and cross-validation. Report the contaminated figures only as a contrast.

Half the rows collapsed and the conflict rate never counted

EVIDENCE

The section-thirteen conclusion states: the de-duplication variant keeps the first label when a feature vector appears twice. Cell twenty-two computes X_{dd} equals X dot drop_duplicates and takes y_{dd} equals y indexed by X_{dd} 's index, removing fifty-two percent of rows. No count of conflicting labels is printed anywhere in the notebook.

SCIENTIFIC CONSEQUENCE

With over half the corpus duplicated and the classes close to balanced at fifty-four point six seven percent malware, some of those repeated nine-thousand-dimension vectors will carry a benign label in one row and a malware label in another. That is a benign app and a malicious app that are indistinguishable under this feature set, and the rate at which it happens is a hard ceiling on any achievable accuracy, including the 0.984201 the conclusion calls honest. Keeping the first label silently deletes that ceiling instead of reporting it. The number that bounds every claim in the notebook was never computed.

REQUIRED REPAIR

Count unique feature vectors and unique vector-label pairs, report the difference as the conflicted-key rate, and quote the resulting irreducible error floor next to the headline.

DEFENSIBLE REPLY

You say the rate at which a benign and a malicious app share an identical vector is a hard ceiling on every accuracy in the notebook, including the 0.984201. I measured it on the parquet, reproducing the notebook's feature matrix exactly — 9,417 columns retained, exact-duplicate row rate 0.519 matching the printed 0.519 to three decimals. There are 172,278 unique feature vectors and 172,365 unique vector-label pairs. Eighty-seven conflicted keys. Three thousand two hundred and fourteen rows, nine tenths of one percent of the corpus, and the implied Bayes floor is under half a percent. It bounds nothing at 0.984 and nothing at 0.997. The mechanism you assign — near-balanced classes plus half the corpus duplicated must produce conflicts — is a plausible inference that the data does not support, and you can tell it is an inference because you levelled the identical charge at notebook 34, where the same measurement returns 11,266 conflicted keys and 37.0% of rows.

ORDERED CORRECTION

Keep only the true residual: the de-duplication resolves ties by keeping the first label, which is arbitrary, and the notebook never counts how often it fires. Then state the measured answer — 87 keys, 0.9% of rows — and say plainly that on this corpus the arbitrary rule costs nothing, which is a finding rather than a fault. Reserve the Bayes-floor argument for notebook 34, where it lands.

Six per-category recalls, no intervals

EVIDENCE

Cell ten sets `RANDOM_STATE` to zero with the comment single seed used everywhere. Cell eighteen prints worst per-family recalls: NoCategory 0.806, PUA 0.898, Zeroday 0.937, Backdoor 0.937, Scareware 0.96, Adware 0.971. Cell twenty-four's plus-or-minus 0.0006 is computed on the first forty thousand training rows and describes nothing in the held-out set.

SCIENTIFIC CONSEQUENCE

The conclusion builds its central argument on the ordering of those six recalls, saying the rarer and harder categories are where recall drops. Zeroday and Backdoor are printed as identical at 0.937, and the reader has no way to know whether 0.937 and 0.898 differ, because no interval is attached to any of them and no positive count is printed for any category. The one number the notebook nominates as its honest story is the one it makes undecidable.

REQUIRED REPAIR

Print the held-out positive count for each category, attach a binomial confidence interval to each recall, and bootstrap the held-out AUC.

DEFENSIBLE REPLY

The conclusion says the rarer, harder categories are where recall drops. It does not claim Zeroday differs from Backdoor — it prints them as identical, which is the honest rendering of two equal numbers. The claim that is made concerns the extremes, 0.806 against 0.971, and that gap is secure: 89,452 held-out apps at 54.67% malware give 48,906 positives across fourteen categories with none skipped by the count-5 floor, so the binomial standard error you prescribe is on the order of 0.005 against a gap of 0.165. What I concede is the omission itself: the held-out positive count per category should be printed, and it is not.

ORDERED CORRECTION

Charge the missing counts, not the undecidability: 'You report six recalls and no denominators, so a reader cannot compute the interval you did not supply.' Drop the claim that the notebook's central argument is undecidable — the argument it makes concerns the extremes and they are forty standard errors apart.

The right curve, rendered and unread

EVIDENCE

Cell eighteen imports `precision_recall_curve`, calls it on the winner's held-out probabilities, and plots it in the third panel under the title Precision-Recall. The function `average_precision_score` appears nowhere in the notebook. Cell sixteen sorts on `roc_auc` and prints best model, RandomForest.

SCIENTIFIC CONSEQUENCE

ROC-AUC is invariant to the class prevalence of the evaluation set, which sounds like a virtue and is here the defect. This corpus is fifty-four point six seven percent malware and a real store is nothing close, so an ROC-AUC of 0.997291 transports to a deployment setting while carrying no information about how many legitimate apps will be blocked there. Precision does carry that information, you computed the curve that shows it, you drew it into the figure, and you did not take its area or read a single point off it.

REQUIRED REPAIR

Report average precision with a bootstrap interval, report precision at a fixed recall, and re-report both at a realistic store prevalence rather than the corpus prevalence.

DEFENSIBLE REPLY

The malware rate here is 54.67%, so the positive class is the majority and the textbook imbalance argument does not apply — you are right not to make it, and the prevalence-transport version you make instead is correct. It is also fault 5's version: your fault 5 already requires me to re-report at a realistic store prevalence. And the precision you say nobody read is arithmetic on printed output: 40,546 benign held out, 608 false positives, accuracy 0.980716 on 89,452 rows gives 47,789 true positives, so precision is 0.9874 and recall 0.9772 at threshold 0.5. Average precision would be roughly 0.997. Taking the area would have confirmed the headline, not exposed it.

ORDERED CORRECTION

Fold into fault 5, or re-mechanise per CORRECTION 2 to the form that survives on an attack-majority corpus: the PR panel is drawn with the majority class as positive and therefore says nothing about the class a store actually needs bounded.

A gatekeeping decision taken by a default argument

EVIDENCE

Cell eighteen: pred equals pb greater than 0.5. The false-positive rate is reported at 0.5. No cost matrix appears anywhere in cells ten through twenty-six.

SCIENTIFIC CONSEQUENCE

The cost of admitting a banking trojan and the cost of rejecting a legitimate developer's application are not on the same scale, and the entire value of a scanner lies in where you place the point between them. That point was inherited from a scikit-learn default and never examined. It is not a detail downstream of the result. It is the result.

REQUIRED REPAIR

State an explicit cost matrix for a missed malware admission and a wrongly blocked benign app, select the threshold by minimum expected cost at a realistic prevalence, and report the score there.

DEFENSIBLE REPLY

Fault 5 tells me to choose the threshold from a review-capacity budget. This tells me to choose it from a cost matrix. Same cell, same refit, one defect.

ORDERED CORRECTION

Fold into fault 5 as its remedy.

One draw, six decimals, and noise asserted without measurement

EVIDENCE

Cell ten: `RANDOM_STATE` equals zero, single seed used everywhere. The conclusion says each ablation re-splits and refits, so tiny differences are re-split noise. In fact the shortcut-dropped row in cell twenty-two calls `train_test_split` with the same seed, the same row count and the same label vector as the headline, which returns the identical partition, so nothing is re-split there.

SCIENTIFIC CONSEQUENCE

The blanket sentence is false for one of the two ablation rows and irrelevant to the other. The shortcut-dropped comparison uses the identical split, so the move from 0.997291 to 0.997303 is the genuine effect of removing F50 rather than noise. The de-duplicated row does change the split, but by fifty-two percent of the corpus, which is not a noise term. Neither difference has a measured variance behind it, because no second seed was ever drawn.

REQUIRED REPAIR

Rerun the pipeline over at least ten seeds, report the mean and standard deviation of the held-out AUC, and use that measured spread to decide which ablation differences are real.

DEFENSIBLE REPLY

Your identical-partition finding is right and I verified it: same `n`, same `y`, same `test_size`, same `random_state` returns the same partition, so the shortcut-dropped row is not re-split and 0.997291 to 0.997303 is the measured effect of removing F50. But that effect is twelve millionths, and the conclusion's reading of it — that the separability is multi-feature — is exactly what a twelve-millionth effect licenses. Nothing in the notebook changes if the sentence is corrected. Strip that observation out and what remains is 'one seed, no interval', which is fault 8.

ORDERED CORRECTION

Keep the identical-partition correction as a one-line accuracy note and merge the rest into fault 8; do not carry it as an independent FATAL.

Packing and obfuscation named, never applied

EVIDENCE

The section-thirteen conclusion says the structural limits are that static features miss runtime behaviour, packing and obfuscation. That phrase is the only appearance of the concept in the notebook. No perturbation budget, no evasion attack and no worst-case recall appears in any cell.

SCIENTIFIC CONSEQUENCE

Android malware authors pack and obfuscate as routine practice, precisely to defeat static feature extraction, and the notebook says so. Naming that is not measuring it. A static scanner's worst case against an obfuscating adversary is the only number a store operator can plan against, and it is absent. Under this rubric, naming a limitation scores one point, never more, and this is exactly why.

REQUIRED REPAIR

Apply a stated obfuscation and repackaging budget to the held-out malware, re-extract features, and report worst-case recall alongside the average-case figure.

DEFENSIBLE REPLY

There is no evasion experiment and I concede your framing is at its sharpest here, because packing and obfuscation exist specifically to defeat static feature extraction and the notebook names them. But the claim the notebook makes is that static features separate malware from benign apps on this corpus, and adversarial robustness is a claim it never makes. Your own rubric line — that naming a limitation scores one point and never more — is a scoring rule, not a demonstration that the reported number is wrong.

ORDERED CORRECTION

Keep the packing argument, which is genuinely corpus-specific, and drop the general worst-case framing that separates none of the thirty-six.

Drebin is cited and no comparable number is produced

EVIDENCE

Cell twenty-six lists reference three: Arp and colleagues, 2014, Drebin, effective and explainable detection of Android malware in your pocket. The notebook reports a random-split binary ROC-AUC of 0.997291 on a third-party static parquet, and reports no precision, no F-measure and no result on any published partition.

SCIENTIFIC CONSEQUENCE

Drebin and the work that follows it report precision, recall and F-measure at operationally chosen thresholds, on published splits. A shuffled-split ROC-AUC on a re-derived feature parquet has no counterpart in that table. Your number cannot be shown to be better, worse or comparable, which for a benchmark contribution is the same as contributing nothing to the benchmark.

REQUIRED REPAIR

Report precision, recall and F-measure under the Drebin evaluation protocol, or on the CCCS-CIC release's own published partition, alongside the notebook's own figure.

DEFENSIBLE REPLY

The notebook never claims to enter the Drebin table. Section 2's related-work block is headed, in bold, that these are drawn from the wider literature and are not measurements reproduced on this exact corpus, and the Drebin row itself reads 'strong in-sample; drifts over time'. So the comparison you say was silently omitted is explicitly disclaimed and its direction stated. What survives is the cheap omission: precision 0.9874, recall 0.9772 and an F-measure of 0.982 were arithmetic away from the printed confusion matrix and none of the three was reported.

ORDERED CORRECTION

Recast as the cheap-omission charge, which is true and unanswerable, and drop the claim that comparability was asserted.

The transfer harness is in the same folder

EVIDENCE

Notebook thirty-six, cell twenty, fits one model per corpus and scores every model on every corpus, printing a four-by-four matrix with an off-diagonal mean of 0.5434 against a diagonal mean of 0.9966. Notebook thirty-five reports only in-distribution numbers and names no cross-corpus experiment anywhere.

SCIENTIFIC CONSEQUENCE

Notebook thirty-six establishes, on real data in this same codebase, that a detector scoring above 0.99 in distribution can fall to 0.3315 on a corpus it has not seen. That is the ceiling on what any in-distribution number in this series licenses, and it was measured by the author. The capability is present, the demonstration is present, and this notebook applies neither. There is no longer an excuse available.

REQUIRED REPAIR

Fit on CCCS-CIC-AndMal-2020 and score on a second Android corpus such as Drebin or AndroZoo under a shared static feature representation, in both directions, and report the off-diagonal.

DEFENSIBLE REPLY

Notebook 36's loop is twenty lines because its four parquets carry identical 43-column NetFlow schemas — I checked them. You ask me to fit on AndMal-2020 and score on Drebin 'under a shared static feature representation'. Drebin's features are sparse strings drawn from manifests and disassembly; mine are 9,504 anonymised columns named F0 to F9503. Building the mapping between them is not a missing cell, it is the whole contribution — and your own fault 17 charges me, correctly, with being unable to say what F50 means. Faults 14 and 17 cannot both stand: one says the mapping is trivially available, the other says I do not have it.

ORDERED CORRECTION

Re-mechanise to the weaker true claim: 'No cross-corpus evaluation of any kind was attempted, and notebook 36 proves in this same codebase that a detector above 0.99 in distribution can reach 0.3315 on a corpus it has not seen — which is the ceiling on what your 0.997291 licenses.' Drop 'a shared static feature representation', which fault 17 establishes does not exist here.

A letter grade applied to a maximum over nine thousand statistics

EVIDENCE

Cell twenty loops over every feature, records the direction-agnostic AUC of each, takes `best_auc` equals max of those values on a sixty-thousand-row sample with `random_state` one, and then grades it with hard cutoffs: F at 0.999, D at 0.99, C at 0.95, B at 0.85. The printed value is 0.8737 on feature F50, and the printed grade is B. The overall grade of F is then driven by the contamination half at 0.514.

SCIENTIFIC CONSEQUENCE

A maximum order statistic over nine thousand candidates has a different sampling distribution from a single AUC, and the notebook applies a cutoff designed for the latter to the former without correction. More broadly, the grade is a letter derived from two uninterval-ed point estimates against thresholds the notebook chose, and its only consequence is a sentence in section thirteen: it did not reach the abstract and it retracted no number.

REQUIRED REPAIR

Report the single-feature AUC distribution with its maximum's null reference under the number of candidates tested, attach intervals to both sub-checks, and drop the letter in favour of the quantities.

DEFENSIBLE REPLY

The observed 0.8737 sits far above anything a maximum over nine thousand null AUCs on sixty thousand rows would produce, and the independent eight-seed probe found the grade stable across a range of 0.8719 to 0.8773, so the correction would not have moved the letter in this instance.

Your weakest category is the one with no category

EVIDENCE

Cell eighteen prints worst per-family recalls with NoCategory at 0.806, the weakest entry. Cell eleven derives family directly from the raw label string, so NoCategory is whatever the upstream release left uncategorised. No positive count for that bucket is printed, and the count-under-five floor in cell eighteen prints nothing about what it skipped, though the independent probe confirms fourteen groups with none skipped.

SCIENTIFIC CONSEQUENCE

The conclusion nominates the rarer, harder categories, including the un-categorised bucket, as exactly the apps a store most needs to stop. But a bucket defined by missing metadata is not a threat class. It is a label-quality artefact, and a recall of 0.806 on it may reflect heterogeneity of the bucket rather than difficulty of the malware. The notebook's honest lead number rests on a class it cannot characterise.

REQUIRED REPAIR

Report the held-out positive count and the composition of the NoCategory bucket, and either resolve those apps to categories or exclude them from the per-category table with an explicit note.

DEFENSIBLE REPLY

The notebook names the un-categorised bucket in the abstract and gives it its own printed recall rather than folding it silently into the positive class, which is more transparent than dropping it would have been.

Nine thousand manifest flags describe a build, not an intent

EVIDENCE

Cell eleven's loader comment describes the features as permissions, intents and API-call flags. The strongest single feature, printed in cell twenty, is F50 at an alone-AUC of 0.8737, an anonymised column name that the notebook never maps to a semantic meaning. Cell eighteen's importance panel plots twelve similarly anonymous columns.

SCIENTIFIC CONSEQUENCE

The notebook cannot say what any of its top features means. F50 is a column index, not a permission. So even the multi-feature story is uninspectable: the reader cannot tell whether the model keys on a dangerous permission combination or on a signature of the packer, the advertising SDK or the build tool that a family happens to share. An adversary changes a manifest flag at zero cost and does not change behaviour at all.

REQUIRED REPAIR

Map the feature indices back to their named permissions, intents and API calls from the CCCS-CIC release, ablate the build-toolchain-coupled indicators, and report the score on behaviour-carrying features only.

DEFENSIBLE REPLY

The anonymised column names come from the upstream parquet rather than from any choice this notebook made, and permission and intent flags are the standard, well-validated Android static feature set going back to Drebin.

18

MAJOR

UNTESTED

VERIFIED

MULTI-FEATURE SEPARABILITY MISREAD AS REASSURANCE

The right reading, and no measurement behind it

EVIDENCE

Cell twenty-two prints the headline at 0.997291 and the shortcut-dropped variant at 0.997303. The conclusion says the separability is multi-feature, that reflects how this corpus was generated, not one leaky column. The independent probe agrees: dropping the top two features leaves the score at 0.9969.

SCIENTIFIC CONSEQUENCE

If dropping the two strongest of nine thousand four hundred and seventeen columns costs nothing, then the benign and malicious distributions barely intersect in this representation, and the size of that intersection is the number that decides whether the task is hard. The interpretation is asserted and the quantity behind it is never printed.

REQUIRED REPAIR

Estimate the class overlap directly, by nearest-opposite-class distance or a Bayes-error estimate, and report it as the quantity the multi-feature claim rests on.

DEFENSIBLE REPLY

The notebook is already reading the ablation in the harder direction rather than treating survival as vindication, which is the correct instinct and rarer than the criticism implies.

19

MAJOR

UNTESTED

VERIFIED

IMPURITY IMPORTANCE REPORTED AS EXPLANATION

Gini importance across nine thousand correlated flags

EVIDENCE

Cell eighteen reads `imp equals best dot feature_importances_` and plots the top twelve columns under the title `top importances slash absolute coefficient`. The forest in cell sixteen uses sixty trees over 9,417 columns, considering roughly ninety-seven candidate features per split by default.

SCIENTIFIC CONSEQUENCE

With sixty trees over nine thousand columns, most features are examined at few nodes, and impurity importance under those conditions is close to arbitrary among correlated indicators. The panel appears in the results section as an account of what the model uses, and it does not support that reading.

REQUIRED REPAIR

Permutation importance computed on held-out data, or SHAP values, with a stability check across seeds.

DEFENSIBLE REPLY

The notebook never claims the importance panel is causal or complete, it is one of four diagnostic panels, and the ablation rather than the importance plot is what carries the argument about shortcuts.

A winner named on five ten-thousandths

EVIDENCE

Cell sixteen's ranking: RandomForest 0.997291, XGBoost 0.996782, LightGBM 0.996325. The gap between first and second is 0.000509 on an 89,452-row held-out set. Cell sixteen prints best model, RandomForest, and cells eighteen, twenty, twenty-two and twenty-four all descend from that choice.

SCIENTIFIC CONSEQUENCE

Three learners within one thousandth are declared ranked, the top one is named best, and the entire results section is conditioned on that ordering with nothing establishing that a second draw reproduces it.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of RandomForest and XGBoost, or overlapping bootstrap intervals on the AUC difference, before any model is named best.

DEFENSIBLE REPLY

The comparison is paired on the same eighty-nine thousand held-out rows, where the standard error of an AUC difference is considerably smaller than the standard error of either AUC alone, so a gap of five ten-thousandths may well be a real ordering.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here RandomForest. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 89,452 held-out rows, and no validation object is ever created.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number. Every downstream number in this notebook inherits the same dependence.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 22 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Ten out of thirty. Reject. I want to be clear that this notebook earns a point on the reproducibility axis that most of the series does not, because it actually re-derived its headline under de-duplication and adopted the corrected figure. That is real work and I am counting it. But it corrected one scalar and left every diagnostic in the notebook computed from the model it had just disowned, and it did all of it on a random split of a corpus it told us has a hundred and ninety-one families it declined to use. Group by family, order by date, and regenerate the results section from the de-duplicated fit, and this becomes a defensible mobile-malware result.

Does It Transfer? Cross-Dataset Intrusion Detection on One Schema

Four intrusion corpora re-featured onto one NetFlow v2 schema — BoT-IoT, ToN-IoT, UNSW-NB15 and CSE-CIC-IDS2018 — four hundred thousand stratified flows from each, one point six million pooled, forty-one shared features.

HEADLINE

Two headlines. The pooled in-distribution figure: XGBoost, held-out ROC-AUC 0.999200 on 400,000 rows. And the one that matters: a four-by-four transfer matrix with an in-distribution diagonal mean of 0.9966 against a cross-corpus off-diagonal mean of 0.5434, minimum 0.3315, with five of twelve cells below chance.

DEFENSE

F

Selected-model estimand

DOCTORAL

14

/30 · REJECT

PRAXIS RUBRIC

9.5

/10 · self B

D1 4 External validity	D2 2 Statistical inference	D3 3 Construct validity	D4 0 Decision quality	D5 2 Reproducibility	D6 3 Honesty / scope

- 21 live faults
- 2 critical (FATAL)
- 19 major (CONTESTED)
- 2 withdrawn after refutation

WHAT THIS NOTEBOOK GETS RIGHT

I want to be unambiguous about this before anything else. Notebook thirty-six ran the right experiment. Every other notebook in this series ends by naming cross-distribution testing as the most important missing test, and this one built it: one model per corpus, each corpus split once before any fit so the diagonal is a genuine held-out score, then every model scored on every corpus. It printed a diagonal mean of 0.9966 against an off-diagonal mean of 0.5434 with five of twelve cells below chance and a worst cell of 0.3315, and it read that correctly, saying that below one half the learned rule is inverted rather than merely weak. Then it did the harder thing: it named what the matrix does not license, in its own words, no threshold, no false-positive rate, no per-corpus recall, and it explicitly refused its own in-distribution 0.999200 as a deployment estimate. That is the single most scientifically valuable result in thirty-six notebooks, and everything I say next is said because it is worth holding to a standard the others did not reach.

EXAMINER RULING

This is the only notebook in the series that ran the experiment the other thirty-five named and declined, and it deserves to be judged on that experiment rather than on theirs — so I am going to tell you that the experiment as built cannot distinguish a model that does not know your network from a model that does not know your attacks, because the column that separates those two was in every one of your four dataframes and you dropped it.

PART I

Lead with these five objections

Oral-defense priorities. Prefer FATAL items that survived refutation; include MAJOR items when they carry a known reply.

The Attack column was in every frame and you dropped it

EVIDENCE

Cell eleven reads: DROPCOLS equals Label, Attack, Dataset. The Attack field is part of the NF-v2 schema in every one of the four parquets. Cell twenty then fits one model per corpus and scores it on every corpus, and cell twenty's printed matrix gives an off-diagonal mean of 0.5434 and a worst cell of 0.3315 for train BoT-IoT, test ToN-IoT.

SCIENTIFIC CONSEQUENCE

Your section-one framing says deployment asks whether the model works on a different network, a different attack generator and a different year. The matrix varies all three at once and separates none of them. A cell at 0.3315 is consistent with two entirely different findings. It could mean the detector learned BoT-IoT's network fingerprint and cannot see through a different capture, which is your thesis. Or it could mean ToN-IoT contains attack classes BoT-IoT never held, in which case the model is failing at a task it was never trained for, which is not a transfer failure at all. Those two have opposite engineering consequences, and the column that tells them apart was sitting in the dataframe. You dropped it in the same line that dropped the label. This is the notebook whose entire contribution is that in-distribution scores are uninterpretable, and its own headline finding is uninterpretable for the same class of reason.

REQUIRED REPAIR

Retain the Attack column, restrict each off-diagonal comparison to attack types present in both the source and the target corpus, and report the matrix twice: once on shared attack classes, which measures network shift, and once on the full class set, which adds novel-class failure. The difference between those two matrices is the actual contribution.

A headline mean over twelve numbers spanning point three to point eight

EVIDENCE

Cell twenty prints cross-corpus off-diagonal mean 0.5434, minimum 0.3315, worse-than-random cells five of twelve. The twelve cells, readable in the printed matrix, are 0.3315, 0.6120, 0.7862, 0.4910, 0.3822, 0.5468, 0.7879, 0.3527, 0.3953, 0.5814, 0.6293 and 0.6243. Cell twenty-seven prints seed equals zero, and cell ten's comment reads single seed used everywhere. No bootstrap, no interval and no repeated seed appears anywhere.

SCIENTIFIC CONSEQUENCE

Twelve numbers with a spread of nearly half a unit have been collapsed into one figure that the conclusion calls the honest number, and the reader is given no way to know what that mean is a mean of. Twelve cells from four corpora is a sample of size four in the thing that actually varies, and the standard error of a mean over four correlated captures is not small. The count of five below one half, which the conclusion leans on hardest, is a count of point estimates against a hard cutoff with no interval attached: cell 0.4910 sits nine thousandths below one half, and a second seed could move it across. The notebook has produced the most important result in the series and has not established that any digit of it is stable.

REQUIRED REPAIR

Bootstrap each cell over held-out resamples and report an interval; repeat the entire matrix across at least ten seeds and report cell-wise mean and standard deviation; and report the count of below-chance cells with its uncertainty rather than as an integer.

Your ROC baseline is not one half, it is 0.94

EVIDENCE

Cell eleven prints four hundred thousand rows from each corpus at attack rates 0.9957, 0.7258, 0.0378 and 0.1184. Cell sixteen prints the majority-class baseline accuracy of 0.5306 and the note that for ROC-AUC the trivial baseline is 0.5, and the conclusion repeats it. Now do the arithmetic on the printed rates. Four hundred thousand times 0.9957 is three hundred ninety-eight thousand two hundred and eighty BoT-IoT attacks. The four corpora together supply seven hundred fifty-one thousand and eighty attacks, so BoT-IoT alone is fifty-three percent of every positive in the pooled corpus. UNSW-NB15 supplies three hundred eighty-four thousand eight hundred and eighty benign flows, forty-five percent of every negative. Score each flow by nothing but its own corpus's attack rate and the resulting area under the curve is zero point nine three seven, and you can check that against the printed pooled positive rate, which comes out at 0.4694 exactly as the probe reports.

SCIENTIFIC CONSEQUENCE

The pooled task is very close to the task of identifying which capture a flow came from, and the forty-one NetFlow features almost certainly permit that: four captures from four laboratories have different throughput scales, different port distributions and different duration profiles. So 0.999200 is not being compared against 0.5. It is being compared against roughly 0.94, and the honest margin is six hundredths, not half. The notebook is scrupulous about not mixing the accuracy axis and the ROC axis, and then quotes the wrong ROC reference on both. Every reader will take the wrong margin away.

REQUIRED REPAIR

Report the corpus-prior scorer's area under the curve as an explicit row in the ranking table, exactly as the majority-class row is already carried there, and state the pooled headline's margin over it. Better, drop the pooled headline entirely: it is a warm-up whose only defensible use is to show that the transfer collapse is not caused by a leaky column.

DEFENSIBLE REPLY

Your arithmetic is right to four places and I am not going to argue with it: 751,080 positives, 53% of them BoT-IoT at 99.57%; 848,920 negatives, 45% of them UNSW-NB15 at 3.78%; pooled positive rate 0.4694; corpus-prior AUC 0.93677. I will go further than you did — I computed the pair decomposition, and 91.4% of the pooled comparison is cross-corpus pairs. The confound is real and worse than you said. But two things break the charge as you have framed it. First, 0.937 is not a baseline. A baseline is what a competitor could achieve; that scorer reads the Dataset column, which cell 11 drops precisely so no model can use it. The notebook's sentence 'for ROC-AUC the trivial baseline is 0.5' is the textbook-correct statement for a no-information classifier and is not wrong. Second, your load-bearing hedge is that the 41 NetFlow features 'almost certainly permit' corpus identification, and the notebook's own printed output refutes it. Cell 18 reports per-corpus recall of 0.969 on UNSW-NB15 at an overall false-positive rate of 0.0065. A corpus-prior scorer assigns every UNSW flow 0.0378, predicts benign, and returns a UNSW recall of zero. The pooled model discriminates inside the lowest-prevalence corpus while holding false

positives under one percent, which the corpus-prior scorer cannot do at all. And 0.9992 against 0.9368 is not 'six hundredths' — it is a seventy-nine-fold reduction in discordant pairs.

ORDERED CORRECTION

Re-mechanise from 'your baseline is wrong' to 'your pooled number is confounded': 'Ninety-one percent of the pairs your pooled ROC-AUC ranks are cross-corpus pairs. It is overwhelmingly a statement about which capture a flow came from, and you never report the pooled model's AUC within each corpus, which is the un-confounded quantity and costs four lines.' That charge survives the printed recalls; the baseline framing does not.

The BoT-IoT model saw five hundred benign flows

EVIDENCE

Cell twenty fits each model on 120,000 rows of its own corpus. Cell eleven gives the attack rates. So the BoT-IoT model was fitted on 120,000 times one minus 0.9957, which is about five hundred and sixteen benign flows. The UNSW-NB15 model was fitted on 120,000 times 0.0378, about four thousand five hundred and thirty-six attack flows. And BoT-IoT's diagonal cell of 0.9999 is scored on that corpus's held-out hundred thousand rows, of which about four hundred and thirty are benign. Nowhere in the notebook does `class_weight` or `scale_pos_weight` appear.

SCIENTIFIC CONSEQUENCE

Section three states that ROC-AUC is insensitive to the base rate, so the numbers isolate distribution shift rather than prevalence. That is true of the evaluation side and false of the fitting side, and the notebook only addresses the first. A classifier fitted on five hundred negatives has no meaningful estimate of the benign region at all. The row of the matrix that produces your worst cell, 0.3315, is the row belonging to the model that barely saw a benign flow. So the flagship number of this series does not isolate what it says it isolates: it mixes the shift you meant to measure with a training-set imbalance that varies by two orders of magnitude down the rows.

REQUIRED REPAIR

Refit each source model at a common, stated class balance, or with `scale_pos_weight` set so the effective prevalence is equalised across the four, and report the matrix again. The difference between the balanced and unbalanced matrices is how much of the collapse was prevalence and how much was shift.

DEFENSIBLE REPLY

The counts are right and the omission is real: no `class_weight`, no `scale_pos_weight`, the BoT-IoT model fitted on roughly 516 benign flows and the UNSW model on roughly 4,536 attacks. But you assert that this is what produces the collapse, and you support it by pointing at one cell. Take row means instead, which is the aggregate the claim requires. BoT-IoT, the model that barely saw a benign flow, averages 0.5766 across its three off-diagonal cells — the second best row in the matrix, above the overall off-diagonal mean of 0.5434, and it also produces 0.7862, the second-highest cell anywhere. CSE-CIC-IDS2018 at 11.84% attack averages 0.6117, UNSW-NB15 at 3.78% averages 0.5120, and the worst row is ToN-IoT at 0.4733 with a middling 72.58% attack rate. There is no monotone relationship between source prevalence and transfer in your own data. The row that produced your worst cell also produced your second-best.

ORDERED CORRECTION

Drop 'The row of the matrix that produces your worst cell is the row belonging to the model that barely saw a benign flow' — it is true of one cell and false of that row's average. Keep the defensible form: four models fitted at prevalences spanning two orders of magnitude with no weighting are not four comparable instruments, and the balanced-versus-unbalanced matrix difference is unmeasured. That charge does not require the causal claim your data contradicts.

The result that matters has no operating point at all

EVIDENCE

The section-thirteen conclusion says, in its own words: ROC-AUC is the only metric reported here, and the matrix reports no threshold, no false-positive rate and no per-corpus recall. Cell twenty computes `roc_auc_score` and nothing else. The only operating-point figure in the notebook, cell eighteen's false-positive rate of 0.0065 with one thousand three hundred seventy-eight benign flagged of two hundred twelve thousand two hundred twenty-three, belongs to the pooled warm-up model, not to any transfer model.

SCIENTIFIC CONSEQUENCE

The finding this notebook exists to deliver is that a detector does not travel. A defender's response to that finding depends entirely on quantities the matrix does not contain: at what threshold does the transferred model still catch anything, and how many false alarms does that cost. An area under a curve of 0.5434 is compatible with a model that is useless at every threshold and with a model that is usable at one, and the notebook cannot say which. Worse, a model fitted at ninety-nine point five seven percent attack outputs probabilities that are grossly miscalibrated on a corpus at three point seven eight percent, so the operating point does not merely need reporting, it needs re-deriving on every target. The notebook names all of this correctly and stops.

REQUIRED REPAIR

For each off-diagonal cell report precision at a fixed recall, the false-positive rate at a threshold chosen on the source corpus, and the same after recalibration on a small labelled sample of the target. That last quantity is what a defender would actually buy.

DEFENSIBLE REPLY

The finding is that a rule learned on one capture is inverted on another. That is a statement about ordering, and a rank statistic is the right and complete instrument for it — five cells below 0.5 means the model is worse than random at every threshold in the ranking sense, which no operating point would add to. Reporting precision at fixed recall across four corpora whose attack rates run from 3.78% to 99.57% would reintroduce the base-rate confound that the metric choice exists to remove, and section 3 says so before any number appears. The conclusion then states the boundary of its own instrument in its own words — no threshold, no false-positive rate, no per-corpus recall. What I concede, and it is your strongest point, is the middle of the range: an area of 0.5434 is genuinely compatible with a model that is useless everywhere and with one usable in a narrow high-precision region, and the matrix cannot say which.

ORDERED CORRECTION

Narrow the charge to where it bites and it becomes much harder to answer: 'For the seven cells above 0.5 your matrix cannot say whether any usable operating point exists. Precision at fixed recall on those seven, in the target's own prevalence, is the number a defender would buy and it is four lines away.' Also drop the recalibration requirement from the required clause — it duplicates fault 17 and, as fault 17's own rebuttal concedes, it cannot move a rank statistic.

PART II

Complete live fault record

Remaining non-withdrawn faults. Each was written and, unless marked unreviewed, attacked by an independent refuter.

You proved disclosure is not a fix, then used it as yours

EVIDENCE

The section-thirteen conclusion's scope paragraph reads: the split is random, not temporal or entity-grouped. Every number in this paragraph therefore measures in-distribution separability only. That applies to the pooled headline and, per cell twenty, to each per-corpus split that produces the diagonal. The same conclusion says of the other notebooks: this is the experiment the other 35 notebooks name and do not run.

SCIENTIFIC CONSEQUENCE

You are entitled to that sentence about the other thirty-five precisely because you ran the experiment. Which is why the diagonal is the problem. Half of your headline comparison, the 0.9966 mean that the off-diagonal is measured against, is produced under exactly the protocol your notebook exists to discredit, and you discharged that with the same disclosure paragraph the other thirty-five used. If a disclosed random split is not a defence for them, it is not a defence for your diagonal, and the size of the diagonal-to-off-diagonal gap, which is the number the room will remember, depends on it.

REQUIRED REPAIR

Produce the diagonal under an entity-grouped or time-ordered split within each corpus, so that both halves of the comparison are defensible. Report how much of the 0.4532 gap between 0.9966 and 0.5434 survives once the diagonal is honest.

DEFENSIBLE REPLY

The move is elegant and I take the hit on the diagonal — it is a random-split number and I did discharge it with the same paragraph I criticise the others for using. But look at what it touches. Every off-diagonal cell is scored on PREP[tgt], the entire 400,000-row target corpus, by a model fitted only on a different corpus. No split protocol inside the source can inflate a score on a corpus the model has never seen. So 0.5434, 0.3315 and five-of-twelve — the numbers the room will remember — are untouched by this charge. What it touches is the 0.4532 gap, and it touches it in the direction that costs me: an honest diagonal would be lower, so my headline contrast is overstated and my criticism of the other thirty-five is what shrinks. And the remedy is not available: I read the dhoogla parquets and they carry 41 features plus Label and Attack, with no source address, no session id and no timestamp. There is no entity to group on and no time to order by.

ORDERED CORRECTION

State the scope honestly and the charge survives at full force within it: 'Your diagonal is an entity-memorisation score, so the gap you print is not the gap you measured. The off-diagonal is untouched — say so, because it is the only half of your comparison that is defensible, and then stop quoting a 0.4532 gap you cannot support.' Also drop the entity-grouped remedy, or re-mechanise it as mirror selection: dhoogla's cleaned NF-v2 strips the two IPv4 address columns Sarhan's release carries, so you chose a mirror on which the grouped diagonal is impossible.

You named a different year and never tested one

EVIDENCE

Section one, cell two, reads: it asks whether the model still works on a different network, a different attack generator and a different year. Cell eleven draws a stratified random sample of four hundred thousand rows from each corpus, which destroys time ordering before any split. Cell twenty splits each corpus at random. The conclusion confirms: the split is random, not temporal or entity-grouped.

SCIENTIFIC CONSEQUENCE

You stated the three-part definition of the question and then built an instrument that varies two of the parts together and the third not at all. Concept drift within a single network over months is the shift a defender meets first and meets continuously, long before they meet a different corpus. Nothing in this notebook measures it, and the stratified subsample in cell eleven removes the ordering that would allow it even in principle.

REQUIRED REPAIR

Order each corpus by flow start time, train on an earlier window, test on a later window within the same corpus, and report that within-corpus temporal drop as a third row of the ledger alongside the diagonal and the off-diagonal.

DEFENSIBLE REPLY

Your fault 1 says, in your words, that the matrix 'varies all three at once and separates none of them'. This fault says the third, time, 'is never varied at all'. Those cannot both be true, and fault 1 has it right: UNSW-NB15 was captured in 2015, BoT-IoT and CSE-CIC-IDS2018 in 2018, ToN-IoT in 2019-20, so every off-diagonal cell already crosses years — that is precisely why fault 1's confounding charge lands. The within-corpus drift experiment you want instead is a different and legitimate ask, and it is one I cannot run: the dhoogla parquets carry 43 columns, 41 features plus Label and Attack, and there is no flow start time among them. FLOW_DURATION_MILLISECONDS is a duration, not an origin. Your evidence blames the stratified sample in cell 11 for destroying an ordering that was never in the file.

ORDERED CORRECTION

Either withdraw as contradicting fault 1, or re-mechanise to corpus selection: 'Concept drift within one network is the shift a defender meets first, and the mirror you chose carries no timestamp, so your instrument cannot vary the axis your section 1 names third. Say that in section 3 instead of listing time among the things the matrix measures.'

Two prevalence regimes, one prevalence-blind metric

EVIDENCE

Cell eighteen imports `precision_recall_curve`, calls it on the pooled winner's probabilities, and plots it under the title Precision-Recall. The function `average_precision_score` appears nowhere in the notebook. Cell twenty reports only `roc_auc_score`. The four corpora carry attack rates from 0.0378 to 0.9957.

SCIENTIFIC CONSEQUENCE

The notebook chose ROC-AUC deliberately, and gives a good reason: the four corpora differ sharply in prevalence and the aim is to isolate shift from base rate. That reasoning is sound for the matrix and it is exactly why the matrix cannot answer a defender's question, because a defender operates at a prevalence and needs precision. And on the pooled side there is no such excuse: you computed the precision-recall curve, drew it into the figure, and did not take its area or read a point off it. Plotting the right diagnostic and declining to read it is the recurring failure of this whole series and it is present here too.

REQUIRED REPAIR

Report average precision for the pooled model with an interval, and report precision at fixed recall for every off-diagonal cell alongside the area under the receiver operating curve, so that both the prevalence-free and the prevalence-dependent views are on the page.

DEFENSIBLE REPLY

You grant, correctly, that ROC-AUC is the right choice for the matrix and give my reason back to me. So the charge reduces to the pooled side, and there the pooled positive rate is 0.4694 — near balance, where PR-AUC and ROC-AUC carry almost the same information and the textbook imbalance argument has no purchase. The pooled model is labelled a warm-up in section 9's header, in the section 13 ledger heading, and refused as a deployment estimate in the conclusion; its only defensible use is showing the collapse is not a leaky column, which is exactly what sections 10 and 11 use it for — single-feature AUC 0.8643, dropping the top feature costs nothing, contamination 0.000. I concede the mechanical point: `precision_recall_curve` is called, plotted, and its area never taken.

ORDERED CORRECTION

Keep only the mechanical omission and drop the two-prevalence-regimes framing, which your own text concedes away in the same paragraph.

The size of the collapse is a function of an untuned depth

EVIDENCE

Cell twenty instantiates XGBClassifier with `n_estimators` equal to one hundred twenty and `max_depth` equal to six for all four source corpora, with no grid, no random search and no per-corpus tuning. The pooled comparison in cell sixteen likewise hand-sets sixty trees for the forest and eighty estimators at depth six for the boosters.

SCIENTIFIC CONSEQUENCE

A depth-six ensemble of a hundred and twenty trees on a hundred and twenty thousand rows is a high-capacity fit, and high capacity is how a model memorises the idiosyncrasies of a single capture. So the off-diagonal mean of 0.5434 is not a property of the corpora, it is a property of the corpora at one arbitrary regularisation setting that nobody searched. A more heavily regularised model would very likely transfer better and score lower on the diagonal, which would change both halves of your headline gap. The central quantity of this notebook is a function of a hyperparameter that was never varied.

REQUIRED REPAIR

Sweep model capacity, at minimum depth and estimator count, and plot the diagonal and off-diagonal means as functions of it. The transfer story is the shape of those two curves, not one point on them.

DEFENSIBLE REPLY

One untuned configuration across four corpora is a real omission and I will sweep it. But you assert that a more heavily regularised model 'would very likely transfer better', and that is speculation with a ceiling you can compute. Shrinking capacity pushes every off-diagonal cell toward 0.5. My off-diagonal mean is 0.5434 — already only 0.0434 above it. In the limit, regularisation raises 0.3315, 0.3527, 0.3822, 0.3953 and 0.4910 toward 0.5 and drags 0.7862 and 0.7879 down toward it, and the mean moves from 0.5434 to 0.5000. That is worse, not better. Five cells below chance is not a capacity finding at all: an inverted rule means the class-conditional relationship reverses on the target, which regularisation does not repair. Holding one configuration fixed across all four rows is also what makes the rows comparable; tuning each separately would put a new confound down the rows of a matrix whose whole point is that only the capture changes.

ORDERED CORRECTION

Strike 'would very likely transfer better'. Ask for the sweep as a sensitivity analysis without predicting its direction: 'Your central quantity is a function of a regularisation setting nobody varied, and you cannot say whether the curve of off-diagonal mean against capacity is flat, rising or falling.' That is true and needs no prediction.

One draw for the most important result in the series

EVIDENCE

Cell ten sets `RANDOM_STATE` to zero with the comment single seed used everywhere. Cell eleven samples each corpus with `random_state` zero. Cell twenty passes `RANDOM_STATE` to both splits and to all four classifiers. Cell twenty-seven prints seed equals zero.

SCIENTIFIC CONSEQUENCE

The stratified four-hundred-thousand-row sample of each corpus is itself a random draw, and it is a small slice of corpora that are far larger. So there are two nested single-seed choices between the raw data and the printed matrix, and neither was varied. The conclusion nevertheless quotes the off-diagonal mean to four decimal places and the below-chance count to the unit. Neither resolution is supported by one draw.

REQUIRED REPAIR

Repeat the corpus subsampling and the entire matrix over at least ten seeds and report the cell-wise mean and standard deviation and the distribution of the below-chance count.

DEFENSIBLE REPLY

Strip out what fault 4 already says and one observation remains: the 400,000-row slice of each corpus is itself a single draw, and BoT-IoT's parquet holds 30,420,086 rows, so I sampled 1.3% of it. That is worth stating. It is also stratified on Label and 400,000 rows is a large absolute sample, so the composition of common attack types is preserved to within a fraction of a percent and the sampling contribution to any cell is small next to the corpus-level variation fault 4 identifies. The remedy you ask for here — repeat the matrix over ten seeds — is word for word the remedy in fault 4.

ORDERED CORRECTION

Merge into fault 4, keeping only the nested-draw observation as an additional line of its required clause.

The one notebook that measures a ceiling never measures an adversary

EVIDENCE

A search of the full source of this notebook for the words adversarial, evasion, evade and perturb returns nothing outside a code comment about a function being robust to XGBoost's two-column probability output. The reference list in cell twenty-nine names Sarhan, Sommer and Paxson, Arp and colleagues, and Apruzzese and colleagues. Arp's Dos and Don'ts is cited for sampling and evaluation pitfalls and applied only to that.

SCIENTIFIC CONSEQUENCE

Distribution shift and adversarial shift are the two ways a deployed detector fails, and this notebook has measured one of them beautifully and not looked at the other. The result makes the omission sharper rather than softer: having demonstrated that a detector loses everything against an unintended change of capture, the obvious next question is what it loses against an intended one. An attacker who alters flow duration and byte counts within a stated budget is a cheaper and more likely adversary than a change of corpus.

REQUIRED REPAIR

Evasion under a stated perturbation budget on the NetFlow features, applied to held-out attack flows, with worst-case recall reported for the diagonal and for every off-diagonal cell.

DEFENSIBLE REPLY

There is no evasion experiment anywhere and I will not pretend otherwise. But notice what this notebook claims: that a detector scoring 0.9966 on its own capture averages 0.5434 on one it has not seen. That is a negative result, and a negative result does not have a worst case to report — there is no proposed detector here whose robustness a defender would plan against. Your own framing gives it away: 'having demonstrated that a detector loses everything against an unintended change of capture, the obvious next question is what it loses against an intended one.' The next question is the next paper. Eight of the twelve fatal charges on this sheet are the same charges pressed onto all thirty-six notebooks, and this is the clearest of them.

ORDERED CORRECTION

Keep it as a scope note rather than a fatal finding, or make it specific: an adversary who perturbs flow duration and byte counts within a budget would let you say whether adversarial shift is larger or smaller than the corpus shift you measured — which would be a genuine extension of this notebook's contribution rather than a demand for a different one.

13

MAJOR

UNTESTED

VERIFIED

THE TRANSFER EXPERIMENT ARRIVES LAST

You measured the ceiling and left the headlines standing

EVIDENCE

The section-thirteen conclusion reads: this is the experiment the other 35 notebooks name and do not run, their scores should be read in its light. This is notebook thirty-six of thirty-six. The other notebooks' abstracts and conclusions carry their headline figures unamended.

SCIENTIFIC CONSEQUENCE

The capability demonstrated here, one shared schema and a twenty-line double loop, was available in this codebase from the first notebook onward. Had it been notebook two, the following thirty-four would have been designed around it. Placed at thirty-six it functions as an epilogue rather than a constraint: every headline it invalidates has already been published, and the notebook that invalidates them does not restate a single one. The reading it asks for is left to the reader, which is the same move the other thirty-five made with their own limitations.

REQUIRED REPAIR

Publish a revision table that restates each affected notebook's headline with the transfer ceiling attached, or move this experiment to the front of the series so the remaining notebooks are built under it.

DEFENSIBLE REPLY

The series is pedagogically ordered and a capstone that synthesises and then falsifies what came before is a legitimate and common teaching structure, and no notebook has the authority to edit its siblings.

14

MAJOR

MEASURED

VERIFIED

MODEL RANKING WITHOUT A TEST

A pooled winner named on thirty-eight millionths

EVIDENCE

Cell sixteen's ranking: XGBoost 0.999200, LightGBM 0.999162, RandomForest 0.998457, LogisticRegression 0.945969, MajorityBaseline 0.500000. The gap between first and second is 0.000038. Cell sixteen prints best model, XGBoost, and cells eighteen, twenty-three and twenty-five descend from that choice.

SCIENTIFIC CONSEQUENCE

Thirty-eight millionths separate the top two, and no test establishes the ordering. Every pooled diagnostic in the notebook is conditioned on a choice that has not been shown to reproduce.

REQUIRED REPAIR

McNemar's test on the paired held-out predictions of the top two, or overlapping bootstrap intervals on the difference, before a winner is named.

DEFENSIBLE REPLY

The pooled model is explicitly a warm-up in this notebook rather than the contribution, the conclusion refuses its number as a deployment estimate, and the transfer matrix uses a fixed XGBoost regardless of that ranking, so the selection does not propagate into the finding.

Four hundred thousand from each, one hundred twenty thousand fitted

EVIDENCE

Cell eleven sets PER_SET equal to four hundred thousand and stratifies on Label. Cell twenty sets TRAIN_N equal to a hundred and twenty thousand, and prints fitted on each corpus with a hundred and twenty thousand rows. Cell sixteen prints materialized 1,600,000 rows, trained on 120,000, held out 400,000.

SCIENTIFIC CONSEQUENCE

The off-diagonal collapse is being attributed to distribution shift, but a model fitted on a hundred and twenty thousand rows of a corpus has not necessarily saturated on that corpus, and an unsaturated model transfers worse for reasons that have nothing to do with shift. No learning curve appears anywhere, so the reader cannot tell how much of the 0.4532 gap is shift and how much is sample size.

REQUIRED REPAIR

Plot the diagonal and off-diagonal means as functions of training size, from ten thousand rows to the full slice, and show that both have saturated before attributing the gap to shift.

DEFENSIBLE REPLY

Section three states plainly that the four hundred thousand is a stratified sample rather than the full corpus, cell twenty carries the comment that the budget is the same one the other notebooks use so that the comparison is like-for-like, and the diagonal cells at 0.988 and above are difficult to reconcile with a model that has not saturated.

A B that sits one hundredth from its cutoff

EVIDENCE

Cell twenty-three prints best single-feature AUC 0.8643 on DST_TO_SRC_AVG_THROUGHPUT, exact-duplicate row rate 0.000, train-test contamination 0.000, and data trust grade B, driven by the single-feature check with the contamination check scoring A. The grade cutoff for B is 0.85. The independent eight-seed probe ranged from 0.8608 to 0.8643, so the closest seed clears the boundary by 0.0108, which passes the proximity rule by eight ten-thousandths.

SCIENTIFIC CONSEQUENCE

The grade is quoted in the conclusion as a finding and it is a letter applied to an interval-free point estimate that a different draw moves by three and a half thousandths, against a boundary about one hundredth away. That is not a measurement, it is a rounding of one.

REQUIRED REPAIR

Report the single-feature AUC with a confidence interval and the contamination with one, and drop the letter.

DEFENSIBLE REPLY

Do the arithmetic the correction requires. Eight seeds span 0.8608 to 0.8643 — a width of 0.0035. The closest seed sits 0.0108 above the 0.85 cutoff and the distance to the next cutoff up is at least 0.0857. The spread is three times narrower than the distance to the boundary and eight of eight seeds return B, on the same winning feature every time. And the other sub-check is not near anything at all: exact-duplicate rate 0.000 and contamination 0.000, grade A, confirmed in two independent windows. Citing that spread as instability proves the opposite of instability. The durable version of your point is about method, not about this letter.

ORDERED CORRECTION

Rewrite to the method complaint and delete the proximity claim: 'A letter derived from hard cutoffs on two interval-free point estimates is not a measurement, whatever its stability, and your conclusion quotes the letter as a finding.' Do not say the sweep nearly crosses the boundary — it clears it by three times its own width.

Four laboratory captures, described as four networks

EVIDENCE

Cell eleven's loader comment reads: Sarhan et al. re-featured four different intrusion datasets into the same 43-field NetFlow v2 layout, that is what makes train-on-one, test-on-another possible, the columns line up so the only thing that changes between train and test is the capture. Cell twenty-nine cites Sarhan, Layeghy and Portmann, 2022, as the source of both the feature set and all four corpora.

SCIENTIFIC CONSEQUENCE

Holding the feature extractor constant is what makes the comparison clean, and it also removes one of the largest sources of shift a real deployment meets, because two organisations do not run the same flow exporter with the same configuration. So 0.5434 is an optimistic estimate of transfer failure, not a pessimistic one. And a testbed capture is not a network: none of these four carries the benign traffic of an operating enterprise. The conclusion's phrase, a network it has never seen, promises more than four laboratory captures can deliver.

REQUIRED REPAIR

Add a fifth corpus captured with a different exporter, or re-extract one corpus with a different flow tool, and report how much further the off-diagonal falls when the feature pipeline is allowed to vary.

DEFENSIBLE REPLY

Holding the schema constant is the deliberate and correct design choice, because it is the only way to attribute the collapse to the capture rather than to feature-engineering differences, and the notebook says so explicitly in section one and again in the loader comment.

One corpus was silently made the schema authority

EVIDENCE

Cell eleven reads: feat equals the list comprehension over frames of UNSW-NB15 dot columns, excluding DROPCOLS. That list is then applied to every corpus through the `_prep` function. No assertion, no set comparison and no printed column count per corpus appears anywhere. The pooled path separately applies a constant-column filter, `X dot loc`, colon, `X dot nunique greater than one`, that the transfer path's PREP does not apply.

SCIENTIFIC CONSEQUENCE

UNSW-NB15 was made the schema authority by position in a dictionary literal, not by argument. The entire validity of a transfer matrix rests on the columns meaning the same thing in the same order across four files, and that is exactly the invariant the notebook never checks or prints. And the two halves of the notebook prepare their features by different code paths, so nothing guarantees that the audit in cell twenty-three and the matrix in cell twenty ran on the same list.

REQUIRED REPAIR

Assert set equality and ordering across the four column indexes, print the per-corpus column count, and route both the pooled audit and the transfer matrix through one preparation function.

DEFENSIBLE REPLY

The NF-v2 releases are by construction one schema, a mismatch would raise a `KeyError` at the first `_prep` call rather than pass silently, and the printed forty-one shared features is consistent with the forty-three-field schema minus the dropped label columns.

Gini importance, and no attribution for the matrix at all

EVIDENCE

Cell eighteen reads `imp equals best dot feature_importances_` and plots the top twelve for the pooled model. Cell twenty computes no importance for any of the four per-corpus models. Neither permutation importance nor SHAP appears anywhere.

SCIENTIFIC CONSEQUENCE

The finding is that a rule learned on one capture inverts on another, and the immediate scientific question is which columns invert. Four fitted models sit in the models dictionary in cell twenty, and comparing their attributions would answer it directly at almost no cost. Instead the only attribution in the notebook belongs to the model the conclusion disowns, and it is impurity importance, which is biased toward high-cardinality columns.

REQUIRED REPAIR

Compute permutation importance on held-out data for each of the four source models, place the four rankings side by side, and report which features carry opposite sign or opposite rank across corpora.

DEFENSIBLE REPLY

The importance panel is inherited from the shared series template rather than authored for this notebook, it is presented as one diagnostic among four rather than as an explanation, and the ablation rather than the importance plot carries the shortcut argument.

The set that picked the winner is the set that scored it

EVIDENCE

Cell 16 computes `roc_auc_score` on the held-out set for each of the four learners, sorts the table by that column, and assigns `best_name` from the top row — here XGBoost. Cell 18 then reports the confusion matrix, the ROC and precision-recall curves, the operational false-positive rate and every per-family recall on that same held-out set. Four calls to `train_test_split` appear in the notebook, but only one produces the held-out partition that every reported figure is computed on, producing 400,000 held-out rows, and no validation object is ever created. This applies to the pooled-split section only. `nb36`'s headline result — the cell-20 transfer matrix — fits a separate model per source corpus on separate per-corpus splits and never references the selected model, so it does not inherit this dependence.

SCIENTIFIC CONSEQUENCE

A figure computed on the data that selected the model is not an estimate of performance on unseen data. The optimism here is small and can be measured: selecting among these four learners on one random half of the held-out set and scoring on the disjoint half gives a mean optimism of $-4.1\text{e-}08$ over thirty replicates, worst single replicate $6.3\text{e-}06$, with the same winner in twenty-nine of thirty — roughly two orders of magnitude below the sixth decimal the headline is printed to, and smaller still at the notebook's full held-out size. The defect is therefore definitional rather than numerical: the reported figure is not an independent estimate, and the notebook nowhere says so. That is worth fixing and worth stating; it is not grounds for disbelieving the number.

REQUIRED REPAIR

Split three ways: select on a validation partition, then report the chosen model once on a test partition touched by nothing else. Every corpus in this series holds at least 357,805 rows, so a third partition is affordable in all thirty-six and corpus size is no defence.

DEFENSIBLE REPLY

The optimism from choosing the best of four pre-specified learners on a test set this large is small — plausibly the same order as the ablation differences the notebook already discusses. The defect is that the estimate is not independent, not that it is badly wrong.

Every warning is silenced before any model is fitted

EVIDENCE

Cell 10 calls `warnings.filterwarnings('ignore')` with no category and no module restriction. The filter would suppress convergence failures from `LogisticRegression`, dtype coercions in pandas and deprecation notices from scikit-learn alike. Re-running nb01's whole pipeline with warnings enabled on the recorded scikit-learn version emits none of them — its `LogisticRegression` converges at `n_iter_ = 70` of `max_iter = 300` — so on that notebook the filter is demonstrably hiding nothing. On the other thirty-five the question is open, and that is the objection.

SCIENTIFIC CONSEQUENCE

A logistic regression that never converged still returns coefficients and still produces an ROC-AUC. With warnings off, that result is indistinguishable on the page from one that converged. The absence of a warning in this notebook is not evidence that nothing went wrong, because nothing could have been printed.

REQUIRED REPAIR

Restrict the filter to named categories, or remove it and fix what it was hiding. At minimum re-run once with warnings enabled and report what appears — particularly any convergence failure on the linear model, which would change how its comparison against the tree ensembles should be read.

DEFENSIBLE REPLY

The filter is boilerplate that keeps the teaching output readable, and no specific warning has been shown to have fired. That is fair — but it is also the point: with the filter in place neither of us can tell, and the burden sits with the author.

The de-duplication ablation is scored on a different test set

EVIDENCE

Cell 25 defines `_retrain_auc`, which calls `train_test_split` on whatever matrix it is handed, re-subsamples, clones the winning estimator and refits before scoring. For the de-duplicated corpus that is a different row set and therefore a different held-out partition from the one the headline uses. No repetition anywhere estimates how much the score moves between splits.

SCIENTIFIC CONSEQUENCE

The quantity read as the de-duplication effect is that effect plus an unmeasured difference between two random splits. A ten-seed sweep of the notebook's own `_retrain_auc` on the unablated corpus gives a standard deviation of $3.45e-06$ and a range of $1.18e-05$, against a reported de-duplication gap of $7.07e-06$ — the nuisance term is the size of the effect. The clearest demonstration in the series is nb15, where the printed duplicate rate is 0.000 and the row is labelled '0% rows removed', yet the score moves from 0.994731 to 0.989154: five parts per thousand of pure split-and-refit noise, presented as an ablation result.

REQUIRED REPAIR

Repeat the de-duplicated comparison over at least ten seeds and report the paired difference with an interval. Do not attempt to score a de-duplicated model on the original held-out rows — that would put back the duplicates it removed. The shortcut-drop row already holds the split fixed and needs no change.

DEFENSIBLE REPLY

The other arm of this ablation is already paired. Dropping the shortcut feature leaves the row count, the stratify vector and the seed unchanged, so `train_test_split` returns the identical train and test indices and that comparison is exact. Only the de-duplication arm is affected, and the notebook reports both.

Closing. Fourteen out of thirty. Reject, at the very top of the band, one point below major revision, and I want the room to hold both of those facts at once. This notebook ran the experiment that thirty-five others named and declined, and it found that a detector scoring 0.9966 on its own capture averages 0.5434 on a capture it has not seen, with five of twelve cells below chance and a worst cell of 0.3315 where the learned rule is actively inverted. That is the most valuable number produced in this entire series and it should be the first slide of every one of your defences. And it is one point short of a revision band because the instrument that produced it cannot tell a network it does not know from an attack it does not know, has no interval on any of its twelve cells, and reports no threshold at any of them. Restore the Attack column, equalise the source prevalence, bootstrap the cells, and this is publishable.

What this edition is bound to

Examiner fault records: `assignments/review-data/faults-final/` after adversarial verdicts. Defense ledger: `docs/course/SEAS8414-notebook-doctoral-defense-ledger.json` with cell-level proof locators. Praxis rubric totals and self-grades: packaged findings pack. This combined file is a derived publication; the two source reports remain authoritative for their exclusive fields.

Findings pack SHA-256: 49bdad41d72a258c99dc39ef68af688d22819717e999861537053a7d27a9f7b1

Defense ledger SHA-256: b3cd1ae3c3b1543e9716ea7d40473df4ede7d02a4780e227f5a768ec100b8344

Combined generator input freeze: 2026-08-01

Scientific status: measured, documented, or untested as labelled on each fault. Withdrawn faults are excluded from live counts after refutation.