Applied Artificial Intelligence and Computing Systems
https://www.scigate.org
Science Gate Academic
Vol. 5, No. 6, 2026

Article

Packet and Flow Evaluation Boundaries in Network Traffic Learning

Veda Norlin*, Oskar Ellery, Anwen Sorel
School of Computing, University of Utah, Salt Lake City, United States
*Corresponding author: vedanorlin126@gmail.com

Abstract: Evaluation partitions define what a network-traffic model is asked to generalize across. We generate two hundred flows with twelve packets each, a class-related signal, and a near-constant latent fingerprint shared within each flow. Across twenty independent worlds, a nearest-packet classifier obtains 99.92% mean accuracy under random packet splitting but 70.74% when entire flows are held out. A logistic model using only the class-related coordinate changes from 93.20% to 92.82%. Training and test packet counts remain 1,560 and 840 under both protocols. Original split mosaics, nearest-neighbor routes, paired accuracy changes, and a prevalence-to-precision calculation identify why a seemingly superior method can lose its advantage under a different evaluation unit. The data are generated latent features, not captured packets or a real attack benchmark. The study provides an audit of within-flow recognition versus unseen-flow prediction and distinguishes that issue from temporal or cross-network deployment validation.

Keywords: network traffic; evaluation leakage; grouped splitting; intrusion detection; prevalence shift

1 The audit claim

A classifier can recognize another packet from a familiar flow without learning a rule that transfers to a new flow. Both tasks can be legitimate, but they should not share an unlabeled headline accuracy. This paper constructs a setting in which the distinction is large and directly observable. Random packet splitting places related observations on both sides of the evaluation boundary. Holding out complete flows removes that route. The experiment asks how three simple classifiers change when the unit of generalization is made explicit.

The audit does not discover leakage in a named public dataset. Every feature and label is generated for this study. One coordinate carries a transferable class signal; six coordinates carry a flow-specific fingerprint unrelated to class across flows. Packets from the same flow have almost identical fingerprints and the same label. A nearest-neighbor method can therefore perform exceptionally when the test contains siblings of training packets. A linear classifier has less capacity to memorize arbitrary fingerprints and is more likely to use the transferable coordinate.

This setup is intentionally favorable to a diagnostic contrast. It does not represent a particular protocol field, device identifier, or attack family. The word fingerprint names a latent random vector, not an engineered network signature. No traffic is generated on a network, no system is probed, and no malicious behavior is implemented. The security context motivates careful evaluation, while the evidence comes from a statistical model of clustered observations. Its conclusions should remain attached to that model.

2 Why existing evaluation guidance matters

Intrusion-detection datasets such as CICIDS-related traffic characterization [1], UNSW-NB15 [2], and analyses of KDD-derived collections [3] provide important benchmarks and historical context. They do not supply data to this experiment. Sommer and Paxson [4] emphasize the difficulty of moving beyond a closed evaluation world, and the security-learning guidance of Arp and colleagues [5] further motivates explicit assumptions. Our generated construction isolates one issue from that broader setting: whether related records cross a partition intended to test unseen entities.

Leakage analysis [6] and the reproducibility discussion of Kapoor and Narayanan [7] explain why a plausible score can answer the wrong question. Technical debt in machine-learning systems [8] and dataset shift [9] broaden the concern beyond one split. Probability calibration [10], precision-recall relationships [11], and imbalance-aware evaluation [12] are relevant after a model reaches deployment, where positive events may be rare. We use those concepts to translate held-out-flow rates analytically, without claiming to have measured a real security operations workload.

Structured validation [13] motivates treating flows as groups. Standard machine-learning software [14], random forests [15], support-vector methods [16], and statistical learning [17] provide additional modeling context; only nearest-neighbor and logistic classifiers are executed. A 2026 leakage-safe re-evaluation of RPL-based intrusion detection [18] gives a current example of revisiting the evaluation boundary. We do not reproduce its dataset or architecture. The present contribution is a transparent generative audit where the source of near-perfect packet accuracy is known exactly.

3 The generated evidence

Each of twenty worlds contains two hundred flows and twelve packets per flow. A binary flow label is drawn with equal probability and copied to its packets. A transferable signal coordinate has class means of negative 0.8 and positive 0.8, plus a shared flow effect with standard deviation 0.2 and packet noise with standard deviation 0.5. Six fingerprint coordinates are independent standard Gaussian values per flow, perturbed by packet noise with standard deviation 0.025. Thus siblings are close in fingerprint space, while the fingerprint itself has no population-level class meaning.

$$x_{ij,0} = 0.8(2y_i - 1) + u_i + \epsilon_{ij} \tag{1}$$

$$x_{ij,1:6} = b_i + \delta_{ij} \tag{2}$$

Seeds 3801 through 3820 generate independent worlds. A packet split randomly assigns 1,560 of 2,400 packets to training and 840 to testing. A flow split randomly selects 130 complete flows for training and seventy for testing, producing exactly the same packet counts. Matching counts prevents a simple data-volume explanation for the contrast. The partitions still answer different statistical questions: the first includes familiar-flow siblings, while the second asks for prediction on new generated flow identities.

The plot displays actual memberships rather than an idealized alternating pattern. Under random packet splitting, a flow could in principle appear entirely on one side, although the chosen proportions and twelve siblings make mixed membership common. Under grouped splitting, separation is guaranteed by construction. This is a distinction between probabilistic and structural separation. Merely observing that most flows are mixed is not necessary to diagnose the evaluation question; the partition rule itself defines what information a test prediction may inherit from training.

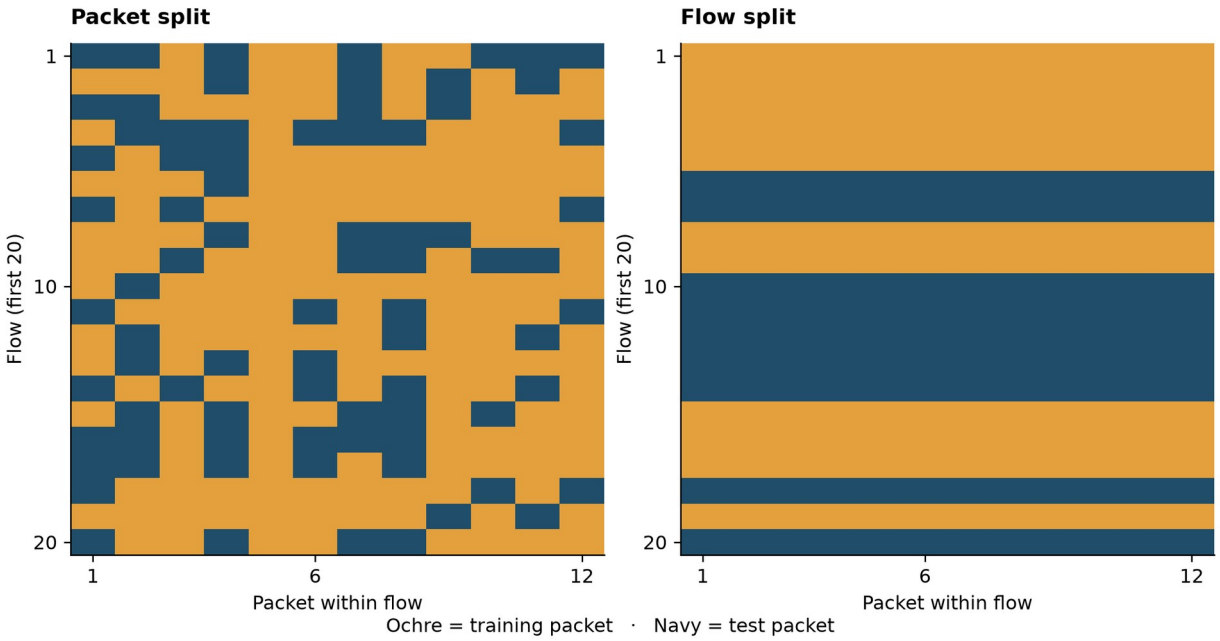


Figure 1. Actual partition membership for the first twenty flows in the first generated world. Ochre cells are training packets and navy cells are test packets. Packet splitting mixes membership within rows; complete-flow splitting keeps each row on one side.

4 Three models under two protocols

The nearest-packet classifier uses Euclidean distance in all seven raw coordinates and copies the label of the closest training packet. A logistic classifier uses the same seven coordinates, while a second logistic classifier uses only the transferable signal. Both logistic fits append an intercept, run six hundred fixed gradient steps, and use a small quadratic penalty. No model performs test-driven feature selection, hyperparameter tuning, or threshold adjustment. The decision threshold is one half for every logistic result.

The signal-only model receives an evaluator-selected feature subset based on the known generator. It is therefore a diagnostic baseline with privileged knowledge of which coordinate transfers. That advantage should not be hidden. In an actual dataset, identifying a valid feature could require domain knowledge, held-out environments, or causal analysis. The comparison shows that a simple robust signal exists in this construction; it does not demonstrate an automatic feature-selection method that would find such a signal in captured traffic.

Table 1. Mean test accuracy across twenty independent generated worlds.

Classifier	Packet split	Complete-flow split	Change in percentage points
Nearest packet	99.92%	70.74%	−29.17
Logistic: all features	93.18%	92.46%	−0.72
Logistic: signal only	93.20%	92.82%	−0.38

The nearest-packet method appears best under packet splitting and loses that position under complete-flow evaluation. Its mean accuracy drops by about twenty-nine percentage points. Both logistic models change much less. This pattern is consistent with the construction: the arbitrary fingerprint helps locate siblings but does not define a stable class rule across flows. A nearest neighbor can still perform above chance on new flows because the class-related coordinate contributes to distance, but its six unrelated fingerprint coordinates dilute that signal.

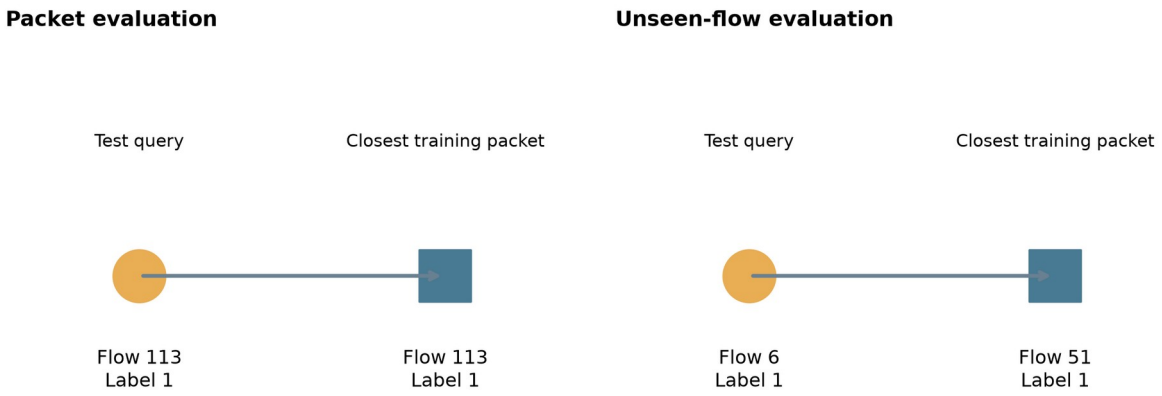


Figure 2. Nearest-training-packet route for the first test query under each protocol in the first world. Labels and flow identities come from the actual generated arrays. The diagram illustrates one query; aggregate accuracy uses every test packet and all twenty worlds.

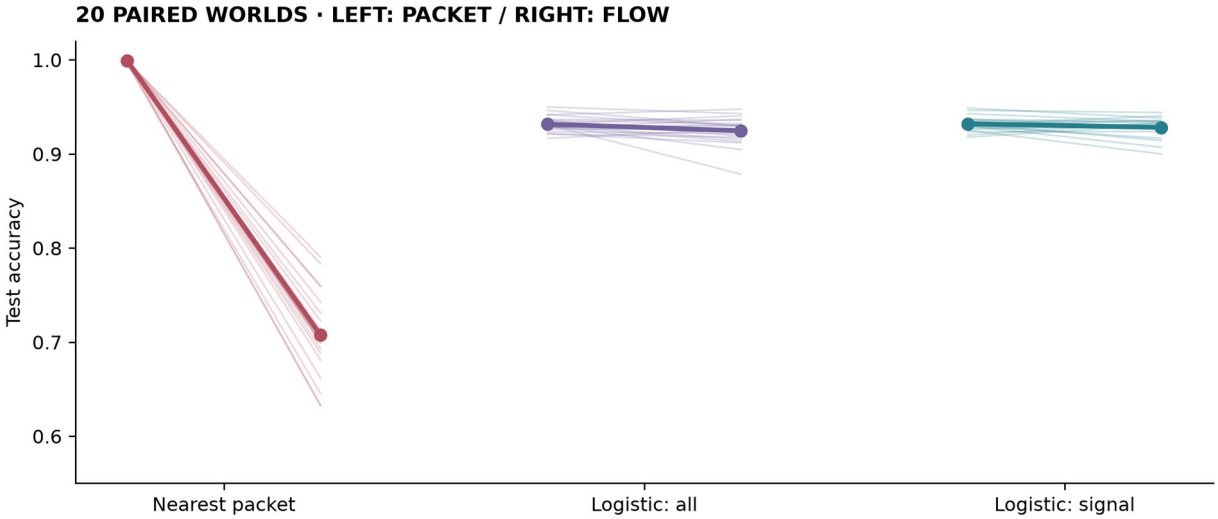


Figure 3. Paired accuracy changes within each of twenty worlds. Thin segments show individual worlds and thick segments connect protocol means. Left endpoints use packet splitting; right endpoints use complete-flow splitting. These are paired protocol comparisons, not confidence intervals.

5 The statistical unit behind the decimal places

There are 840 test packets under each protocol, but they are not 840 independent flow identities. Under complete-flow splitting they come from seventy flows, and siblings share both label and a random effect. Treating packet errors as independent Bernoulli outcomes would understate some forms of uncertainty. We report means across twenty independent worlds and display paired runs rather than deriving a narrow packet-level confidence interval. A deployment study would need uncertainty appropriate to its actual grouping, time dependence, and evaluation target.

The two protocols also differ in the relation between training and test random effects. Packet splitting exposes training samples from most test flows, which is appropriate if the intended task is labeling additional packets from already observed sessions. It is inappropriate as evidence for unseen-flow prediction when that distinction is omitted. Calling every within-group split leakage would be too broad. The problem is a mismatch between the claimed generalization unit and the information crossing the boundary. Naming the task correctly is part of the evaluation, not a cosmetic reporting choice.

Holding out flows does not establish transfer to a future time, a new device family, a different network, or an unseen attack. All flows here come from the same stationary generator and label mechanism. A complete deployment evaluation could require nested boundaries: packets grouped within flows, flows grouped within devices, and devices separated by time or environment. Each boundary removes a different information route. The current result supports one such boundary and should not be promoted into a comprehensive leakage-free security certification.

6 Rare events change the alert story

Balanced generated labels make accuracy easy to interpret, but security alerts may have a different positive prevalence. We therefore calculate analytical precision as a function of an assumed prevalence, holding the mean complete-flow true-positive and false-positive rates fixed. This is a Bayes-rule translation, not a new simulated deployment. For the signal-only model, the mean true-positive rate is approximately 0.9271 and the false-positive rate 0.0709. At a one-percent prevalence, these rates imply precision of about 11.67%, despite the balanced-test accuracy above ninety-two percent.

$$Precision(\pi) = \frac{\pi\, TPR}{\pi\, TPR + (1-\pi)\, FPR} \tag{3}$$

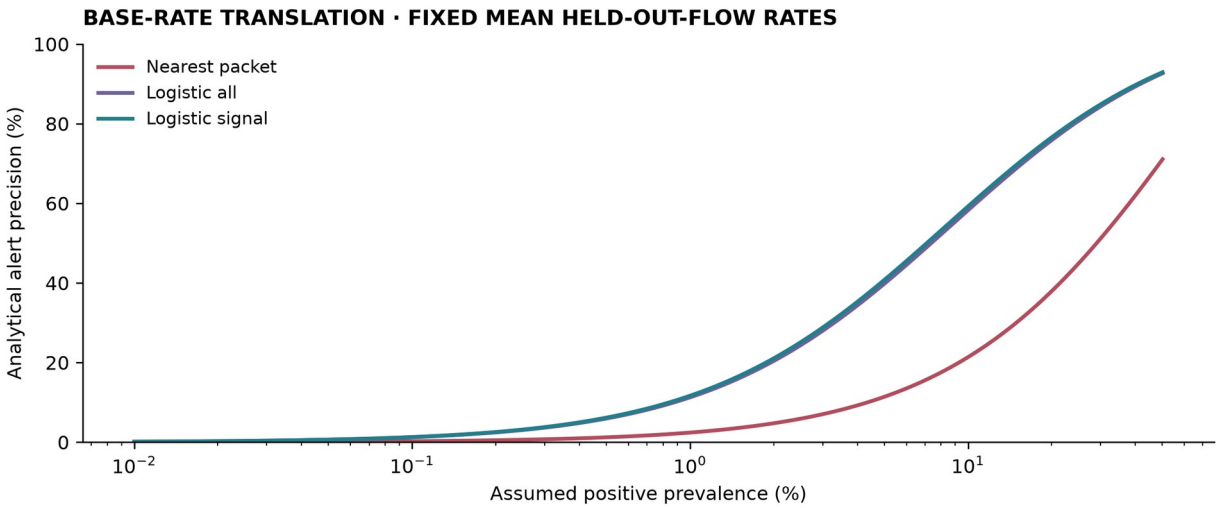


Figure 4. Analytical precision versus assumed positive prevalence using mean complete-flow rates. The calculation holds class-conditional behavior fixed and varies only the prior. It does not model adaptive attackers, threshold changes, or a measured operational alert stream.

The fixed-rate assumption is strong. A new environment may alter both prevalence and class-conditional feature distributions, changing true- and false-positive rates as well. The curve isolates the base-rate effect so that it is not confused with such distribution changes. It also uses averages of rates before applying the nonlinear precision formula, which is not identical to averaging per-world precision curves. The plotted quantity is explicitly a scenario calculation based on mean rates, rather than an empirical confidence band or a forecast for a named organization.

Threshold selection would add another decision layer. A lower false-positive rate might be obtained by accepting lower recall, but the appropriate tradeoff depends on investigation capacity and error costs. This study fixes the threshold and does not optimize an alert budget. Reporting a prevalence curve therefore clarifies a limitation of accuracy without pretending to solve deployment decision-making. A real evaluation should state the alert unit, expected event rate, threshold-selection data, and operational cost before claiming useful detection performance.

7 An audit that can be repeated

A repeatable audit needs the generator, random seeds, group identifiers, partition rules, feature subsets, and model fitting details. Group identifiers should remain available to the evaluator even if they are excluded from the predictor. Otherwise, one cannot verify that a grouped split actually separated the intended entities. In this experiment, latent fingerprints are deliberately retained as predictive coordinates so that the within-flow recognition route is visible. Removing them is another possible intervention, but would answer a different question from changing the partition while keeping the feature matrix fixed.

Preprocessing should be fitted within the training boundary if it estimates anything from data. Our raw-feature construction uses no fitted normalization, imputation, or resampling, so there is no hidden preprocessing fit to audit. Adding standardization would require specifying whether scales are learned from training packets alone. Adding oversampling would require keeping generated duplicates out of the test partition. Those are plausible extensions, not steps silently assumed to have occurred. The small baseline is useful because every information route can be enumerated.

The figures operate as audit evidence rather than security decoration. Membership tiles expose the partition. The neighbor route shows one actual prediction path. Paired run segments reveal whether the result depends on one favorable seed. The prevalence curve separates balanced accuracy from alert precision. None requires a network attack illustration or a fabricated dashboard. This design keeps attention on the statistical claim and makes it harder for a visually impressive score to conceal the unit on which that score was obtained.

8 Reading the score as a statement about information

8.1 The dependency graph crosses the split

Each packet inherits a flow label, a flow effect in its signal coordinate, and a six-dimensional fingerprint. These shared variables create dependence among siblings. A random packet split does not remove that dependence; it places its manifestations in both training and testing. The nearest-neighbor classifier can use the fingerprint to connect a test packet with a labeled sibling. Its accuracy therefore reflects access to a familiar latent identity. That route is allowed by the packet-prediction task but should not be credited as evidence of transfer to a new flow.

The explicit flow identifier is not itself included as a feature. This matters because removing a visible identifier does not necessarily remove information about identity. A collection of near-constant coordinates can act as an implicit identifier even if each column has an innocent name. In the generator, the fingerprint is deliberately constructed for this purpose. In real data, identifying such routes requires domain knowledge and grouped evaluation. A feature list alone may not reveal how several coordinates jointly encode an entity or acquisition session.

The class label is independent of the fingerprint across flows. That independence is a population statement, not a guarantee that finite training samples show zero accidental association. A flexible model can fit such associations, and a simple linear model may also pick up some finite-sample noise. Holding out flows tests whether those fitted relations transfer to independently generated identities. The logistic-all model's small degradation relative to signal-only is consistent with the extra coordinates adding little stable information, while the nearest-neighbor result reveals their strong value for sibling recognition.

8.2 Why a nearest packet is usually a sibling

Within a flow, fingerprint differences are small because packet noise has standard deviation 0.025 in each of six coordinates. Between flows, the underlying fingerprint vectors differ on the scale of independent standard Gaussians. This creates a strong separation in distance even before considering the signal coordinate. When at least one sibling is in training, it is likely to be much closer than packets

from unrelated flows. The classifier then copies a label that is shared by construction, explaining the near-perfect random-packet score without requiring a meaningful class boundary in fingerprint space.

The signal coordinate has larger packet noise, so sibling distances are not exactly zero and unrelated packets can occasionally be closer. That is why the result is near perfect rather than guaranteed perfect. The raw Euclidean metric also assigns equal numerical weight to all coordinates, making the six fingerprint dimensions collectively influential. A different scaling or distance metric could change the balance. The paper states the raw-feature convention rather than implying that nearest-neighbor behavior is invariant to preprocessing or that every distance-based classifier will show the same gap.

Under complete-flow splitting, no sibling can be present in training. The nearest neighbor must come from another latent fingerprint. Its label is not determined by fingerprint proximity in the population. The transferable signal still contributes to distance, so accuracy need not fall to fifty percent. The observed mean around seventy-one percent is therefore consistent with a mixture of useful signal and irrelevant geometric dimensions. This interpretation is more precise than saying the model completely fails or that every part of its packet accuracy was artificial.

8.3 A split defines a conditional prediction problem

Packet evaluation asks for a label given observations from a population in which the same flow may already have labeled packets in training. Flow evaluation asks for a label when that identity has not been observed. These are different conditional information sets. Neither can be converted into the other merely by reporting more decimal places or by using the same nominal train-test percentage. The relevant boundary is the entity whose novelty the claim promises. The experiment matches packet counts to show that quantity alone does not define that boundary.

An application that labels later packets within a known session may legitimately benefit from earlier labeled packets. Its evaluation should preserve chronological order and explain where those earlier labels come from. Our random packet split does not model chronology or label availability, so it should not be treated as a deployment simulation even for that task. It is a diagnostic partition that allows sibling information. A realistic within-session study would need a separate protocol with causal access to past information only.

An application that detects a previously unseen flow needs grouped separation at minimum, but may require more. If several flows share a device, user, capture file, or attack campaign, holding out individual flows can still leave higher-level identity routes. The correct grouping depends on the target. This paper's generated hierarchy stops at flow identity, so complete-flow splitting removes the only constructed identity level. That fact makes the experiment internally clean while limiting its scope. It does not certify that the same rule is sufficient for every traffic dataset.

8.4 Matching counts does not match composition

Both protocols use 1,560 training and 840 test packets, but the class counts can differ because labels are drawn at the flow level and partitions are not explicitly stratified by class. This is part of the declared random construction. Twenty independent worlds reduce reliance on one class composition, and true- and false-positive rates are reported for the prevalence calculation. A reproduction should not silently add stratification and expect exact numerical agreement, although such a design could be a reasonable extension if its objective were stated.

The number of unique training identities differs in the way each protocol uses them. Packet training can include some packets from many or nearly all flows, while flow training includes all packets from exactly 130 identities. This difference is the mechanism being tested, not an accidental imbalance to be normalized away. Matching unique-flow count as well as packet count would require a different experiment. The present comparison asks what happens when a common packet-level split is replaced by the appropriate complete-entity boundary for an unseen-flow claim.

Training examples also have correlated information. Twelve siblings do not provide twelve independent fingerprints, although their signal noise varies. An algorithm that memorizes a flow can benefit from one sibling differently from an algorithm estimating a class-related mean. A sample-size statement should therefore report both packets and flows. The same principle applies to images from one patient or measurements from one machine, but the current numerical result is about the generated traffic hierarchy. The broader analogy should not be mistaken for empirical evidence in those other domains.

8.5 Repeated worlds and paired protocol changes

The twenty seeds regenerate labels, fingerprints, flow effects, and packet noise. This is broader replication than repeating model optimization on one fixed dataset, because each world has a new population of generated flows. Within a world, both protocols use the same full feature array, so their comparison is paired at the world level. The split itself differs by design. Figure 3 shows those paired changes, allowing the reader to see whether the nearest-neighbor gap is consistent or driven by a small number of unusual worlds.

The logistic fits are deterministic given their training arrays and fixed optimization rule. Their variation across worlds therefore reflects data and partition differences rather than random initialization. The nearest-neighbor model has no iterative training, so its score depends on the stored examples and metric. Keeping these details explicit helps interpret variability. A future deep-model comparison would add optimization randomness and might need several fits within each world to separate that variation from the variation in generated flow populations.

A statistical test on twenty paired changes could summarize evidence for a mean difference under the generator, but it would not establish deployment relevance. The central mechanism is already visible in the construction and the neighbor routes. We report descriptive outcomes and avoid turning a small synthetic experiment into a claim about a universal security-model ranking. Replication across worlds strengthens internal evidence for the designed effect; external validation still requires real features, credible labels, and a boundary appropriate to the operational task.

8.6 The signal-only baseline is informative but privileged

The evaluator knows which coordinate carries the transferable class signal because it wrote the generator. Restricting a model to that coordinate is therefore an oracle feature-design choice. It is useful as a diagnostic: it proves that a stable predictive relation exists in the data and gives a simple reference for how much the full-feature model loses. It is not an automatic procedure for finding robust features. The paper credits this knowledge explicitly rather than presenting signal-only selection as a learned discovery.

In a real network dataset, a feature can be predictive for a reason that changes across environments. A packet-length statistic may reflect an attack in one collection and a benign application in another. Removing obvious identifiers does not guarantee causal stability. Testing several environments, time periods, or devices can help evaluate transfer, but each adds a new assumption about the deployment population. Our single stationary generator avoids those complications so that the identity route is unambiguous. That clarity is a methodological benefit, not a substitute for multi-environment evidence.

The all-feature logistic model is less able than nearest neighbor to memorize arbitrary local fingerprint neighborhoods, but it is not immune to spurious association. A linear boundary can still exploit accidental correlations in a finite set of flow fingerprints. The small difference between its packet and flow results should therefore not be promoted into a guarantee that linear models cannot leak. Model capacity changes the route and degree of exploitation; the evaluation boundary remains necessary regardless of the function class.

8.7 Accuracy does not specify an alert workload

At a balanced prevalence, accuracy averages correct positives and correct negatives with approximately equal weight. A rare-event deployment changes those weights. Even a moderate false-positive rate can produce many more false alerts than true alerts when negatives dominate. The analytical precision curve isolates this arithmetic. It does not claim that the balanced experiment has measured a realistic attack rate or that its labels correspond to a particular operational incident. The positive class is a generated binary variable throughout.

The relevant alert unit also needs definition. A model may produce one decision per packet, but an analyst may investigate one flow or one host. Aggregating packet alerts can change both workload and detection probability. The present experiment evaluates packet predictions under a grouped test boundary, not an operational alert-deduplication pipeline. Its prevalence calculation applies to the same prediction unit as the reported rates. A real system should translate to its investigation unit explicitly rather than multiplying packet-level precision by an assumed traffic volume.

Fixed class-conditional rates are another strong assumption. A new environment can alter the feature distribution of positives and negatives, so both recall and false-positive rate may change along with prevalence. The curve therefore represents a prior-shift scenario, not a general deployment forecast. This separation is useful because it shows how much interpretation changes even before adding harder distribution shifts. It also prevents a low analytical precision from being attributed to a newly simulated failure that the experiment never generated.

8.8 Thresholds and costs should be selected outside the test set

A decision threshold of one half is convenient for the balanced generated task. It is not automatically appropriate for a rare-event security application. A threshold might instead be chosen to satisfy an alert budget or a false-positive constraint using separate validation data. Doing so would trade recall against workload. This paper does not tune that decision layer, so its rates should be interpreted as one operating point. The prevalence plot does not imply that no better operating point is available to the logistic models.

Nearest-neighbor labels are hard decisions in this implementation. A distance-based confidence score or a larger neighborhood could produce a thresholdable model, but would change the method. A fair comparison of operating curves would need those scores and a shared selection protocol. We do not invent an ROC or precision-recall curve from one hard-label point. The only continuous curve shown is the analytical prevalence transformation with fixed rates, which is clearly identified as a scenario calculation rather than a threshold sweep.

Operational costs can differ sharply between missing a positive and investigating a false alert. Those costs may also vary by event type. A single balanced accuracy cannot encode them, and a synthetic binary label cannot supply their real values. The experiment therefore stops at rates and a transparent prevalence calculation. Its audit lesson is about what must be specified before a score becomes a deployment claim: the unit, boundary, operating point, prior, and decision cost all contribute to that interpretation.

8.9 A reproducible audit trail

The full generated feature matrix is preserved before either split is created. Packet and flow partitions are then constructed from their stated rules. Nearest neighbors are searched only among training indices, and logistic fitting uses only the permitted feature columns and training labels. These boundaries are simple enough to inspect directly. A reproduction should also verify that no flow identifier accidentally enters the signal-only predictor and that complete-flow test identities are disjoint from training identities. The plotted split mosaic provides a visual check of the latter rule for a small visible subset.

The route diagram uses the first test query from each protocol in the first world, rather than selecting a query that maximizes the apparent leakage effect. Its two examples happen to have matching predicted labels, which is acceptable because their purpose is to show identity routes, not to dramatize an error. Aggregate performance comes from all test packets and all worlds. A diagnostic figure need not show a failure in every panel to be useful; it should show the mechanism honestly and state its selection rule.

The final reporting chain keeps data construction, prediction, aggregation, and scenario translation separate. This makes the study repeatable and limits what it claims. The near-perfect packet score is a real result of the generated task, while the unseen-flow score answers a different question. The analytical precision curve adds a conditional deployment interpretation without inventing traffic. Preserving these distinctions is the practical output of the audit and a useful standard for reporting more elaborate network-learning experiments.

8.10 Correlation limits the effective sample size

For an exchangeable group of equally correlated errors, the variance of an average contains a design effect of one plus group size minus one times the within-group correlation. This familiar calculation illustrates why twelve sibling packets do not necessarily provide twelve independent assessments of performance. We do not estimate that correlation or use the formula to assign an effective sample size to the current results. It is an analytical warning against deriving overly narrow uncertainty from the packet count while ignoring the flow hierarchy.

$$Var(\hat{m}) = \frac{\sigma^2}{n}[1 + (m - 1)\rho] \tag{4}$$

The formula assumes equal group size, a common marginal variance, and a common within-group correlation, with independent groups. Real traffic can violate all three assumptions. A grouped bootstrap, repeated independent captures, or an explicit hierarchical model may be more appropriate depending on the target. The generated experiment instead repeats complete worlds, which provides a clear independent unit at the outer level. The displayed twenty paired outcomes should be interpreted at that level rather than converted into millions of independent feature-coordinate observations.

Unequal flow lengths would create another weighting choice. A packet-level mean gives long flows more influence, while an equal-flow mean treats each identity equally. Our twelve-packet construction makes those averages coincide within a protocol, avoiding that additional confound. A real benchmark should report both counts and its aggregation rule. Otherwise, a method that improves on a few long flows can appear broadly superior even if it performs poorly on many short ones. The present study does not test variable-length flows, but its fixed-size design makes the omitted issue explicit.

8.11 Deduplication and grouping solve different problems

Removing exactly duplicated rows can prevent one obvious form of overlap, but siblings need not be exact duplicates to carry identity. The generated fingerprint coordinates differ slightly within a flow, so an exact duplicate filter would leave the main recognition route intact. Grouped evaluation removes the shared identity regardless of small numerical differences. This distinction is useful when a dataset appears clean after deduplication: the absence of identical rows does not establish independence or an unseen-entity test.

Conversely, grouping alone does not repair every preprocessing leak. A normalization, feature selector, or learned embedding fitted on the full dataset can still incorporate test information before groups are separated. A complete audit must follow the whole data path. Our experiment uses raw generated coordinates and evaluator-declared feature subsets, so there is no fitted preprocessing stage. That simplicity helps isolate the partition effect. More elaborate pipelines should preserve training boundaries for every learned transformation and state whether any unlabeled test access is intentionally permitted.

A temporal split answers another question again. It can prevent future information from entering training while still allowing the same device or flow identity to appear across the boundary, depending on the task. A grouped split can hold identities out while ignoring temporal drift. Neither should be used as a universal synonym for realistic validation. The appropriate design follows the novelty promised by deployment. The current complete-flow protocol is sufficient for the generated unseen-flow task because the generator contains no higher hierarchy or time trend.

8.12 A feature audit can be mechanistic

One can ask whether a coordinate varies mainly within or between flows, whether it is stable across labels, and whether its predictive value survives new identities. These questions help distinguish a transferable class signal from an identity proxy. In this generator the answers are known by construction. In measured data they would be empirical and could depend on acquisition conditions. A feature's name is not enough: several weak proxies can jointly encode an identity even when no explicit identifier is present.

The neighbor-route diagram is a simple mechanistic explanation for a memory-based model. It shows which training observation supplied a test label and whether that observation shares the same flow. For a more complex model, attribution methods might identify influential features, but attribution alone would not establish transfer. A grouped test remains necessary because a model can assign importance to an identity proxy for perfectly understandable reasons. Explanation and validation answer complementary questions: how a prediction was made and whether that route supports the claimed use.

An ablation that removes fingerprints would provide another diagnostic, but the signal-only logistic model is not exactly that ablation for nearest neighbor. It changes both the feature subset and the model class. The paper therefore avoids attributing the entire difference between those two models to feature removal alone. The controlled protocol comparison holds each classifier's features fixed, while the signal-only model supplies a privileged reference. Keeping those roles distinct prevents a multi-change comparison from being mistaken for a clean single-factor experiment.

8.13 What the audit would ask of a claimed improvement

A new method should state whether it improves familiar-packet labeling, unseen-flow prediction, future-time transfer, or another target. It should retain the corresponding grouped or temporal boundary throughout preprocessing and model selection. It should report the number of independent entities and the operating point used for any deployment interpretation. These requirements do not favor a particular architecture. They make an improvement interpretable and prevent a changed evaluation task from masquerading as better learning.

The generated result is valuable because the mechanism is visible without relying on an accusation about another study. The nearest-neighbor packet score is numerically correct for its partition, and the flow score is numerically correct for a different information boundary. The error would be to attach the same unseen-flow claim to both. This framing allows an audit to be precise: identify the claimed target, trace the permitted information, and compare them. It avoids turning every high score into a suspicion or every grouped split into a certification.

A defensible final report can therefore be concise about the outcome while complete about the protocol. The large nearest-neighbor drop demonstrates identity dependence in this construction, the stable signal model demonstrates available transfer, and the prevalence calculation demonstrates the limits of balanced accuracy for alerts. Together they establish a clear methodological case. Real network validation remains necessary, but it can begin with better specified questions and a reproducible audit trail rather than with an unexplained near-perfect percentage.

In the illustrative variance formula, n denotes the total number of observations, m the common number per group, rho the within-group correlation, and sigma squared the marginal error variance. No

numerical value is assigned to these quantities from the current predictions. Defining them prevents the equation from being mistaken for a fitted uncertainty estimate. The study's actual uncertainty presentation remains the repeated-world comparison. This distinction keeps an explanatory statistical identity separate from an empirical result and makes clear which claims have been calculated from the generated arrays.

9 Audit conclusion

Near-perfect packet accuracy can primarily reflect recognition of familiar flow fingerprints in the generated construction. Holding out complete flows reverses the apparent advantage of nearest-neighbor memorization while leaving a transferable signal-based classifier comparatively stable. The result does not diagnose a specific public dataset or establish real intrusion-detection performance. It establishes a reporting requirement: name the generalization unit, enforce its boundary, and interpret operating metrics at the intended prevalence. Those decisions determine what an accuracy number can legitimately mean.

References

[1] I. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,” Proceedings of the 4th International Conference on Information Systems Security and Privacy, pp. 108-116, 2018. doi: 10.5220/0006639801080116.

[2] N. Moustafa, J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1-6, 2015. doi: 10.1109/milcis.2015.7348942.

[3] M. Tavallaee, E. Bagheri, W. Lu et al., “A detailed analysis of the KDD CUP 99 data set,” 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6, 2009. doi: 10.1109/cisda.2009.5356528.

[4] R. Sommer, V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” 2010 IEEE Symposium on Security and Privacy, pp. 305-316, 2010. doi: 10.1109/sp.2010.25.

[5] D. Arp, E. Quiring, F. Pendlebury et al., “Dos and Don’ts of Machine Learning in Computer Security,” Proceedings of the 31st USENIX Security Symposium, pp. 3971–3988, 2022. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/arp.

[6] S. Kaufman, S. Rosset, C. Perlich et al., “Leakage in data mining,” ACM Transactions on Knowledge Discovery from Data, vol. 6, pp. 1-21, 2012. doi: 10.1145/2382577.2382579.

[7] S. Kapoor, A. Narayanan, “Leakage and the reproducibility crisis in machine-learning-based science,” Patterns, vol. 4, pp. 100804, 2023. doi: 10.1016/j.patter.2023.100804.

[8] D. Sculley, G. Holt, D. Golovin et al., “Hidden Technical Debt in Machine Learning Systems,” Advances in Neural Information Processing Systems, vol. 28, 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html.

[9] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer et al., “Dataset Shift in Machine Learning,” The MIT Press, 2008. doi: 10.7551/mitpress/9780262170055.001.0001.

[10] C. Guo, G. Pleiss, Y. Sun et al., “On Calibration of Modern Neural Networks,” arXiv preprint arXiv:1706.04599, 2017. [Online]. Available: https://arxiv.org/abs/1706.04599.

[11] J. Davis, M. Goadrich, “The relationship between Precision-Recall and ROC curves,” Proceedings of the 23rd international conference on Machine learning - ICML '06, pp. 233-240, 2006. doi: 10.1145/1143844.1143874.

[12] T. Saito, M. Rehmsmeier, “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,” PLOS ONE, vol. 10, pp. e0118432, 2015. doi: 10.1371/journal.pone.0118432.

[13] D. R. Roberts, V. Bahn, S. Ciuti et al., “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure,” Ecography, vol. 40, pp. 913-929, 2017. doi: 10.1111/ecog.02881.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html.

[15] L. Breiman, “Random Forests,” Machine Learning, vol. 45, pp. 5-32, 2001. doi: 10.1023/a:1010933404324.

[16] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” Data Mining and Knowledge Discovery, vol. 2, pp. 121-167, 1998. doi: 10.1023/a:1009715923555.

[17] T. Hastie, R. Tibshirani, J. Friedman, “The Elements of Statistical Learning,” Springer Series in Statistics, 2009. doi: 10.1007/978-0-387-84858-7.

[18] M. Raj, H. Tyagi, A. Kaur, “A leakage-safe re-evaluation of machine-learning intrusion detection for RPL-based IoT networks,” Journal of Engineering Research, 2026. doi: 10.1016/j.jer.2026.09.059.