

A Leakage-Controlled Fail-Closed Protocol for Prospective-Replay Evaluation of Adaptive Network Intrusion Detection

Bilal Rafique

Southwest Jiaotong University

Keming Wang

kmwang@swjtu.edu.cn

Southwest Jiaotong University

Sania Kanwal

Southwest University of Science and Technology

Yasir Rafique

University of Haveli Kahuta

Zahid ur Rahman

Hubei University of Automotive Technology

Research Article

Keywords: network intrusion detection, network management, adaptive learning, concept drift, fail-closed prospective-replay evaluation

Posted Date: October 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11247634/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Leakage-Controlled Fail-Closed Protocol for Prospective-Replay Evaluation of Adaptive Network Intrusion Detection

Bilal Rafique ¹, Keming Wang^{1*}, Sania Kanwal ,
Yasir Rafique , Zahid ur Rahman 

^{1*}School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China.

²School of Information and Control Engineering, Southwest University of Science and Technology, Mianyang, 621010, China.

³University of Haveli Kahuta, Kahuta, Azad Jammu and Kashmir, 12350, Pakistan.

⁴School of Automotive Business, Hubei University of Automotive Technology, Shiyan, 442002, China.

*Corresponding author(s). E-mail(s): kmwang@swjtu.edu.cn;

Contributing authors: bilalrafique@my.swjtu.edu.cn;

saniakanwal@mails.swust.edu.cn; yasirrafiquebscs@gmail.com;

20260023@huat.edu.cn;

Abstract

Adaptive intrusion-detection studies can report post-update accuracy even when event construction, score generation, or pre-change detector behavior is invalid. We present an offline, leakage-controlled, fail-closed protocol for chronological prospective replay of adaptive network intrusion detection. Gate 1 retrospectively constructs a label-audited event, Gate 2 audits score validity, Gate 3 selects an operating point on a disjoint 5,000-flow pre-change guard-selection segment, and Gate 4 conditionally executes static and adaptive evaluation. The protocol was applied to NF-CSE-CIC-IDS2018-v3 and NF-UNSW-NB15-v3 with an expandable nearest-prototype model. Ten seeds describe training randomness on each fixed stream. DoS was the only core prospective-replay episode admitted to Gate 4. Random B50 increased normalized recovery area from 0.381 to 0.875 and novel-class recall from 0.000 to 0.843. Each NF-UNSW episode produced 90 defined Gate 4 runs. When averaged over update rules,

uncertainty-diversity exceeded Random at B50 on Exploits (0.325 versus 0.309), whereas Random exceeded uncertainty-diversity on Reconnaissance (0.332 versus 0.315). The protocol provides offline evaluation governance with explicit partial-outcome semantics. It does not provide an independent pre-change safety test, an online deployment controller, or deployment-level inference. The external episodes share one recorded trace, and the evidence does not establish a general acquisition-method ranking.

Keywords: network intrusion detection, network management, adaptive learning, concept drift, fail-closed prospective-replay evaluation

1 Introduction

Random train-test partitions and a single terminal classification score remain common in evaluations of network intrusion detection systems (NIDSs). This setup assumes a fixed class vocabulary and immediate access to representative labels. Operational streams violate both assumptions. Benign traffic evolves, attack prevalence changes, previously unseen families enter the stream, and labels arrive only after observations have been processed. Cross-dataset studies show that strong results within one dataset may not transfer to another environment [1]. Random partitioning can also mix temporally related flows. If preprocessing is fitted before the split, it can use future traffic. Both practices can inflate prospective estimates [2, 3]. In this paper, *prospective replay* means an offline chronological replay that preserves the information order available at each simulated decision; it does not mean a live prospective deployment study.

Temporal consistency requires chronological separation so that future observations cannot influence fitted transforms or model selection [2, 3]. Deployment-oriented assessments also call for clear reporting of the operating context, configuration, and limitations [4, 5]. We retain an auditable evidence record for every gate decision for this reason. Evaluation of security streams must account for delayed labels and evolving classes [6]. Dataset defects can also distort traffic diversity, labels, and the intended generalization task [7]. These issues carry particular weight for adaptive detectors because every reported post-update score depends on several earlier decisions.

We express these decisions as four ordered validity gates. First, a chronological event must satisfy prespecified temporal and prevalence rules. Second, the change score must pass a numerical audit and remain independent of post-change labels and rows. Third, the protocol evaluates several detector candidates on a disjoint pre-change guard-selection segment and selects the most sensitive candidate with no disallowed detector event. Fourth, it permits the frozen replay branch to proceed to label acquisition and adaptation only after that selection criterion is met. A failure at any earlier gate stops the run and leaves all later metrics undefined. Because Gate 3 uses the same segment for candidate screening and acceptance, it is not an independent safety test and does not estimate a future false-alarm rate. The executed gates form an offline evaluation-control plane. A live operational plane would require observable telemetry,

delayed or uncertain analyst labels, approval and rollback rules, and an independent acceptance period; those elements were not evaluated here.

We developed the core experiment on a distributed denial-of-service (DDoS) episode, then froze it before prospective application to denial-of-service (DoS), Web Attacks, and Bot. Infiltration was used as a secondary test of temporal feasibility. A lightweight expandable prototype classifier provided distance-based rejection and allowed the model to add a new class prototype after labels had been acquired. These open-set and label-selection mechanisms are controlled evaluation components. They are not the paper’s principal methodological claim.

Once the DoS episode passed the guard-selection criterion, the experiment compared an explicit static no-update baseline with two acquisition rules, two label budgets, and two prototype-update rules. Model-seed and query-seed analyses were kept separate so that different sources of algorithmic randomness were not pooled. The design separates four questions that are often combined: whether an evaluation branch was admissible, whether adaptation performed better than no update, whether a targeted selector performed better than random acquisition, and whether a more conservative update rule reduced forgetting.

We next applied the protocol to NF-UNSW-NB15-v3. The external sequence retained every failed construction and diagnostic exclusion. Under the original event rule, none of the three ordered episodes could be constructed. A construction-only amendment produced Exploits and Reconnaissance exposures, while DoS remained a failed construction. A development trace then exposed unstable, near-zero reference variances, so the original detector outputs were excluded. The corrected source-anchored score supported a MAD, ADWIN, and Page-Hinkley comparison after guard-based operating-point selection. Exploits later entered Gate 4 as a prespecified external extension whose broad contrast set remains descriptive. After those results were known, the Reconnaissance Gate 4 grid and three primary contrasts were frozen without examining Reconnaissance outcomes. This second episode is a prespecified within-trace second episode, not untouched validation, independent replication, or an independent deployment environment.

The study was organized around four research questions:

- RQ1:** Which prespecified episodes pass temporal construction and the frozen guard-selection criterion?
- RQ2:** Conditional on evaluation admission, how do the adaptive alternatives compare with a static no-update baseline across the admitted core and external episodes?
- RQ3:** How do recovery, novel-class recall, and forgetting vary with acquisition strategy, label budget, update rule, model seed, query seed, and episode?
- RQ4:** On corrected external score traces, do MAD, ADWIN, and Page-Hinkley pass the same guard, and how do their post-change delays compare?

The paper makes four contributions that match its stated claims:

- (1) We formulate adaptive NIDS evaluation as a partial decision process with four ordered gates, an explicit admissibility predicate, and a deterministic stopping

rule. The record retains construction failures, invalid scores, and guard-selection failures as outcomes. Downstream metrics remain undefined instead of being imputed or silently omitted.

- (2) We implement the protocol with source-only preprocessing, prediction-before-update scoring, numerically audited change scores, separate calibration and pre-change guard-selection partitions, and row-level provenance for every queried label. These controls make the full route to an adaptive result auditable, rather than exposing only the final classifier score.
- (3) Within admissible branches, we compare static and adaptive operation, random and uncertainty-diversity acquisition, label budgets of 50 and 200, naive and source-anchored prototype updating, and MAD, ADWIN, and Page-Hinkley on shared score traces. The core DoS case, the descriptive Exploits extension, and the prespecified Reconnaissance conditional computational family have separate analysis boundaries.
- (4) We release a versioned evidence record for two NetFlow datasets. It retains failed episode constructions, guard failures, excluded numerical diagnostics, defined outcomes, configurations, query provenance, archive provenance, omission inventories, and integrity hashes. The record connects each conclusion to the gate decisions that made its outcome defined or undefined [8].

Table 1 fixes the evidentiary meaning of terms used across protocol versions. These labels describe when information was frozen and which data were reused; they do not convert a fixed replay into a live deployment study.

Table 1 Terminology and evidentiary boundaries used in this study.

Term	Meaning in this manuscript	What it does not mean
Prospective replay	Offline chronological replay that preserves the information order at each simulated decision	Live deployment or prospective clinical-style enrollment
Prespecified	Configuration or analysis frozen before the named outcome was examined	Independence from earlier datasets or episodes
Outcome-free	No outcome directory for the named version existed when its plan was frozen	No prior knowledge from related versions
External extension	Evaluation on the second dataset, NF-UNSW-NB15-v3	Independent network deployment
Within-trace second episode	Reconnaissance was frozen after Exploits analysis but before Reconnaissance outcomes were examined	Untouched validation or independent replication
Guard-selection segment	Pre-change rows disjoint from reference, calibration, and post-change monitoring, used to select and accept an operating point	An untouched final safety test or long-run false-alarm estimate
Conditional computational analysis	Variation over frozen randomized training procedures conditional on one fixed stream	Population inference over networks, periods, or attacks

2 Related Work and Research Gap

2.1 Prospective and Deployment-Aware Evaluation

Common NetFlow representations make features more consistent across datasets [9, 10], but they do not make collection environments interchangeable. Cross-dataset tests have reported substantial degradation tied to collection tools, parameters, hardware, and topology [1]. Dataset audits have also identified construction defects that reduce traffic diversity, obscure labels, or leave the generalization task unclear [7]. Considered together with evidence of temporal bias [2, 3], these findings support chronological splits, source-only fitting, and explicit event provenance.

Operational assessment of a NIDS includes more than classifier accuracy. Sommer and Paxson [4] described the gap between benchmark results and deployment. Apruzzese et al. [5] proposed assessments that account for operating context, adversarial conditions, and configuration choices. Sequential evaluation is needed when temporal dependence makes randomized validation optimistic [11]. Security learning on streams introduces delayed labels, concept evolution, and adversarial behavior [6]; recent work also evaluates adaptation with delayed and sampled labels [12]. From a network-management perspective, operational intrusion detection requires observable decisions, response workflows, and management roles rather than an accuracy-only framing [13, 14]. Our executed contribution is narrower: it controls progression through an offline adaptive evaluation and records why a branch was admitted or stopped.

2.2 Change Detection, Score Validity, and Guard Calibration

Concept drift may be sudden, gradual, incremental, or recurrent [15, 16]. Detectors monitor prediction errors, feature summaries, model disagreement, or other stream statistics. ADWIN compares adaptive subwindows [17], whereas Page-Hinkley follows the cumulative-deviation principle introduced by Page [18]. Median absolute deviation provides a robust estimate of scale [19]; here, it defines a fixed threshold from calibration data. Reviews of adaptive anomaly detection report wide variation in the monitored signals, update rules, and evaluation designs [20].

A detector comparison is interpretable only when the input signal and acceptance criterion are controlled. Numerical score validity must also be checked before latency is measured because a standardized squared contribution is proportional to s_j^{-2} and a nonzero deviation can dominate the multivariate score as its scale s_j approaches zero. Our external comparison therefore gives MAD, ADWIN, and Page-Hinkley the same source-preprocessed score trace and enforces an explicit scale floor. Calibration scores rank candidates, and a separate 5,000-flow pre-change segment selects the most sensitive candidate with zero disallowed detector-specific trigger events. Because this second segment is inspected across four MAD, five ADWIN, and four Page-Hinkley settings, it is a guard-selection segment rather than an independent safety test. Candidate-grid sizes differ, and detector delays were not matched to a common estimated false-alarm rate. ADWIN and Page-Hinkley were implemented with the River package [21] at the exact frozen release described in [22].

2.3 Conditional Adaptation and Label Acquisition

Open-set recognition allows rejection when an observation has insufficient support among known classes [23, 24]. Prototype-based NIDS methods use this mechanism to identify unsupported traffic [25]. Dynamic open-set NIDS methods can then update the model and expand retained knowledge [26]. Continual methods manage the balance between learning a new class and preserving earlier knowledge [27–30]. Our nearest-prototype classifier follows this functional sequence, but its simple form keeps the evaluation decisions auditable.

In this design, adaptation depends on the realized class composition of the queried labels. Uncertainty sampling targets ambiguous observations [31], whereas core-set selection seeks coverage in the representation space [32]. NIDS studies have used reinforcement learning for zero-day selection [33] and selective labeling under distribution shift [34]. We treated insufficient emerging-family yield under class imbalance as a study hypothesis, not as an established property of geometry-based selection. The experiment therefore compares a prespecified uncertainty-diversity heuristic with random acquisition and reports realized class composition, exact row provenance, recovery, and forgetting.

2.4 Research Gap, Scope, and Contributions

Recent NIDS studies have advanced semi-supervised continual learning, adaptive ensembles, few-shot learning of new classes, and replay-based adaptation under drift [28–30, 34]. SmartVille [35] formulates online, open-world, multimodal deep-learning NID and provides a controllable implementation for training and benchmarking such systems. Related studies have examined safety-constrained self-healing and computationally efficient staged intrusion detection [36, 37]. These works strengthen detector design, training, or update components. Our protocol addresses a complementary evaluation-governance question: whether the chronological event, monitored score, and selected operating point make adaptation and downstream performance claims admissible within an offline replay. If these conditions are treated as informal preprocessing choices, a later accuracy value can conceal an earlier validity failure.

The four research questions examine this dependency at separate levels. RQ1 addresses admissibility by asking which prespecified episodes survive temporal construction and the guard-selection criterion, with every failure retained in the evaluation denominator. RQ2 addresses conditional benefit: the study compares adaptation with a static no-update baseline only after an episode is admissible. RQ3 addresses component attribution. Acquisition strategy, label budget, update rule, model seed, and query seed vary through paired or deliberately separate analyses, and the realized class composition of each query set is recorded. RQ4 addresses detector comparability. MAD, ADWIN, and Page-Hinkley receive the same corrected score trace; the same calibration and guard-selection partitions screen their candidate settings; and delay is interpreted only for accepted detector branches.

The technical contribution is a validity-preserving evaluation architecture rather than a new classifier or selector. It contains four elements. First, the admissibility predicate and partial-outcome rule define when a downstream result exists. Second, leakage

and score audits bind each result to source-only fitting, stable numerical scales, and prediction-before-update execution. Third, the guard-selection criterion makes pre-change candidate behavior explicit while retaining its selection dependence. Fourth, the evidence record connects gate decisions, selected labels, model updates, outcome support, and integrity hashes for successful and failed branches. These elements provide an auditable answer path for RQ1–RQ4, rather than a leaderboard comparison separated from evaluation eligibility.

The scope is deliberately narrow. The expandable prototype classifier, uncertainty-diversity heuristic, and three change detectors operate as controlled components within the protocol. We do not claim that they outperform recent deep, ensemble, or continual NIDS methods. Published accuracy values from those methods are not directly comparable because the data partitions, label access, task boundaries, output spaces, and update schedules differ. A quantitative model-level comparison would require each method to be rerun under the same chronological events, source-only transformations, guard rule, label budgets, prediction-before-update order, and outcome definitions.

3 Materials and Methods

3.1 Protocol Overview and Evidence Boundary

The protocol is organized as a layered architecture rather than as an unconstrained sequence of processing steps. Its versioned control plane records data roles, episode rules, source-only transformations and model fitting, ordered detector candidates, seeds, budgets, outcome definitions, and integrity requirements. The core NF-CSE-CIC-IDS2018-v3 and external NF-UNSW-NB15-v3 datasets instantiate separate profiles and are never pooled. Within either profile, Gate 1 emits a constructed event and Gate 2 emits the audited score trace q_e . Gate 3 screens the assigned detector candidates on a pre-change guard-selection segment. For an admissible pair, $A(e, h) = 1$ permits the central Gate 4 replay workflow: static prediction, budgeted label acquisition, prediction-before-update adaptation, and conditional outcomes. The external profile also retains the accepted MAD, ADWIN, and Page-Hinkley monitoring branch for detector-delay analysis. Thus, “core evaluation” in Fig. 1 denotes the central evaluation workflow, not a restriction to the core dataset. A fail-closed controller retains every rejected or excluded branch in the versioned evidence record while leaving dependent outcomes $\perp$.

The core protocol was developed on DDoS and frozen before it was applied prospectively to DoS, Web Attacks, and Bot. Infiltration provided a secondary extension for testing temporal feasibility. Seed 11 was designated as the authoritative model seed in the frozen protocol before the prospective-replay outcomes were inspected; it was an arbitrary reproducibility anchor, not a claim that this seed was representative. Additional seeds described computational sensitivity but did not alter that primary decision. Table 2 retains the complete denominator for the core study.

DDoS and DoS are separate dataset class labels, not interchangeable names for one episode. DDoS supplied protocol-development evidence; the frozen decisions were then applied to the distinct DoS episode without using its post-change outcomes for redesign.

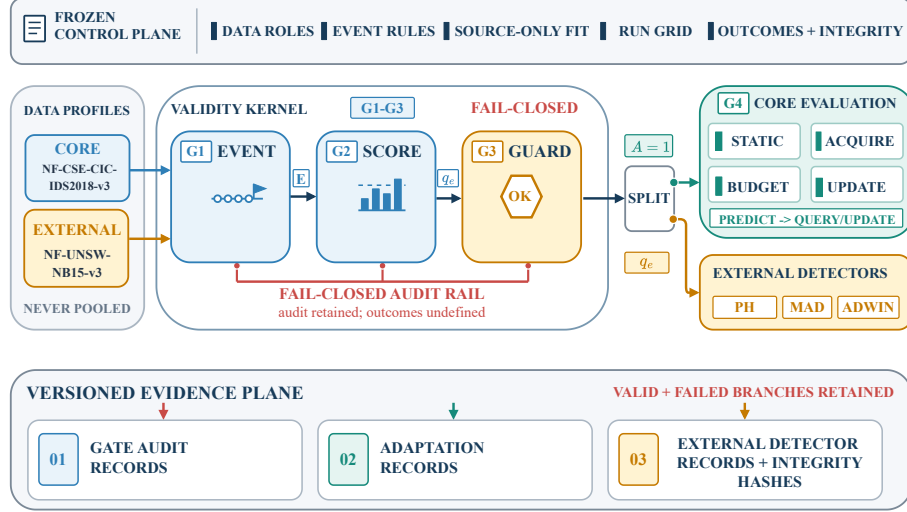


Fig. 1 Evidence-aligned four-gate architecture of the RAIDS-NIDS prospective-replay protocol. Separate, never-pooled data profiles pass through event construction (G1), source-only score auditing (G2), and guard-based operating-point selection (G3). An admissible pair, $A(e, h) = 1$, may enter the central Gate 4 static and adaptive replay; “core evaluation” in the diagram names that workflow rather than the source dataset. The external detector branch retains PH, MAD, and ADWIN delays on the audited score trace q_e . Gate 3 is not an independent safety test. Failed and excluded branches retain audit records while dependent outcomes remain undefined.

Table 2 Core-study episode roles and authoritative outcomes. Later stages are undefined after a failed gate.

Episode	Protocol role	Novel share	Gate 1: event	Gate 3: guard	Gate 4 executed
DDoS	Development	63.215%	Passed	Passed	Yes
DoS	Prospective	26.368%	Passed	Passed	Yes
Web Attacks	Prospective	0.102%	Passed	Failed	No
Bot	Prospective	3.536%	Passed	Failed	No
Infiltration	Secondary	1.765%	Failed continuity	Not reached	No

The external study followed a separate version history. Protocol v0.19 froze the order of the three families before their outcomes were known. After no episode proved eligible, v0.20 amended construction. Protocol v0.21 corrected score scaling after a development-only numerical diagnostic. This correction did not change the core evidence. Protocol v0.22 then prespecified and executed the Exploits Gate 4 extension. Protocol v0.23 froze the Reconnaissance Gate 4 grid and three primary contrasts after v0.22 outcomes were known but before Reconnaissance outcomes were examined. The

Table 3 Executed offline decisions and requirements for a future online management plane. The online column is a design extension, not an evaluated result.

Gate	Executed offline decision	Fail-closed action	Online management requirement
G1	Construct a labeled chronological episode and verify future support and continuity	Retain failure reason; define no later metric	Replace oracle construction with telemetry, analyst-defined events, and an observable episode policy
G2	Audit source-only transforms, score scales, data roles, and hashes	Reject the score and retain diagnostics	Keep the current static policy; alert the operator to pipeline or scale failure
G3	Select the most sensitive candidate with zero disallowed detector-specific trigger events on the guard-selection segment	Withhold Gate 4 for the branch	Freeze the setting before a separate acceptance period; report false alarms over time with uncertainty
G4	Replay static or adaptive prediction, immediate label return, and update	Withhold query, update, and dependent claims	Queue delayed or uncertain labels, require update approval, record completion time, and support rollback

Exploits extension is post-diagnostic, and its broad contrast set is descriptive. Reconnaissance is an outcome-free prespecified within-trace second episode, not independent replication.

Table 3 separates the executed offline controller from an operational extension. Gate 1 uses the known family identity, future support in the first 500 and 5,000 flows, and the availability of 100,000 subsequent rows. It is therefore a retrospective research-episode construction rule and cannot be evaluated by an operator at an unknown live onset.

3.2 Formal Gate Model, Stopping Algorithm, and Complexity

Equations (1)–(3) define when an episode–detector–configuration branch is admissible, but equations do not by themselves enforce the order of data access and computation. Algorithm 1 provides that enforcement. The central issue is execution integrity. A conventional experiment driver can continue after an infeasible event, reuse a numerically invalid score, recalibrate after a Gate 3 failure has been observed, or compute adaptive metrics for a branch that was never eligible for evaluation. Each path produces a number whose evidential status is unclear.

Algorithm 1 therefore acts as an offline evaluation controller, not as a new learning algorithm or a live deployment controller. It binds each gate to a concrete input artifact and permits the next artifact to be created only after the preceding gate passes. Gate 1 establishes a valid research event; Gate 2 establishes a leakage-free and numerically valid score trace; Gate 3 establishes an accepted operating point on the guard-selection segment; and Gate 4 dispatches the static, adaptive, or detector-comparison replay

branch. It also operationalizes the partial-output semantics: a failed branch returns an audit record and $\perp$, while a passing branch returns only the defined outcomes.

Let $e \in \mathcal{E}$ identify a prespecified episode, $h \in \mathcal{H}$ a change-detector family, and $a \in \mathcal{A}$ a static or adaptive configuration. The frozen event rule returns $C_e = 1$ only when episode e satisfies every chronological, prevalence, support, and continuity requirement. After source-only fitting, $q_e = (q_{e,1}, \dots, q_{e,W})$ denotes the resulting W -window score trace. The score audit returns $V_e = 1$ only when the trace is finite, its effective scales are at least $\varepsilon = 10^{-6}$, its data roles and hashes agree with the frozen manifest, and neither target labels nor post-change rows contributed to fitting, reference estimation, or calibration.

For detector family h , let Θ_h be its prespecified candidate set and let $N_{\text{sel}}(e, h, \theta)$ count disallowed detector-specific trigger events produced by candidate $\theta \in \Theta_h$ on the pre-change guard-selection segment. The three evaluation-gate indicators and the observed zero-event candidate set are

$$\begin{aligned} g_1(e) &= \mathbf{1}[C_e = 1], \\ g_2(e) &= \mathbf{1}[g_1(e) = 1 \wedge V_e = 1], \\ \Theta_{e,h}^{\text{acc}} &= \{\theta \in \Theta_h : g_2(e) = 1 \wedge N_{\text{sel}}(e, h, \theta) = 0\}, \\ g_3(e, h) &= \mathbf{1}[\Theta_{e,h}^{\text{acc}} \neq \emptyset]. \end{aligned} \tag{1}$$

Here, $\mathbf{1}[\cdot]$ is the indicator function, $\wedge$ denotes logical “and,” and $\emptyset$ denotes an empty set. The episode–detector pair is admissible only when

$$A(e, h) = g_1(e) g_2(e) g_3(e, h) = 1. \tag{2}$$

If the pair is admissible, the protocol selects the most sensitive candidate $\theta_{e,h}^*$ from $\Theta_{e,h}^{\text{acc}}$ according to the frozen order. This set records zero disallowed detector-specific trigger events on the inspected segment; it does not imply independent or long-term safety. Let $\mathbf{M}(e, a, h)$ contain the outcomes defined for branch a : detection status and delay for the detector branch, and normalized recovery area (NRA), macro-F1, novel-class recall, and forgetting for the adaptive branch. Gate 4 is the following partial output:

$$\mathbf{Y}(e, a, h) = \begin{cases} \mathbf{M}(e, a, h), & A(e, h) = 1, \\ \perp, & A(e, h) = 0, \end{cases} \tag{3}$$

where $\perp$ means *undefined*, not a numerical zero. An audit record is retained in both cases; only the downstream performance vector is withheld after failure. This distinction prevents failed construction or calibration from being averaged into classifier performance.

Algorithm 1. Leakage-Controlled Fail-Closed Prospective-Replay Evaluation Protocol

Input: chronological flows; prespecified episode set $\mathcal{E}$; frozen construction rules; source-only preprocessing and reference model; detector families $\mathcal{H}$ with ordered candidate sets Θ_h ; configurations $\mathcal{A}$; frozen seeds, acquisition points, and outcome definitions.

Output: a complete gate audit and either defined outcome vectors $\mathbf{Y}$ or $\perp$.

1. Freeze episode identities, data roles, candidate order, seeds, label budgets, selectors, update rules, outcome definitions, and integrity requirements before inspecting the named prospective-replay outcomes.
2. For each $e \in \mathcal{E}$, construct the chronological source and target event. If $C_e = 0$, record the Gate 1 reason, set every downstream branch to $\perp$, and stop that episode.
3. Fit all preprocessing and the reference model on source data only. Partition the pre-change target segment into disjoint reference, calibration, and guard-selection windows. Produce the score trace with each prediction recorded before any query or update.
4. Audit chronology, roles, hashes, finite scores, effective scales, and absence of target-label or post-change leakage. If $V_e = 0$, retain the diagnostic, set every downstream branch to $\perp$, and stop that episode.
5. For each $h \in \mathcal{H}$, evaluate its frozen candidates on calibration and guard-selection data. Form $\Theta_{e,h}^{\text{acc}}$ and select its most sensitive member. If the set is empty, record the Gate 3 failure and set that detector branch to $\perp$.
6. For each accepted branch, monitor post-change windows without refitting the detector and latch the first valid trigger. For a static configuration, do not query or update.
7. For an adaptive configuration, query only at the frozen acquisition point, record row indices, query-set hash, budget, and realized class yield, then update the model after labels arrive. Continue prediction-before-update evaluation.
8. Compute only support-eligible outcomes, store run-level artifacts and gate decisions, and return $\mathbf{Y}$ according to Eq. (3).

3.2.1 Computational Complexity and Feasibility

Let n_e be the number of target flows processed for episode e , d_e the embedding dimension, k_e the number of active class prototypes, W_e the number of score windows, and $R = \sum_{h \in \mathcal{H}} |\Theta_h|$ the total number of prespecified detector candidates. Let $\mathcal{A}_e^{\text{acc}}$ be the configurations that reach Gate 4, N_q the rows in the current 500-flow acquisition window, b the label budget, and $m = \min(N_q, 10b)$ the uncertainty-diversity candidate-set size. Thus, $N_q \leq 500$ in the executed design. After chronological ordering, event construction is $O(n_e)$; if sorting is required, it is $O(n_e \log n_e)$. For the expandable nearest-prototype model, source transformation and target prediction are dominated by $O(n_e k_e d_e)$ distance operations, while construction of the windowed score is $O(n_e d_e)$.

Candidate calibration and guard screening require $O(W_e R)$ detector-update calls in an upper-bound accounting, and monitoring the selected candidates adds $O(W_e |\mathcal{H}|)$. Random acquisition requires $O(N_q)$ time for the frozen permutation. The implemented uncertainty-diversity selector requires $O(N_q \log N_q)$ time for score ordering and $O(b m d_e)$ time for the incremental farthest-first distance updates. Prototype updating is $O(b d_e)$, and metric accumulation is linear in the evaluated target rows. Define $C_{\text{G4}}(e, a) = C_{\text{pred}}(e, a) + C_{\text{query}}(e, a) + C_{\text{update}}(e, a)$. Thus, omitting lower-order terms, the episode-level cost can be written as

$$T_e = C_{\text{fit}}(e) + O(n_e k_e d_e) + O(W_e R) + \sum_{a \in \mathcal{A}_e^{\text{acc}}} C_{\text{G4}}(e, a). \quad (4)$$

Table 4 Measured prototype-model resource results from the frozen evidence: median [minimum, maximum].

Episode	Fit (s)	Predict (s)	Flows/s	Update (ms)	Peak memory (MB)	Model (MB)
DoS	2.107 [1.914, 2.425]	1.002 [0.933, 1.513]	119,743 [79,329, 128,581]	3.727 [3.239, 12.077]	1,185.33 [894.80, 2,067.61]	0.523 [0.510, 0.557]
Exploits	2.226 [2.102, 3.677]	1.272 [1.175, 2.373]	94,323 [50,564, 102,110]	4.963 [4.190, 13.889]	1,299.21 [1,243.23, 1,332.91]	1.087 [1.084, 1.090]
Reconnaissance	2.308 [2.105, 3.674]	1.264 [1.187, 2.499]	94,944 [48,026, 101,079]	5.013 [4.059, 12.606]	1,316.15 [1,268.57, 1,332.76]	1.126 [1.124, 1.127]

The stored embedding and provenance arrays provide a conservative space bound of $O(n_e d_e + W_e R)$ per active episode, in addition to the bounded source reservoir and model state. Each constructed target stream contained 120,000 flows, the 500-flow windowing produced $W_e = 240$, and the external comparison used 13 candidate settings in total: four MAD multipliers, five ADWIN values, and four Page-Hinkley thresholds. The largest label budget was $b = 200$, so the executed acquisition-window bound was $m \leq 500$ and required at most 100,000 vector-distance updates.

The frozen run summaries also retained measured costs (Table 4). Values are medians with ranges over 90 runs per episode; update time is summarized over the 80 adaptive runs because static runs do not update. Prediction throughput is 120,000 divided by recorded target-prediction time, so it is a batch-processing measure rather than a latency percentile or guaranteed arrival-rate capacity.

The archived environment was Windows 11 (build 26200), Python 3.12.13, NumPy 2.5.1, pandas 2.3.3, scikit-learn 1.9.0, and SciPy 1.18.0. CPU, installed RAM, and GPU identifiers were not retained. Peak memory is process-level, query-selection time was not instrumented separately, and no latency percentile, audit-storage cost, or real-time packet/flow ingestion test was recorded. The measurements support reproducibility of the prototype runs but not an operational capacity claim. A failed gate still reduces avoidable computation by terminating dependent branches and reusing one score trace across detector candidates.

3.3 Baseline and Comparator Design

Table 5 lists the executed baselines and controlled alternatives. The static detector is the primary baseline for adaptive classification. Random acquisition is the sampling baseline for the targeted uncertainty-diversity selector. MAD is the fixed-threshold reference for the two streaming change detectors. In every comparison, the stream, source split, preprocessing contract, and relevant seed pairing remained fixed.

These experiments use claim-matched component baselines under one fixed, transparent detector. They are not a leaderboard comparison with state-of-the-art deep adaptive NIDS models. The comparisons test whether conditional adaptation, targeted

Table 5 Executed baseline and comparator experiments. B50 and B200 denote label budgets of 50 and 200.

Question	Baseline or reference	Comparator(s)	Controlled elements
Adaptation benefit	Static prototype detector; no labels or update	Eight adaptive factorial cells	Same event, source fit, model seeds, and prediction-before-update stream
Sample selection	Nested random acquisition	Uncertainty-diversity acquisition	Same budgets, acquisition point, update rules, and model seeds
Label budget	B50	B200	Exact budgets; Random B50 is the prefix of Random B200
Update mechanism	Naive prototype update	Source-anchored update	Identical queried rows within each selection-budget cell
Change detection	MAD threshold	ADWIN; Page-Hinkley	Same corrected score trace, calibration interval, guard interval, and post-change boundary

acquisition, label budget, prototype anchoring, and detector choice changed outcomes while the evidence path remained fixed. We make no claim that the prototype classifier or any one change detector is universally superior to current adaptive NIDS models.

Each comparator has a defined role. The static branch tests whether conditional adaptation changes performance. Random acquisition tests whether the prespecified targeted selector adds value. Budget and update-rule contrasts isolate label support and prototype retention, while the three detectors isolate response to a common score trace. Adaptive ensembles, semi-supervised continual learners, few-shot class-incremental models, and replay-based methods remain relevant model-level comparators [28–30, 34], but their published results cannot be used as quantitative baselines here. A fair extension would freeze a new protocol version and rerun selected methods through Gates 1–4 with the same source and target rows, label budgets, query times, guard criteria, and outcome definitions.

At least one structurally different learner must be executed before the gate logic can be called model-independent. ARF or SRP [38, 39] would provide a lightweight streaming comparison, while SPIDER, BFS-NID, TAMR, or SSF [28–30, 40] would test class-incremental or replay behavior. A valid extension must freeze each method’s Unknown mapping, expose only the same queried labels at the same acquisition point, preserve prediction-before-update order, and report matched runtime, memory, and predictive outcomes. Those experiments are outside the current evidence, so the present empirical claim is limited to a nearest-prototype proof of concept.

3.4 Gate 1: Temporal Event Construction

The core study used NF-CSE-CIC-IDS2018-v3, a NetFlow representation of CSE-CIC-IDS2018 that contains 53 extracted features and six attack types [41, 42]. A chunked read of the executed v0.18 source file observed 20,115,529 data rows, with the header excluded. We mapped labels to Benign, Brute-Force, Bot, DoS, DDoS, Infiltration, and Web Attacks. For every episode, the emerging family was excluded entirely from source training.

A deterministic, class-aware priority reservoir was drawn strictly before the target warm-up. For each eligible source class, the allocator first reserved up to 500 rows, subject to availability. It distributed the remaining capacity toward a total cap of 500,000 in proportion to each class’s residual historical capacity, assigned deterministic pseudo-random priorities using the frozen seed plus a class-specific offset, and retained the lowest-priority rows up to each quota. The selected source rows were then restored to chronological order. The target consisted of 20,000 consecutive pre-change flows followed by 100,000 rows that began at the first admissible onset of the emerging family. Construction failed if the held-out family appeared in the warm-up, if the warm-up violated its prescribed composition, or if the interval crossed a timestamp gap longer than 24 h. We hashed every retained artifact and its boundaries. Infiltration failed because its warm-up crossed a gap of 85.498 h.

The external study used NF-UNSW-NB15-v3, which has 53 NetFlow features and nine attack families [42, 43]. The executed temporal audit observed 2,365,424 rows after the header was excluded. Under the original v0.19 rule, a 20,000-flow warm-up had to be fully Benign; this rule constructed 0/3 episodes. The construction-only amendment in v0.20 allowed known non-Benign families during warm-up but excluded the held-out family from the immediately preceding 20,000 flows. Each warm-up family required at least 500 strictly earlier source rows. The warm-up and onset boundary could not cross a 24-hour gap, and 100,000 post-change rows had to remain. The held-out family also had to supply at least 5 of the first 500 and 50 of the first 5,000 post-change flows.

Row-count and integrity checks were completed before event construction. The NF-UNSW-NB15-v3 file matched SHA-256 4ebb97bd74412d566137d95a6fc3ffd8f374f1cf8cfe204d007848e7a668f9b5. The frozen v0.18 evidence retained the observed row count for NF-CSE-CIC-IDS2018-v3, but not the full raw-file SHA-256. No hash was reconstructed retrospectively. A fresh reproduction must record the CSE download hash and stop if either the source or observed count differs.

Table 6 records each external episode’s construction outcome and evidentiary role across protocol versions.

DoS was not replaced with another family. Every successful external artifact contained 500,000 source flows and a target stream of 120,000 flows. Table 7 quantifies the dependence that can be recovered from the frozen manifests. The target intervals started 14.513 s apart and the onsets were 95.044 s apart. The two source cutoffs differed by only 461 temporal positions. The Exploits target contained 1,342 Reconnaissance rows (1.118%), while the Reconnaissance target contained 2,965 Exploits

Table 6 External episode outcomes and roles retained across the v0.19–v0.23 evidence boundary.

Family	v0.19	v0.20 construction	Selected onset/support	Final role
DoS	Not constructed	Failed both onsets	3/500; 16/5,000	Retained failure
Exploits	Not constructed	Constructed	18 Feb 2015 01:04:57.146; 6/500; 139/5,000	v0.22 Gate 4 extension
Reconnaissance	Not constructed	Constructed	18 Feb 2015 01:06:32.190; 6/500; 55/5,000	v0.23 prespecified within-trace second episode

Table 7 Dependence between the two NF-UNSW target constructions recoverable from frozen event manifests.

Quantity	Exploits episode	Reconnaissance episode	Interpretation
Target start	00:53:32.935	00:53:47.448	Starts differ by 14.513 s
Constructed onset	01:04:57.146	01:06:32.190	Onsets differ by 95.044 s
Historical source cutoff position	1,165,130	1,165,591	Cutoffs differ by 461 positions
Other emerging family in full target	1,342 Reconnaissance	2,965 Exploits	1.118% and 2.471% of 120,000 rows
Other emerging family in warm-up	0 Reconnaissance	6 Exploits	Warm-up compositions differ
Other emerging family after onset	1,342 Reconnaissance	2,959 Exploits	1.342% and 2.959% of 100,000 rows

rows (2.471%). Six Exploits rows occurred in the Reconnaissance warm-up; no Reconnaissance row occurred in the Exploits warm-up. The corresponding post-change shares were 1.342% and 2.959%.

Exact target-row intersection, target Jaccard overlap, and source-reservoir row overlap cannot be reconstructed from the publication archives because cross-episode raw-row identifiers were not retained. They are therefore reported as unavailable rather than inferred from timestamps. The episodes are treated as dependent within-trace constructions, and an exact row-overlap audit remains a required check for a future data-backed protocol version.

3.5 Gate 2: Leakage Controls and Score Validity

All numeric and categorical imputers, one-hot vocabularies, scaling parameters, and optional principal components were fitted only on source training data. Target labels

were unavailable during preprocessing, reference estimation, threshold calibration, and guard-based operating-point selection. Each prediction was recorded before any query or model update. The audit trail retained queried row indices, class counts, and SHA-256 hashes for the query sets.

The reference model was an expandable nearest-prototype classifier. For source class c , its centroid was

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{x}_i. \quad (5)$$

A sample was assigned to its nearest centroid only when the distance did not exceed a frozen source-validation rejection quantile; otherwise, the model emitted *Unknown*. With $d_{\min}(\mathbf{x}) = \min_c \|\mathbf{x} - \boldsymbol{\mu}_c\|_2$, the continuous unknown score was

$$u(\mathbf{x}) = 1 - \exp[-d_{\min}(\mathbf{x})/s], \quad (6)$$

where s was the median nearest-prototype distance on source-validation data.

For the core study, let $\bar{\mathbf{z}}_w$ denote the mean D -dimensional embedding in 500-flow window w , and let $\mathbf{m}_{\text{ref}}$ and $\mathbf{s}_{\text{ref}}$ be the component-wise mean and standard deviation from the reference windows. The shift score was

$$q_w = \frac{1}{\sqrt{D}} \left\| \frac{\bar{\mathbf{z}}_w - \mathbf{m}_{\text{ref}}}{\mathbf{s}_{\text{ref}} + 10^{-6}} \right\|_2. \quad (7)$$

The external v0.20 development trace exposed a validity failure: `DNS_TTL_ANSWER` or `DNS_QUERY_TYPE` contributed 99.9787%–99.9996% of several scores because target-reference variance was nearly zero. Those detector outputs were retained as diagnostics and excluded from inference. For dimension j , v0.21 defined

$$s_j^{\text{ext}} = \max(\sigma_{\text{ref},j}, \sigma_{\text{source},j}, 10^{-6}), \quad (8)$$

and computed

$$q_w^{\text{ext}} = \sqrt{\frac{1}{D} \sum_{j=1}^D \left(\frac{\bar{z}_{w,j} - \mu_{\text{ref},j}}{s_j^{\text{ext}}} \right)^2}. \quad (9)$$

Here, $\sigma_{\text{source},j}$ is the population standard deviation (NumPy’s default, `ddof = 0`) of embedding dimension j over the source-training reservoir. The source anchor used source embedding values but did not use source labels, target labels, or post-change target rows. IP-address, flow-ID, timestamp, and label-like identifier fields were excluded from model input, while chronology was stored separately. Once identifiers and unavailable fields were removed, the external representation had 48 dimensions. One model fit generated one score trace for each episode and seed, and all three change detectors used that same trace. A score exceedance is a distributional-change alarm; it is not an attack alarm and does not identify an attack family.

3.6 Gate 3: Calibration and Guard-Based Operating-Point Selection

Each 20,000-flow pre-change segment was split into 40 windows of 500 flows. Windows 0–9 supplied the reference, windows 10–29 calibrated candidate parameters, and windows 30–39 formed the guard-selection segment. Monitoring started at window 40. The MAD candidates for the core calibration scores were

$$T_k = \text{median}(q_{\text{cal}}) + k(1.4826) \text{MAD}(q_{\text{cal}}), \quad k \in \{3, 4, 5, 6\}. \quad (10)$$

A MAD trigger required two consecutive exceedances. We selected the smallest candidate that produced zero persistent MAD triggers on the guard-selection segment. If no candidate passed, the episode failed closed. This operating point was frozen before any post-change evaluation. The segment was disjoint from reference estimation, calibration, and post-change monitoring, but all four MAD candidates were inspected on it. It therefore supplies an observed segment-specific selection criterion, not independent evidence of safety.

For the external comparison, the calibration scores underwent robust normalization:

$$r_w = \text{clip} \left[\frac{q_w^{\text{ext}} - \text{median}(q_{\text{cal}})}{1.4826 \text{MAD}(q_{\text{cal}})}, -8, 8 \right]. \quad (11)$$

MAD tested multipliers 3, 4, 5, and 6 with two consecutive exceedances. ADWIN tested $\delta \in \{0.1, 0.05, 0.01, 0.005, 0.002\}$, and Page-Hinkley tested thresholds 5, 10, 20, and 50 with $\delta = 0.005$ and upward-change mode. ADWIN and Page-Hinkley used River 0.25.0 [21, 22]; the public lockfile and run environments record that version, and the release provenance resolves to source tag 0.25.0 and commit 1a78c5c0bce3. For each detector, the most sensitive prespecified candidate with zero disallowed detector-specific trigger events on the ten inspected windows was selected, and a one-shot latch retained the first monitoring trigger. Repeated inspection raises a selection effect, and the unequal candidate-grid sizes prevent a matched false-alarm-rate comparison. The observed zero-event result was not converted to alarms per flow or hour because ten temporally adjacent windows do not supply an independent-rate sample.

Seed 11 was reserved for the corrected external score audit. The v0.21 detector comparison used seeds 22, 33, 44, 55, 66, 77, 88, 99, 110, and 121. Recorded outcomes were guard status, post-change detection, selected candidate, and delay from the true-change boundary. Before each later external Gate 4 experiment, Gates 2 and 3 were rerun with 32 source-fitted principal components and a memory limit of 1,000 rows. The Gate 4 model seeds were 11, 23, 37, 53, 71, 83, 97, 109, 127, and 149. All ten MAD guard runs passed for each external episode. We did not treat computational seeds as independent networks, sites, or attack episodes.

3.7 Gate 4: Conditional Adaptation and Factorial Ablation Design

DoS was the only prospective core episode whose authoritative branch passed the guard-selection criterion. The same conditional Gate 4 grid was later executed for

the admissible Exploits and Reconnaissance episodes under their own frozen protocol versions. Post-change predictions were summarized in 5,000-flow evaluation blocks. A $2 \times 2 \times 2$ factorial compared nested random and uncertainty-diversity acquisition, budgets of 50 and 200 labels, and naive and source-anchored prototype updates. For each executed episode, ten model seeds produced 80 adaptive runs, and ten static runs formed the no-update baseline. The main factorial fixed query seed 11 and paired identical selected rows across update rules.

Random acquisition used one seeded permutation, with B50 nested exactly in B200. The uncertainty-diversity (UD) rule was a manuscript-specific heuristic motivated by uncertainty sampling and core-set coverage [31, 32]; it was not the algorithm of Sener and Savarese. For budget b and N available rows, UD formed the candidate set $\mathcal{C}_b$ from the $m = \min(N, 10b)$ largest unknown scores. For $i \in \mathcal{C}_b$, score priority was

$$p_i = \begin{cases} \frac{u_i - \min_{j \in \mathcal{C}_b} u_j}{\max_{j \in \mathcal{C}_b} u_j - \min_{j \in \mathcal{C}_b} u_j}, & \max_{j \in \mathcal{C}_b} u_j > \min_{j \in \mathcal{C}_b} u_j, \\ 1, & \text{otherwise.} \end{cases} \quad (12)$$

Starting from $S_1 = \{\arg \max_{i \in \mathcal{C}_b} p_i\}$, the next row maximized

$$i_{t+1} = \arg \max_{i \in \mathcal{C}_b \setminus S_t} \left[\min_{j \in S_t} \|\mathbf{z}_i - \mathbf{z}_j\|_2 (0.5 + 0.5p_i) \right], \quad (13)$$

where juxtaposition denotes multiplication and $\mathbf{z}_i$ is the source-preprocessed embedding.

Algorithm 2. Prespecified Uncertainty-Diversity Acquisition

Input: unknown scores u_i ; embeddings $\mathbf{z}_i$; budget b ; frozen query seed q ; acquisition-window index w .

Output: sorted queried row indices and their SHA-256 query-set hash.

1. Set $m = \min(N, 10b)$ for the current 500-flow acquisition window and retain the m rows with largest u_i .
2. Min-max normalize their scores to $p_i \in [0, 1]$; if all candidate scores are equal, set every $p_i = 1$.
3. Initialize S with the candidate of largest p_i .
4. Until $|S| = b$, compute each unselected candidate’s minimum Euclidean distance to S , multiply it by $0.5 + 0.5p_i$, and append the maximizing candidate.
5. Sort the selected original row indices and store the indices, class yield, and query-set hash. Unknown-score ties inherit the frozen NumPy **argsort** order; selection-score ties use the first local **argmax** position.

For random acquisition, the selector seed was $10,000q + w$ and one permutation supplied both nested budgets. The naive rule, labelled **replay** in the public evidence files, updated a class prototype directly from acquired target examples. The source-anchored rule required at least five target examples and scaled the target contribution by $n/(n + 25)$, capped at 0.05 for known source classes.

This factorial provides the protocol’s component ablation study. Static operation removes label acquisition and updating. Random and uncertainty-diversity replace one another as the selection component; B50 and B200 vary the available label support;

and naive and source-anchored rules replace one another as the update component while the queried rows remain fixed. The validity gates are not removed. Bypassing a gate would invalidate the experimental path instead of isolating an adaptive component. The v0.22 Exploits grid was frozen after the v0.21 detector study. The v0.23 Reconnaissance grid preserved those settings and was frozen after v0.22 outcomes were known but before any v0.23 outcome directory existed.

3.8 Outcomes and Statistical Analysis

The primary recovery outcome was NRA. Let F_t be acquisition macro-F1 in support-eligible post-change block t , F_{ref} the pre-change reference, and $\mathcal{E}$ the set of eligible blocks:

$$\text{NRA} = \frac{1}{|\mathcal{E}|} \sum_{t \in \mathcal{E}} \text{clip}\left(\frac{F_t}{F_{\text{ref}}}, 0, 1\right). \quad (14)$$

Acquisition macro-F1 covered the true classes present in the target stream. An Unknown prediction counted as a false negative for the emerging class, and novel exact recall required an exact prediction of that class. Source forgetting was

$$\text{Forgetting}_t = \max(0, F_{\text{source,ref}} - F_{\text{source},t}). \quad (15)$$

The frozen support rule required at least one known class in an evaluation block, zero minimum rows per known class, and at least one non-Benign row. Macro-F1 used the true classes present in each eligible block, so its class set could change across blocks. Blocks that failed the rule were excluded from class-specific recovery calculations; they were not scored as zero. To test the effect of clipping, we also reconstructed

$$\text{NRA}_{\text{unclipped}} = \frac{1}{|\mathcal{E}|} \sum_{t \in \mathcal{E}} \frac{F_t}{F_{\text{ref}}}. \quad (16)$$

The frozen summaries support this unclipped analysis and exact eligible-block counts. They do not retain per-flow predictions needed to recompute alternative block sizes, fixed class sets, or a no-exclusion analysis. Those analyses require a new data-backed rerun rather than a post hoc approximation.

The exact estimand was expected performance over the frozen randomized training procedure conditional on one fixed stream and configuration. The factorial contrasts used ten seed-paired, within-stream differences as the resampling unit. For the paired percentile bootstrap, those ten differences were resampled with replacement 100,000 times. The 2.5th and 97.5th percentiles formed the interval, and the deterministic random seed was the integer encoded by the first eight hexadecimal digits of `SHA-256(metric|analysis-family|contrast)` [44]. Descriptive condition means in Table 9 used model seeds as clusters. The two adaptive update-rule observations were first averaged within each seed, after which the ten seed-level means were resampled with replacement. The static baseline contributed one observation per seed.

For paired differences d_i , we computed $d_z = \bar{d}/s_d$, where s_d is the sample standard deviation of the ten differences; d_z was reported as undefined when $s_d = 0$ [45]. The exact two-sided sign-flip test enumerated all $2^{10} = 1,024$ sign assignments and

compared the absolute mean difference; zero differences were retained, an all-zero contrast returned $p = 1$, and equality was counted in the tail [46]. Interpretation of the sign-flip results assumes exchangeability of the paired-difference signs under the null.

The analysis reported mean differences, paired d_z , percentile intervals, and positive, negative, and zero seed counts. Holm adjustment [47] was computed at three declared scopes: within each metric and analysis family (seven factorial effects, four simple effects, or four static comparisons), across all 15 contrasts for each metric, and as a sensitivity correction across all 60 metric-contrast tests. These quantities describe conditional algorithmic randomness on one fixed DoS stream. They do not quantify event-selection, temporal-sampling, attack, organization, or deployment uncertainty. The selection-by-budget interaction was $(\text{Random}_{200} - \text{Random}_{50}) - (\text{UD}_{200} - \text{UD}_{50})$.

The v0.22 Exploits extension used the same 15 contrasts for each of the four outcomes. Its 60 exact paired tests were adjusted at the frozen within-family, within-outcome, and all-test scopes, but the condition means and contrast estimates remain descriptive. The v0.23 Reconnaissance plan froze a separate three-test conditional computational family for NRA: UD minus Random, B200 minus B50, and UD B200 minus static. Exact paired sign-flip p values for these three contrasts were adjusted together by Holm’s method. The smallest attainable nonzero two-sided value with ten nonzero pairs is $2/2^{10} = 0.001953125$, so repeated values of 0.00195 record complete directional agreement rather than high-resolution evidence. Six additional v0.23 analyses were secondary. Results from v0.18, v0.22, and v0.23 were never combined into one multiplicity correction or treated as independent deployment samples.

The two robustness analyses were kept separate. Guard feasibility was recomputed for nine additional model seeds, while seed 11 remained authoritative. Random acquisition was then repeated over ten query seeds with model seed 11 fixed. These analyses measure different sources of uncertainty and were not pooled. The public repository contains full contrast matrices, query-seed tables, run summaries, configurations, and integrity records.

4 Results

4.1 Gate Outcomes Defined the Evaluation Denominator

At authoritative seed 11, DoS passed event construction and the guard-selection criterion, then proceeded to static and adaptive evaluation. Web Attacks and Bot produced valid events but had no accepted operating point. Infiltration failed temporal continuity before model fitting. Primary adaptive outcomes were therefore defined for one of the three prespecified prospective-replay episodes. Failed episodes remained in the denominator; they were not assigned zero accuracy or replaced.

Across ten model seeds, DoS met the Gate 3 criterion in 10/10 fits, Web Attacks in 8/10, and Bot in 0/10 (Table 8; Fig. 2). The secondary Web Attacks acceptances did not retroactively alter the authoritative seed-11 decision. They show computational sensitivity near the observed segment boundary. Bot had no zero-trigger candidate under any tested fit.

Table 8 Core guard-feasibility robustness. Pass counts describe computational seeds, not independent deployments.

Episode	Accepted seeds	Seed 11	Median zero-trigger candidates	Selected MAD among accepted seeds
DoS	10/10	Passed	4.0	MAD 3: 9; MAD 4: 1
Web Attacks	8/10	Failed	1.0	MAD 5: 3; MAD 6: 5
Bot	0/10	Failed	0.0	None

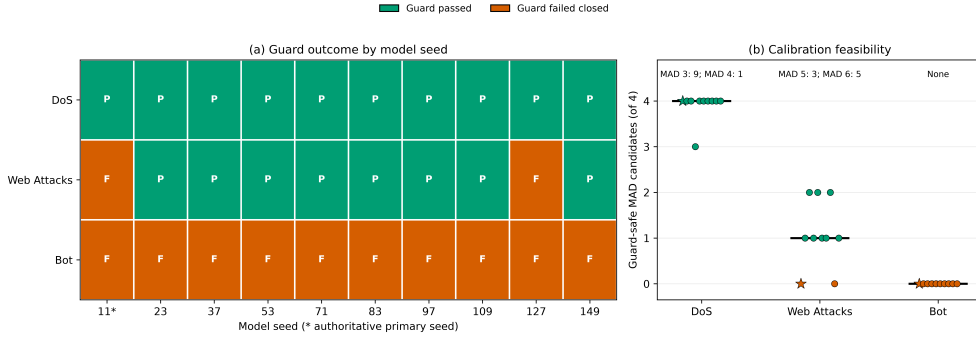


Fig. 2 Core guard-based operating-point feasibility across ten model seeds. The star marks seed 11, which the frozen protocol designated as authoritative before the prospective-replay outcomes; circles are secondary computational seeds and bars show medians. A Gate 3 acceptance requires at least one candidate with zero persistent MAD triggers on the ten guard-selection windows. Failure stops the replay branch; acceptance is not an independent safety test.

4.2 Factorial Ablation Against the Static Baseline

The ablation kept the event, reference model, preprocessing, stream, acquisition point, and model seeds fixed. Static operation removed the adaptive components. Within the factorial, one acquisition, budget, or update component was replaced at a time. The static detector retained its ability to recognize known classes but never identified the novel DoS family. Random B50 produced a large gain, whereas uncertainty-diversity (UD) B50 stayed near the static baseline. The two selectors converged at B200 (Table 9).

The unclipped sensitivity in Table 10 shows where Eq. (14) discarded performance above the pre-change reference. Clipping raised no external issue and did not change the DoS condition ranking. The DoS archive retained 12 eligible blocks per 20-block run; eight blocks per run failed the frozen support rule. Both external episodes retained all 20 blocks in every run.

Table 11 presents the direct component contrasts calculated from the frozen marginal means. Removing adaptation from Random B50 reduced recovery area by 0.494 and novel recall by 0.843. Replacing random acquisition with UD at B50 produced a similar loss. The larger budget had no material effect under Random, but

Table 9 DoS performance across ten model seeds: mean [95% bootstrap interval]. Adaptive values are marginal means over the two update rules.

Condition	Recovery area	Acquisition macro-F1	Novel exact recall	Source forgetting
Static	0.381 [0.374, 0.388]	0.398 [0.396, 0.400]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
Random B50	0.875 [0.872, 0.878]	0.799 [0.789, 0.809]	0.843 [0.842, 0.844]	0.016 [0.014, 0.018]
UD B50	0.372 [0.367, 0.378]	0.393 [0.392, 0.394]	0.017 [0.014, 0.021]	0.018 [0.015, 0.021]
Random B200	0.872 [0.871, 0.874]	0.794 [0.787, 0.803]	0.843 [0.842, 0.844]	0.019 [0.017, 0.021]
UD B200	0.872 [0.870, 0.874]	0.792 [0.786, 0.801]	0.836 [0.831, 0.840]	0.019 [0.016, 0.021]

Table 10 Frozen NRA clipping and support sensitivity. “Above reference” counts stored post-change blocks with $F_t/F_{\text{ref}} > 1$.

Episode	Condition	Clipped NRA	Unclipped NRA	Eligible blocks/run	Above reference
DoS	Static	0.380739	0.380739	12/20	0/200
DoS	Random B50	0.874736	0.883491	12/20	100/400
DoS	UD B50	0.371961	0.371961	12/20	0/400
DoS	Random B200	0.872238	0.879518	12/20	98/400
DoS	UD B200	0.872006	0.878167	12/20	89/400
Exploits	All five	No change by condition	Equal to clipped	20/20	0
Reconnaissance	All five	No change by condition	Equal to clipped	20/20	0

it produced a large gain under UD. On average, source-anchored updating changed every primary metric by less than 0.001.

The selection-by-budget interaction was -0.5025 $[-0.5098, -0.4955]$ for recovery, -0.4037 $[-0.4134, -0.3950]$ for macro-F1, and -0.8181 $[-0.8220, -0.8149]$ for novel recall. Every seed had the same direction. Exact sign-flip p values were 0.00195; Holm-adjusted values were 0.029 within each metric and 0.117 across all 60 tests. The large effect and consistent direction support the interaction, while the global multiplicity result does not support an unqualified family-wise significance claim.

Fig. 3 shows this selection-by-budget interaction across the paired model seeds.

4.3 Realized Label Yield Explained the B50 Difference

Random B50 acquired 49 DoS labels, which was 98% of its budget; UD B50 acquired 23, or 46%. At B200, Random acquired 188 DoS labels (94%) and UD acquired 172 (86%). For Random, the query seed produced identical row identities across model

Table 11 DoS component ablation contrasts from frozen marginal means. Each row reports the first condition minus the second. “Abs.” denotes an absolute mean difference.

Component test	Contrast	Δ NRA	Δ macro-F1	Δ novel recall	Δ forgetting
Remove adaptation	Static – Random B50	−0.494	−0.401	−0.843	−0.016
Substitute selector at B50	UD B50 – Random B50	−0.503	−0.406	−0.826	+0.002
Increase Random budget	Random B200 – Random B50	−0.003	−0.005	0.000	+0.003
Increase UD budget	UD B200 – UD B50	+0.500	+0.399	+0.819	+0.001
Substitute update rule	Anchored – naive	< 0.001 abs.	< 0.001 abs.	< 0.001 abs.	< 0.001 abs.

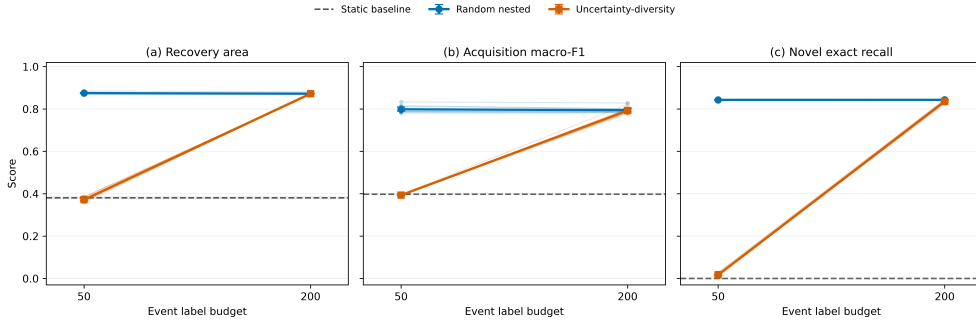


Fig. 3 DoS selection–budget interaction. Thin lines show paired model seeds; thick points and bars show means and 95% bootstrap intervals. The B50 deficit occurred under uncertainty-diversity, while Random had already saturated at 50 labels.

seeds. UD row identities varied because its uncertainty scores depended on the source-fitted model, although its class counts remained constant at 27 Benign/23 DoS for B50 and 28 Benign/172 DoS for B200. Update-rule comparisons still used identical queried rows within each model-seed, selection, and budget cell. The observed performance interaction tracked realized class support, but the design does not isolate label yield from query geometry or row identity (Fig. 4).

The negligible update-rule ablation did not rescue the low-yield UD B50 query set. In the separate query-seed analysis, Random B50 recovery averaged 0.8760 (SD 0.0076; range 0.8615–0.8878), and every query seed exceeded the static recovery value. Mean B200-minus-B50 differences were +0.0013 [−0.0028, 0.0062] for recovery, +0.0023 [−0.0054, 0.0121] for macro-F1, and +0.0060 [−0.0001, 0.0158] for novel recall. Thus, the B50 result was not tied to one favorable random query set, and the higher random budget gave no material gain.

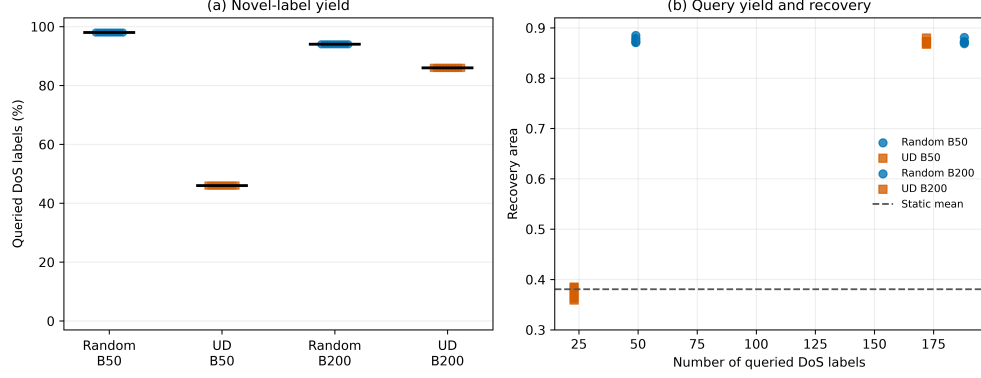


Fig. 4 Realized DoS labels under the two acquisition rules and budgets. Percentages are shares of the fixed query sets.

Table 12 Marginal external Gate 4 outcomes across ten model seeds. Adaptive outcomes are means over the naive and source-anchored update rules; label yield is mean [minimum, maximum].

Episode	Condition	NRA	Macro-F1	Novel recall	Forgetting	Novel labels
Exploits	Static	0.307392	0.148348	0.000000	0.000000	–
Exploits	Random B50	0.309248	0.149155	0.004109	0.000141	1.0 [1, 1]
Exploits	UD B50	0.324724	0.156051	0.041765	0.001150	5.2 [5, 6]
Exploits	Random B200	0.325855	0.156402	0.040937	0.000319	5.0 [5, 5]
Exploits	UD B200	0.363458	0.177414	0.196094	0.001034	12.7 [12, 13]
Reconnaissance	Static	0.306930	0.154921	0.000000	0.000000	–
Reconnaissance	Random B50	0.331942	0.163986	0.112351	0.003487	1.0 [1, 1]
Reconnaissance	UD B50	0.314512	0.157712	0.033705	0.001337	0.3 [0, 1]
Reconnaissance	Random B200	0.331781	0.164008	0.112351	0.003527	3.0 [3, 3]
Reconnaissance	UD B200	0.333097	0.164275	0.121243	0.003276	1.6 [1, 2]

4.4 External Gate 4 Extensions

Exploits and Reconnaissance passed event construction, the corrected source-anchored score audit, and the MAD guard-selection criterion for all ten Gate 4 model seeds. Each episode therefore produced 90 defined runs: ten static and 80 adaptive. Table 12 reports the static condition and the four selection–budget cells for all four outcomes. Each adaptive cell is averaged over the two update rules because their queried rows are paired within that cell.

Table 12 averages over update rules and therefore hides their episode-specific effect. Exploits mean NRA was 0.331334 for the naive update and 0.330308 for source

Table 13 Prespecified v0.23 Reconnaissance conditional computational NRA contrasts. Differences use the first condition minus the second; p values are Holm-adjusted across these three fixed-stream tests.

Contrast	Mean difference	95% paired bootstrap interval	Holm p
UD – Random	−0.0081	[−0.0129, −0.0033]	0.0176
B200 – B50	0.0092	[0.0044, 0.0139]	0.0117
UD B200 – Static	0.0262	[0.0181, 0.0334]	0.0059

anchoring. On Reconnaissance, the naive and source-anchored updates produced NRA 0.348738 versus 0.306928, macro-F1 0.170069 versus 0.154922, novel recall 0.189825 versus 0.000000, and forgetting 0.005625 versus 0.000188. Anchoring required five queried target examples per class, but the query sets contained only 0–3 Reconnaissance examples; no anchored run created that prototype. The naive update used smaller class samples when present, recovering some novel recall with greater source forgetting. This descriptive summary was outside the three v0.23 primary contrasts and is conditional on the fixed stream.

For Exploits, the mean budget effect (B200 minus B50) was 0.0277 [0.0245, 0.0307]. The mean selection effect (Random minus UD) was −0.0265 [−0.0298, −0.0230], and the selection–budget interaction was −0.0221 [−0.0260, −0.0180]. Each exact sign-flip test returned $p = 0.00195$. The Holm-adjusted values were 0.0137 within the factorial NRA family, 0.0293 across the 15 NRA contrasts, and 0.1172 across all 60 outcome–contrast tests. These estimates describe the frozen Exploits extension; the 60-test sensitivity correction does not support one global family-wise claim.

Reconnaissance had a separate three-contrast conditional computational family (Table 13). Marginal UD performance was lower than Random by 0.0081, while B200 exceeded B50 by 0.0092. The prespecified UD B200 condition exceeded static operation by 0.0262. All three Holm-adjusted exact p values were at most 0.0176. These values assess the frozen randomized training procedure on one fixed stream; they do not provide deployment-level significance. Table 14 reports all ten paired differences rather than relying on the p values alone. The six secondary v0.23 analyses remain outside this family.

The cross-episode direction is therefore conditional. UD exceeded Random on Exploits, whereas the prespecified Reconnaissance marginal contrast favored Random. Budget 200 improved average NRA in both episodes, but by different amounts. These results reject a general ranking of the acquisition rules and motivate episode-specific reporting of label yield and recovery (Fig. 5).

4.5 Alarm, Query, and Update Timing

Table 15 exposes the acquisition timeline for the authoritative DoS branch and the two external Gate 4 episodes. In every executed adaptive run, the MAD alarm was evaluated after predictions for the trigger window had been recorded. The query used rows from that 500-flow window, labels were assumed to return immediately and

Table 14 All seed-paired v0.23 Reconnaissance NRA differences in the frozen seed order.

Model seed	UD – Random	B200 – B50	UD B200 – Static
11	−0.000249	0.000035	0.005017
23	−0.021377	0.018027	0.036914
37	−0.016645	0.018030	0.038612
53	−0.013641	0.017476	0.038606
71	−0.012837	0.015579	0.032543
83	0.002472	−0.000416	0.026271
97	−0.005437	0.005492	0.008400
109	−0.004251	0.002784	0.010501
127	0.002250	0.000770	0.032117
149	−0.010856	0.014347	0.032691
Signs (+/ − /0)	2/8/0	9/1/0	10/0/0
Range	[−0.021377, 0.002472]	[−0.000416, 0.018030]	[0.005017, 0.038612]

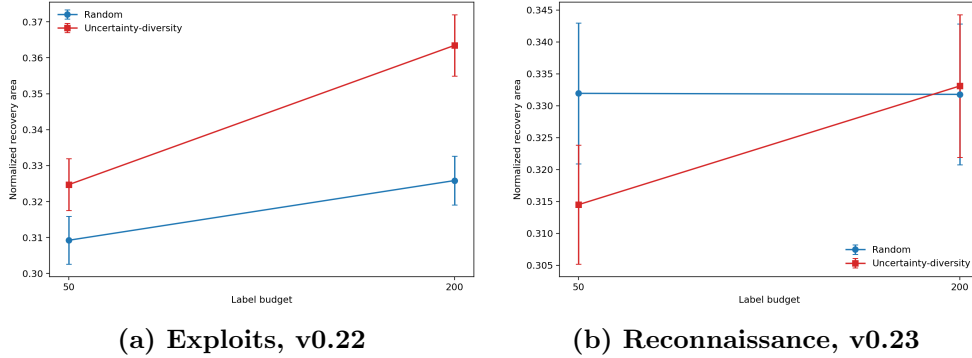


Fig. 5 Selection–budget interaction for the two external Gate 4 episodes. Points show means over the two update rules and ten model seeds; bars show ± 1 standard error across the ten seed-level update-rule marginal means. Both episodes came from the same recorded NF-UNSW trace.

without error, and the model update completed before the next window. No separate analyst-response or label-return timestamp was generated.

This mixed first block is retained in NRA and macro-F1 because the outcome measures the complete post-change trajectory, including time before adaptation. A delayed-label extension would queue queried row identifiers and preserve all intervening prediction-before-update outputs until labels arrived; the update would then occur once, before the next unprocessed window. Delays of 1, 5, or 10 windows were not executed, so no delayed-label performance number is reported.

4.6 External Detector Comparison on Shared Score Traces

The external evidence preserved the failed v0.19 construction, the v0.20 construction amendment, and the excluded v0.20 score diagnostics. After the source-anchored correction in v0.21, 14 of 48 Exploits dimensions and 15 of 48 Reconnaissance dimensions used the source scale because it exceeded the target-reference standard deviation.

Table 15 Executed MAD alarm-to-update timing. Delay is zero-based; decision flows count all processed post-change flows through the trigger window.

Episode	Change	MAD	Action window	Delay	Decision flows	First-block mix
DoS	40	3	41	1	1,000	1,000 pre-update; 4,000 post-update
Exploits	40	3	48	8	4,500	4,500 pre-update; 500 post-update
Reconnaissance	40	3	47	7	4,000	4,000 pre-update; 1,000 post-update

Table 16 External guard-selection outcomes and post-change delay across ten computational seeds per episode. Delay is zero-based; decision flows equal $(\text{delay} + 1) \times 500$.

Episode	Detector	Selected	Detected	Parameter	Median delay (range)	Median decision flows
Exploits	Page-Hinkley	10/10	10/10	Threshold 5	0 (0–0)	500
Exploits	MAD	10/10	10/10	Multiplier 3	8 (8–8)	4,500
Exploits	ADWIN	10/10	10/10	$\delta = 0.1$	93 (92–93)	47,000
Reconnaissance	Page-Hinkley	10/10	10/10	Threshold 10	1 (1–7)	1,000
Reconnaissance	MAD	10/10	10/10	Multiplier 3	7 (7–7)	4,000
Reconnaissance	ADWIN	10/10	10/10	$\delta = 0.1$	92 (91–94)	46,500

The minimum effective scale was approximately 1.00. This removed the near-zero denominator that had invalidated the development trace.

All 20 score-generation runs passed chronology, preprocessing, score-scaling, role, and integrity checks. The aggregate contained 60 episode–seed–detector records with no duplicate or development-seed row. Every selected operating point produced zero disallowed detector-specific trigger events on the guard-selection segment and later detected a post-change shift. MAD selected multiplier 3 and ADWIN selected $\delta = 0.1$ in every run; Page-Hinkley selected threshold 5 for Exploits and 10 for Reconnaissance. These are selection-dependent fixed-trace results, not independent safety passes.

For every Exploits seed, Page-Hinkley triggered after processing the first post-change window. Its median zero-based delay for Reconnaissance was one window. The MAD delays were eight and seven windows, while the ADWIN medians were 93 and 92 (Table 16; Fig. 6). Thus, an ADWIN median delay of 93 corresponds to a decision after 47,000 post-change flows, not 46,500. Page-Hinkley was faster than MAD in 19

Post-change detection delay across computational seeds

Circles: individual seeds Diamond: median Line: minimum to maximum Lower is faster

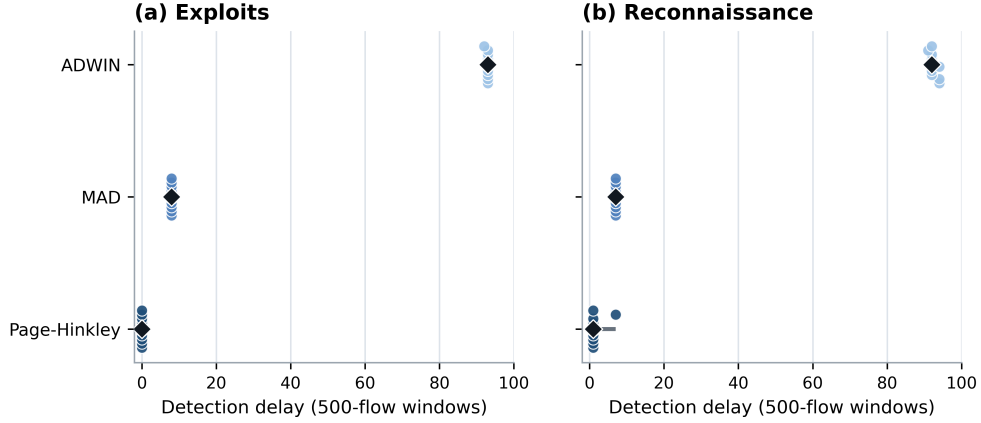


Fig. 6 External detection delay in 500-flow windows. Circles show computational seeds, diamonds show medians, and lines span the range. Delay 0 denotes a decision after processing the first post-change window, equivalent to 500 post-change flows.

of the 20 paired episode-seed comparisons and tied once. It was faster than ADWIN in all 20 comparisons, and MAD was faster than ADWIN in all 20.

This ordering applies to an unlabeled distributional score from two related controlled exposures. It does not show that a detector identified the attack family. The computational seeds also do not estimate deployment probabilities.

5 Discussion

5.1 Conditional Validity Preceded Adaptive Performance

The primary result occurred before adaptation. Under the frozen guard-selection rule, DoS had an accepted operating point, Web Attacks was sensitive to the model seed, and Bot had none. Infiltration stopped during temporal construction. If only DoS were reported, two failed prospective-replay decisions would disappear. Replacing the authoritative Web Attacks failure with a passing secondary seed would also revise the decision after its outcome was known. The four-gate protocol retains these cases and distinguishes an inadmissible evaluation branch from poor accuracy after admission.

The 8/10 count for Web Attacks is not an 80% probability of safe deployment. Seeds perturb a model fit within one fixed trace; they do not sample new organizations, dates, network policies, or attack campaigns. The count identifies sensitivity near the observed acceptance boundary. The segment also selected the candidate, so its zero-trigger result cannot verify the selected operating point independently. Operational NIDS assessments call for disclosure of the conditions under which a detector is used [4, 5]. Our audit record therefore retains those conditions and every gate outcome.

The external sequence exposed the same dependency at two earlier gates. The first event rule produced no eligible episode. The first amended score was then dominated

numerically by nearly constant features. Reporting the v0.20 detector outputs as ordinary comparisons would have concealed a score-validity failure. The source-anchored correction addressed that denominator problem without using target labels or post-change rows. Only after the corrected branches met the guard-selection criterion did Exploits and Reconnaissance enter Gate 4 under frozen extension protocols.

5.2 Interpretation of the Executed Comparators

The explicit static baseline attributes the DoS gains to conditional adaptation rather than to the original prototype detector. Random B50 increased recovery by approximately 0.49 over the static result and supplied 49 DoS labels. UD B50 supplied only 23 DoS labels and remained near static. At B200, UD acquired 172 DoS labels and reached the Random result. The composition of the query sets offers a direct explanation for the selection–budget interaction.

This result does not establish random sampling as a generally superior labeling strategy. DoS prevalence after the change was 26.368%, which made random sampling highly productive on this stream. Earlier NIDS studies motivate active selection for zero-day detection and under distribution shift [33, 34]. Whether a targeted selector becomes preferable when the emerging family is rare, dispersed, or close to known classes remains a hypothesis for a prespecified future study. The bounded conclusion is that an acquisition method should be judged by the labels it actually obtains, rather than by its nominal budget alone.

The two external Gate 4 episodes directly bound that conclusion. UD had the higher mean NRA on Exploits, but the prespecified Reconnaissance marginal contrast favored Random. Budget 200 improved mean recovery in both episodes, with a larger gain on Exploits. The selector reversal occurred within one recorded NF-UNSW trace, so it cannot be attributed to independent deployment environments. It instead shows that a selector ranking obtained on one admitted episode need not persist on a second admitted episode. A general acquisition rule would require independently prespecified networks, periods, and attack processes.

Source anchoring had a negligible effect on DoS and a small effect on Exploits, but its five-example threshold blocked the emerging-class prototype in every Reconnaissance run. Compared with the naive update, it reduced forgetting from 0.005625 to 0.000188, while novel recall fell from 0.189825 to zero and NRA from 0.348738 to 0.306928. This fixed-stream result records a stability–plasticity trade-off, not superiority of either rule. Repeated updates, representation learning, or several new families could produce a different result [27, 28, 30]. Label noise also remains a risk [34].

The external detector experiment addressed a separate baseline question. MAD, ADWIN, and Page-Hinkley used the same corrected score trace and underwent the same guard-based selection procedure. Page-Hinkley then declared change first, MAD responded later, and ADWIN was much slower. Because preprocessing, score construction, and stream boundaries were shared, this order records detector response on the two traces, not differences among classifier fits. The settings were selected from unequal grids and were not matched to a common false-alarm budget, so the result is limited to these two related onsets and does not establish universal detector superiority.

Table 17 Counterfactual reporting consequences recoverable from the frozen evidence.

Shortcut	Counterfactual report	Fail-closed record
Report only admitted core episodes	1/1 prospective episode appears usable	1/3 was admitted; Web Attacks and Bot remain in the denominator
Impute failed core branches as zero	Random B50 NRA becomes $0.875/3 = 0.292$	Failed outcomes are undefined, so 0.292 is not a classifier estimate
Report only constructed external episodes	2/2 constructions appear successful	2/3 were constructed; external DoS remains a retained failure
Use the v0.20 score without Gate 2	Detector outputs would be reported from the defective trace	Near-zero-variance dimensions contributed 99.9787%–99.9996% of several scores, so those outputs remain excluded
Bypass Gate 3 or tune after change	Web Attacks and Bot could be advanced without the recorded criterion	No valid downstream metrics exist; post-change threshold tuning was not executed

The comparison establishes conditional component effects, not state-of-the-art model dominance. The large Random-B50 gain, in particular, does not show that the prototype classifier is superior to recent adaptive ensembles or continual learners. For the one admissible DoS stream, it shows that a transparent model could recover the emerging family when the queried labels supplied enough class support. Model-level ranking requires every competing learner to pass through the same validity gates and receive the same information at the same time.

5.3 Counterfactual Reporting Audit

The frozen record permits a limited protocol-off audit without rerunning invalid branches (Table 17). This comparison asks how the reported conclusion would change if common reporting shortcuts were applied. It does not treat those shortcuts as valid baselines and does not invent performance for branches that never reached Gate 4.

The audit shows that the protocol changes eligibility, denominators, and the status of numerical outputs. It does not show that every conventional workflow would produce the same bias, and a randomized-split comparison would require a newly specified experiment.

5.4 Reporting and Reproducibility

The architecture implies a compact reporting contract for adaptive NIDS studies:

- (1) list every prespecified episode and its construction result;
- (2) state the source, reference, calibration, guard-selection, and monitoring boundaries;
- (3) audit the score definition and numerical scales before detector comparison;
- (4) report every candidate and the rule used to select an observed zero-trigger operating point;

- (5) keep downstream outcomes undefined after a failed gate; and
- (6) disclose exact label budgets, realized class yield, update timing, excluded diagnostics, and protocol amendments.

These fields reduce the risk that a study reports only successful episodes or silently corrects a threshold after post-change behavior is known. They also separate classifier benchmarking from the validity of the evaluation path. A complex classifier cannot compensate for target leakage, an invalid score, or a pre-change alarm rule that fails its own acceptance test.

Code, portable configurations, compact evidence, run-level summaries, omission inventories, archive provenance, and integrity records are publicly available at commit `8372049e7f29f1a78864573e6f0645d881c8ac92`, tagged `v0.23-figure5-seed-clustered-correction` [8]. `MANIFEST.sha256` verifies the repository tree. The `v0.22` evidence is frozen by tag `v0.22-unsw-exploits-evidence`; its complete archive has 114,039,766 bytes and SHA-256 `77c87be900e732fb64c505ac01b33b6dd8243f96845e7500317a200e4145c6ab`. The `v0.23` outcome-free plan is tagged `v0.23-unsw-recon-prespec`, and the completed evidence is tagged `v0.23-unsw-recon-evidence`. Its complete archive has 117,834,865 bytes and SHA-256 `41594744d24581d74c31624be24dbfec6013ed2b5fc0aaba4780934d1f37200b`. Raw datasets are not redistributed; the repository provides acquisition instructions and source identifiers.

5.5 Limitations and Threats to Validity

First, Gate 3 selected and accepted an operating point on the same 5,000-flow pre-change segment. The segment was disjoint from calibration and post-change monitoring, but it was inspected across several candidates. Zero disallowed detector-specific trigger events in ten adjacent windows therefore does not establish an independent safety test, a false-alarm rate, or long-term average run length. A stronger version must use calibration for ranking, a validation segment for selection, and a second untouched pre-change segment for final acceptance.

Second, the executed protocol governs offline evaluation. Gate 1 uses the known emerging-family identity, future support, and the later availability of 100,000 rows. Gate 4 assumes one immediate, correct label return and one update. A live controller would need an observable event policy, analyst and label queues, response-time and noise models, update approval, fallback operation, and rollback. None of those operational mechanisms was tested.

Third, the core factorial was executed for one admitted NF-CSE-CIC-IDS2018-v3 episode. The two NF-UNSW episodes share one trace, start 14.513 s apart, and have onsets 95.044 s apart. Their targets contain one another’s emerging family, and exact target-row and source-reservoir overlap cannot be recovered from the publication archives. The evidence therefore supports controlled fixed-episode case studies, not independent replication or cross-network validation.

Fourth, model and query seeds represent computational randomness within fixed streams. The seed-level bootstrap intervals, effect sizes, sign-flip tests, and Holm adjustments estimate the randomized training procedure conditional on those streams.

They do not sample networks, periods, attacks, or organizations. Prespecification protects against outcome-based analysis selection but does not change the inferential unit.

Fifth, NRA uses a permissive support rule, a block-specific true-class set, and clipping at one. The unclipped audit did not change the reported rankings, but the DoS analysis excluded eight of 20 blocks per run. Alternative block sizes, fixed-class macro-F1, results without support exclusion, and class-specific trajectories cannot be reconstructed from the frozen summaries because per-flow predictions were not retained.

Sixth, the expandable nearest-prototype model and combined uncertainty-diversity selector provide a proof of concept. No structurally different adaptive learner was executed, and the design cannot separate uncertainty from diversity or label yield from query geometry. The measured runtime and memory summaries omit query-selection time, latency percentiles, CPU and installed-RAM identifiers, audit-storage cost, and a real-time ingestion test.

Finally, the v0.18 archive retained the observed NF-CSE-CIC-IDS2018-v3 row count but not the complete raw-file SHA-256. The public repository is pinned by its commit and manifests but has no archival DOI. These gaps remain explicit. Future validation should freeze the entire construction, score, independent acceptance, delayed-label, and adaptation sequence before applying it to a new network or period.

6 Conclusion

The four questions are answered within a fixed-stream, offline prospective-replay boundary. For RQ1, DoS was the only core prospective episode admitted after construction, score audit, and guard-based operating-point selection. Web Attacks and Bot had no accepted authoritative operating point, and Infiltration failed temporal continuity. External DoS failed construction; Exploits and Reconnaissance reached Gate 4 after the corrected score audit and MAD selection procedure.

For RQ2, Random B50 increased DoS NRA from 0.381 to 0.875, macro-F1 from 0.398 to 0.799, and novel-class recall from 0.000 to 0.843. The external conditions showed smaller and episode-dependent changes across NRA, macro-F1, novel recall, forgetting, and label yield. These outcomes describe admitted fixed episodes, not expected performance in a future deployment.

For RQ3, the DoS selection–budget interaction tracked realized label yield, although yield was not isolated from query identity and geometry. UD exceeded Random on Exploits, while Random had the higher marginal NRA on Reconnaissance. Update-rule effects were negligible for DoS and small for Exploits. On Reconnaissance, source anchoring reduced forgetting but its five-label threshold blocked the emerging-class prototype and reduced novel recall to zero. This descriptive result was outside the three primary v0.23 contrasts. Random B50 exceeded static operation for all ten DoS query seeds (mean 0.8760, SD 0.0076; range 0.8615–0.8878). Both seed analyses describe fixed-stream computational sensitivity, not deployment uncertainty; neither supports a universal selector or update-rule ranking.

For RQ4, the selected Page-Hinkley settings responded earlier than MAD and ADWIN on both external score traces. Median zero-based delays were 0 and 1 windows for Page-Hinkley, 8 and 7 for MAD, and 93 and 92 for ADWIN. These settings were not matched to a common false-alarm budget, and a distributional alarm does not identify an attack family.

The supported contribution is an offline evaluation-control architecture with partial-outcome semantics. It prevents a later performance number from overriding failed event construction, an invalid score, or an unmet pre-change selection criterion. It does not establish independent pre-change safety, live deployment authorization, model independence, or population-level effects. Those claims require a new protocol version with independent acceptance data, another network or period, delayed and noisy labels, at least one structurally different adaptive learner, and complete operational timing and resource instrumentation.

Declarations

Acknowledgements. Not applicable.

Author contributions. The authors confirm their contributions to the paper as follows: conceptualization, Bilal Rafique, Keming Wang, Sania Kanwal, Yasir Rafique, and Zahid ur Rahman; methodology, Bilal Rafique, Sania Kanwal, and Yasir Rafique; software, Bilal Rafique; validation, Bilal Rafique, Keming Wang, Sania Kanwal, Yasir Rafique, and Zahid ur Rahman; formal analysis, Bilal Rafique and Keming Wang; investigation, Bilal Rafique, Sania Kanwal, Yasir Rafique, and Zahid ur Rahman; data curation, Bilal Rafique; visualization, Bilal Rafique; writing—original draft preparation, Bilal Rafique; writing—review and editing, Bilal Rafique, Keming Wang, Sania Kanwal, Yasir Rafique, and Zahid ur Rahman; supervision, Keming Wang; project administration, Bilal Rafique and Keming Wang. All authors reviewed the results and approved the final version of the manuscript.

Funding. This work was supported by the Sichuan Science and Technology Program under Grant 2026YFHZ0063.

Data availability. NF-CSE-CIC-IDS2018-v3 is publicly available from The University of Queensland under DOI 10.48610/ece9b83 [41]. NF-UNSW-NB15-v3 is available from the same institution under DOI 10.48610/6e0eda1 [43]. Code, portable configurations, compact result evidence, run maps, omission inventories, archive provenance, and integrity records are available at immutable GitHub commit 8372049e7f29f1a78864573e6f0645d881c8ac92 [8]. Its Figure 5 correction tag is `v0.23-figure5-seed-clustered-correction`. The frozen releases retain tags `v0.22-unsw-exploits-evidence` and `v0.23-unsw-recon-evidence`; the outcome-free `v0.23` plan retains `v0.23-unsw-recon-prespec`. Repository integrity is recorded in `MANIFEST.sha256`, with release-specific checksums and archive provenance beside each evidence set. Raw and derived dataset files are not redistributed. The complete NF-UNSW-NB15-v3 raw-file SHA-256 and acquisition checks are recorded in the repository. The historical NF-CSE-CIC-IDS2018-v3 raw-file SHA-256 was not retained and must be computed from a fresh official download for complete file-level verification.

Supplementary information. The public evidence releases contain machine-readable core and external performance summaries, paired contrasts, query-yield and query-provenance audits, guard records, publication figures, run-directory maps, omission inventories, and integrity manifests. They separately preserve the v0.19 failed construction, excluded v0.20 diagnostics, v0.21 score and detector audits, 90 defined v0.22 Gate 4 runs, and 90 defined v0.23 Gate 4 runs. Complete archives and checksum files are attached to the two tagged GitHub releases.

Ethics approval and consent to participate. Not applicable. The study involved no interaction with human participants and analyzed publicly released network-flow datasets. IP address, flow ID, timestamp, and label-like identifier fields were excluded from model input.

Consent for publication. Not applicable.

Competing interests. The authors declare no competing interests.

Abbreviations. The following abbreviations are used in this manuscript:

ADWIN	Adaptive Windowing
DDoS	Distributed denial of service
DoS	Denial of service
MAD	Median absolute deviation
NIDS	Network intrusion detection system
NRA	Normalized recovery area
PH	Page-Hinkley
SD	Standard deviation
UD	Uncertainty-diversity

References

- [1] Cantone, M., Marrocco, C., Bria, A.: Machine learning in network intrusion detection: A cross-dataset generalization study. *IEEE Access* **12**, 144489–144508 (2024). <https://doi.org/10.1109/ACCESS.2024.3472907>
- [2] Arp, D., Quring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium, pp. 3971–3988. USENIX Association, Berkeley, CA, USA (2022). <https://www.usenix.org/conference/usenixsecurity22/presentation/arp>
- [3] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., Cavallaro, L.: TESSERACT: Eliminating experimental bias in malware classification across space and time. In: 28th USENIX Security Symposium, pp. 729–746. USENIX Association, Berkeley, CA, USA (2019). <https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury>
- [4] Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316 (2010). <https://doi.org/10.1109/SP.2010.25>

- [5] Apruzzese, G., Laskov, P., Schneider, J.: SoK: Pragmatic assessment of machine learning for network intrusion detection. In: 2023 IEEE 8th European Symposium on Security and Privacy, pp. 592–614 (2023). <https://doi.org/10.1109/EuroSP57164.2023.00042>
- [6] Ceschin, F., Botacin, M., Bifet, A., Pfahringer, B., Oliveira, L.S., Gomes, H.M., Gregio, A.: Machine learning (in) security: A stream of problems. *Digital Threats: Research and Practice* **5**(1), 1–32 (2024) <https://doi.org/10.1145/3617897> . Article 9
- [7] Flood, R., Engelen, G., Aspinall, D., Desmet, L.: Bad design smells in benchmark NIDS datasets. In: 2024 IEEE 9th European Symposium on Security and Privacy, pp. 658–675 (2024). <https://doi.org/10.1109/EuroSP60621.2024.00042>
- [8] Rafique, B.: RAIDS-NIDS: Resilience-aware open-world network intrusion detection experiments. Software Figure 5 correction release, GitHub (2026). Commit 8372049e7f29f1a78864573e6f0645d881c8ac92; tag v0.23-figure5-seed-clustered-correction; integrity file MANIFEST.sha256; accessed 2 September 2026. <https://github.com/bilalrafiq/RAIDS-NIDS/tree/8372049e7f29f1a78864573e6f0645d881c8ac92>
- [9] Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications* **27**, 357–370 (2022) <https://doi.org/10.1007/s11036-021-01843-0>
- [10] Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Netflow datasets for machine learning-based network intrusion detection systems. In: *Big Data Technologies and Applications*, pp. 117–135. Springer, Cham, Switzerland (2021). https://doi.org/10.1007/978-3-030-72802-1_9
- [11] Corsini, A., Yang, S.J., Apruzzese, G.: On the evaluation of sequential machine learning for network intrusion detection. In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pp. 1–10. Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3465481.3470065>
- [12] Pederzoli, S., Paganelli, M., Contalbo, M.L., Benassi, R., Tiano, D., Iannucci, S., Guerra, F.: NOCTOWL: Adaptive tree-based model for network anomaly detection under delayed and sampled label availability. *IEEE Access* **13**, 197899–197911 (2025) <https://doi.org/10.1109/ACCESS.2025.3633419>
- [13] Husák, M., Manoj, D., Kumar, P.: Machine learning in intrusion detection: An operational perspective. In: 2024 20th International Conference on Network and Service Management, pp. 1–7 (2024). <https://doi.org/10.23919/CNSM62983.2024.10814637>
- [14] van der Hooft, J., Claeys, M., Bouten, N., Wauters, T., Schönwälder, J., Pras,

- A., Stiller, B., Charalambides, M., Badonnel, R., Serrat, J., Paula dos Santos, C.R., De Turck, F.: Updated taxonomy for the network and service management research field. *Journal of Network and Systems Management* **26**, 790–808 (2018) <https://doi.org/10.1007/s10922-017-9443-y>
- [15] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. *ACM Computing Surveys* **46**(4), Article 44, 1–37 (2014) <https://doi.org/10.1145/2523813>
- [16] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* **31**(12), 2346–2363 (2019) <https://doi.org/10.1109/TKDE.2018.2876857>
- [17] Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007). <https://doi.org/10.1137/1.9781611972771.42>
- [18] Page, E.S.: Continuous inspection schemes. *Biometrika* **41**(1–2), 100–115 (1954) <https://doi.org/10.1093/biomet/41.1-2.100>
- [19] Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. *Journal of the American Statistical Association* **88**(424), 1273–1283 (1993) <https://doi.org/10.1080/01621459.1993.10476408>
- [20] Moriano, P., Hespeler, S.C., Li, M., Mahbub, M.: Adaptive anomaly detection for identifying attacks in cyber-physical systems: A systematic literature review. *Artificial Intelligence Review* **58**(9), 283 (2025) <https://doi.org/10.1007/s10462-025-11292-w>
- [21] Montiel, J., Halford, M., Mastelini, S.M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H.M., Read, J., Abdessalem, T., Bifet, A.: River: Machine learning for streaming data in Python. *Journal of Machine Learning Research* **22**(110), 1–8 (2021)
- [22] River Development Team: River, version 0.25.0 [software]. Technical report, Python Package Index (2026). Source tag 0.25.0; source commit 1a78c5c0bce38c43ad1575e45c09f750d511b85d; source-distribution SHA-256 8756afe90122285a6fe06c5a38fba143bd15f85d8e84f65b661a5edec3a14e3c. <https://pypi.org/project/river/0.25.0/>
- [23] Geng, C., Huang, S.-J., Chen, S.: Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(10), 3614–3631 (2021) <https://doi.org/10.1109/TPAMI.2020.2981604>
- [24] Bendale, A., Boulton, T.E.: Towards open set deep networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572

- (2016). <https://doi.org/10.1109/CVPR.2016.173>
- [25] Yu, W., Chen, Z., Wang, H., Miao, Z., Zhong, D.: Industrial network intrusion detection in open-set scenarios. *International Journal of Information Security* **24**, 39 (2025) <https://doi.org/10.1007/s10207-024-00949-2>
 - [26] Yang, X., Tong, F., Jiang, F., Cheng, G.: A lightweight and dynamic open-set intrusion detection for industrial internet of things. *IEEE Transactions on Information Forensics and Security* **20**, 2930–2943 (2025) <https://doi.org/10.1109/TIFS.2025.3546849>
 - [27] Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. *Neural Networks* **113**, 54–71 (2019) <https://doi.org/10.1016/j.neunet.2019.01.012>
 - [28] Amalapuram, S.K., Reddy Tamma, B., Channappayya, S.S.: SPIDER: A semi-supervised continual learning-based network intrusion detection system. In: *IEEE INFOCOM 2024—IEEE Conference on Computer Communications*, pp. 571–580 (2024). <https://doi.org/10.1109/INFOCOM52122.2024.10621428>
 - [29] Du, L., Gu, Z., Wang, Y., Wang, L., Jia, Y.: A few-shot class-incremental learning method for network intrusion detection. *IEEE Transactions on Network and Service Management* **21**(2), 2389–2401 (2024) <https://doi.org/10.1109/TNSM.2023.3332284>
 - [30] Fathima A H, N., Khraisat, A., Ibrahim S P, S., Li, G.: Adaptive memory replay for network intrusion detection: Tackling data drift and catastrophic forgetting. *Computer Networks* **272**, 111712 (2025) <https://doi.org/10.1016/j.comnet.2025.111712>
 - [31] Settles, B.: Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA (2009). <https://minds.wisc.edu/items/37538f44-36ae-413e-8967-e6c831e17a8e>
 - [32] Sener, O., Savarese, S.: Active learning for convolutional neural networks: A core-set approach. In: *International Conference on Learning Representations*, pp. 1–13 (2018)
 - [33] Wu, Y., Hu, Y., Wang, J., Feng, M., Dong, A., Yang, Y.: An active learning framework using deep Q-network for zero-day attack detection. *Computers & Security* **139**, 103713 (2024) <https://doi.org/10.1016/j.cose.2024.103713>
 - [34] Alotaibi, F., Maffei, S.: Mateen: Adaptive ensemble learning for network anomaly detection. In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 215–234 (2024). <https://doi.org/10.1145/3678890.3678901>

- [35] Cevallos Moreno, J.F., Rizzardi, A., Sicari, S., Coen-Porisini, A.: SmartVille: A framework for realistic deep learning-based online network intrusion detection. *Journal of Network and Systems Management* **34**(4), 127 (2026) <https://doi.org/10.1007/s10922-026-10097-8>
- [36] Al Malwi, W., Asiri, F., Alturki, N., Alnazzawi, N., Kasimatis, D., Pitropakis, N.: Safety-aware reinforcement learning for self-healing intrusion detection in 5G-enabled IoT networks. *Computers, Materials & Continua* **87**(2), 88 (2026) <https://doi.org/10.32604/cmc.2026.074274>
- [37] Kuo, C.-W., Wu, C.-X.: A lightweight two-stage intrusion detection framework optimized for edge-based IoT environments. *Computers, Materials & Continua* **87**(3), 53 (2026) <https://doi.org/10.32604/cmc.2026.076767>
- [38] Gomes, H.M., Bifet, A., Read, J., Barddal, J.P., Enembreck, F., Pfahringer, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream classification. *Machine Learning* **106**, 1469–1495 (2017) <https://doi.org/10.1007/s10994-017-5642-8>
- [39] Gomes, H.M., Read, J., Bifet, A.: Streaming random patches for evolving data stream classification. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 240–249 (2019). <https://doi.org/10.1109/ICDM.2019.00034>
- [40] Zhang, X., Zhao, R., Jiang, Z., Chen, H., Ding, Y., Ngai, E.C.H., Yang, S.-H.: Continual learning with strategic selection and forgetting for network intrusion detection. In: IEEE INFOCOM 2025—IEEE Conference on Computer Communications, pp. 1–10 (2025). <https://doi.org/10.1109/INFOCOM55648.2025.11044615>
- [41] Luay, M., Layeghy, S., Sarhan, M., Hoseininoorbin, S., Moustafa, N., Portmann, M.: NF-CSE-CIC-IDS2018-v3 [dataset]. The University of Queensland, Brisbane, Australia (2025). <https://doi.org/10.48610/ece9b83>
- [42] Luay, M., Layeghy, S., Noorbin, N., Sarhan, M., Kulatilleke, G., Moustafa, N., Portmann, M.: Time matters: Temporal netflow features for ml-based network intrusion detection. *IEEE Access* **14**, 66899–66913 (2026) <https://doi.org/10.1109/ACCESS.2026.3688204>
- [43] Luay, M., Layeghy, S., Sarhan, M., Hoseininoorbin, S., Moustafa, N., Portmann, M.: NF-UNSW-NB15-v3 [dataset]. The University of Queensland, Brisbane, Australia (2025). <https://doi.org/10.48610/6e0eda1>
- [44] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall/CRC, New York, NY, USA (1993). <https://doi.org/10.1201/9780429246593>
- [45] Lakens, D.: Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Frontiers in Psychology* **4**, 863 (2013)

<https://doi.org/10.3389/fpsyg.2013.00863>

- [46] Ernst, M.D.: Permutation methods: A basis for exact inference. *Statistical Science* **19**(4), 676–685 (2004) <https://doi.org/10.1214/088342304000000396>
- [47] Holm, S.: A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* **6**(2), 65–70 (1979)