TITANIUM-4: A Low-Latency Four-Layer Intrusion Detection Pipeline and the Limits of Benchmark-Trained Thresholds under Domain Shift

Muhammad Younas*, Sadeeda, Hoor Ul Ain, Huzaifa Shahzad, Wasim Habib, Salman Ilahi Siddiqui, Ihsan Ul Haq, and Muhammad Farooq
Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan
*Correspondence: 22pwele5980@uetpeshawar.edu.pk

Citation | Younas. M, Sadeeda, Ain. H. U, Shahzad. H, Habib. W, Siddiqui. S. I, Haq. I. U, Farooq. M, "TITANIUM-4: A Low-Latency Four-Layer Intrusion Detection Pipeline and the Limits of Benchmark-Trained Thresholds under Domain Shift", IJIST, Vol. 8 Issue. 4 pp 1559-1582, July 2026
Received | June 04, 2026 **Revised |** July 03, 2026 **Accepted |** July 07, 2026 **Published |** July 11, 2026.

Firewalls enforce access rules within tight latency budgets but give no behavioral visibility; deep classifiers name attack families at a cost that inline placement cannot absorb. TITANIUM-4 addresses both with a four-layer pipeline: a calibrated LightGBM gatekeeper forwards only attack candidate flows to eight-class forensics, while benign flows bypass it and are processed by a One-Class SVM sentinel trained on normal traffic alone. On the cleaned CSE-CIC-IDS2018 benchmark, the gatekeeper attains F1 = 0.99987 and AUC = 0.99997 with zero false positives, routing 82.5% of test flows around the forensic classifier, which reaches macro F1 = 0.99 over eight families; the sentinel flags 45.06% of held-out Botnet and Infiltration flows at 5% FPR. Weighted end-to-end latency is 6.32 ms per flow, against 7.280 ms without early exit. Two findings qualify these results. The cleaned CSV carries no flow identifiers, so duplicates across splits cannot be excluded and the scores are a best-case estimate. Live DoS Hulk captures then expose domain shift: calibrated probabilities reach a maximum of 0.0138 against an operating point of $\tau_B = 1.0$, so nothing is labeled until τ_B is lowered by hand. Two captures processed at manually chosen thresholds of 1×10^{-8} and 1×10^{-4} label 87.3% and 52.2% of an identical attack, so the reported fraction is governed by an unconstrained parameter rather than by the traffic. The sentinel flags nothing in either run, its 95th-percentile rule having been computed over the flows it was applied to, though its live scores stay sharply bimodal, locating that failure in the thresholding rule rather than the model. Benchmark-derived thresholds do not transfer to CICFlowMeter output.

Keywords: Intrusion Detection System; LightGBM; One-Class SVM; CSE-CIC-IDS2018; Isotonic Calibration; Domain Shift; Threshold Transfer.

Introduction:

Network intrusion detection has grown harder as attack tooling has become cheaper to obtain and easier to vary, leaving static rule sets progressively less able to cover the space of observed behavior [1][2]. Machine-learned detectors are the standard response, and on public flow benchmarks they perform extremely well. That success is also the source of a persistent problem: scores obtained on a cleaned benchmark are routinely presented as evidence that a system is ready to run inline, and the two are not the same claim. Recent CSE-CIC-IDS2018 studies continue to report benchmark scores without live deployment tests [3][4][5].

Two operational constraints shape the design space. A firewall resolves a packet in microseconds but reports nothing about the nature of the packet. A deep classifier names the attack family but spends a per-flow budget that inline placement cannot absorb. Multi-stage designs narrow this gap by putting a cheap filter ahead of an expensive one [6][7], though in most such designs every flow still traverses every stage before a decision is issued.

TITANIUM-4 proceeds from the observation that the majority of traffic is benign, and that passing benign flows through a forensic classifier yields no additional information. It therefore departs from the sequential pattern at Layer 2: a flow classified as benign is routed directly to the anomaly sentinel and never enters the forensic classifier, so the majority of traffic follows a shorter path, while flows that warrant further analysis retain full classification depth. Section 4.5 costs the two paths separately and reports the resulting weighted mean. Figure 1 shows the four-layer architecture and the routing decision taken at each stage.

The benchmark results are reported in Section 5. Two independent checks, developed in Section 6 and Section 7.2, qualify how they should be read. The cleaned CSE-CIC-IDS2018 distribution provides no flow identifiers, so it cannot be established that flows from a single attack session are confined to one split; the near-perfect gatekeeper scores are therefore reported as a best-case estimate rather than a demonstrated generalization result. Separately, running the pre-trained pipeline unchanged against a live DoS Hulk capture produces a calibrated probability distribution compressed so far below the benchmark operating point that the threshold selected on the benchmark fails to detect anything at all.

Neither observation is specific to this architecture. The training data is a cleaned derivative of CSE-CIC-IDS2018 redistributed without flow identifiers, and the live data is the output of the extraction tool itself; any system fitted to the former and deployed against the latter inherits both the unverifiable split and the shift in feature distribution, irrespective of how its layers are arranged. The condition that produces the failure reported here, a benchmark preprocessed upstream of the modeler and an operating threshold selected on it, is the common case in the flow-based intrusion detection literature rather than an unusual one.

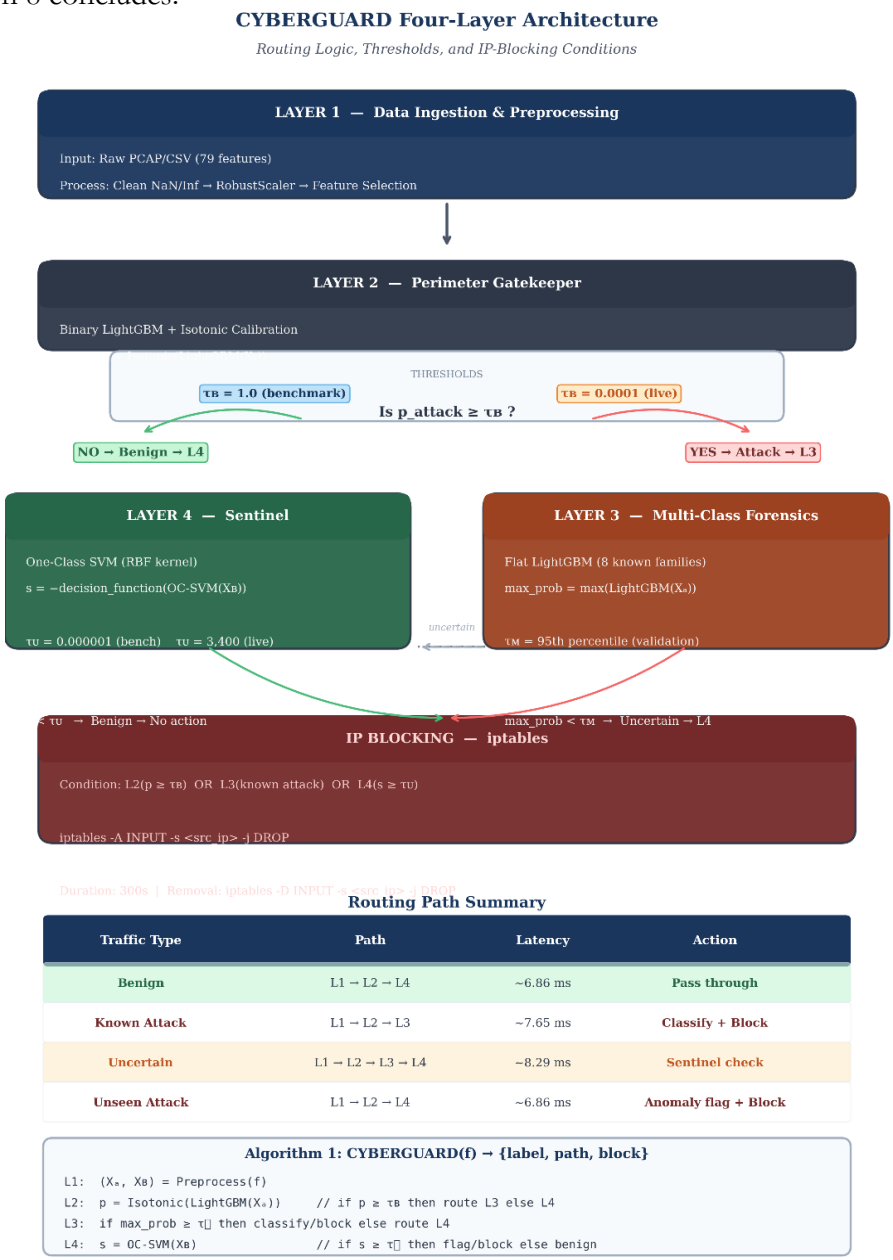
Layer 2 gatekeeper routes 17.5% of benchmark test flows to Layer 3 forensics and 82.5% to the Layer 4 sentinel. Layer 3 and Layer 4 are both terminals, so no flow traverses both. Per-layer latencies are reported in Section 4.5

The contributions of this work are as follows:

A four-layer early-exit pipeline that routes 82.5% of benchmark test flows away from forensic processing at Layer 2, with a weighted end-to-end latency of 6.32 ms measured in software; (ii) an isotonically calibrated gatekeeper reaching F1 = 0.99987 and AUC = 0.99997 with zero false positives, and a class-weighted flat forensics layer reaching macro F1 = 0.99 over eight families despite a near 12,000:1 class ratio; (iii) a label-free One-Class SVM sentinel detecting 45.06% of held-out Botnet and Infiltration flows at 5% FPR; (iv) a live DoS Hulk evaluation quantifying a four-order-of-magnitude threshold gap between benchmark and deployment, in which the sentinel flags nothing because its percentile rule must be computed over the flows it is applied to, locating the failure in the thresholding procedure rather than

the anomaly model; and (v) an explicit leakage analysis identifying the conditions under which the benchmark figures above should not be trusted.

Section 2 reviews related work. Section 3 describes the dataset, splits, and feature sets. Section 4 details the four layers and the latency profile. Section 5 reports benchmark results. Section 6 presents the live evaluation and domain-shift analysis. Section 7 states the limitations, and Section 8 concludes.



Routing Path Summary

Traffic Type	Path	Latency	Action
Benign	L1 → L2 → L4	~6.86 ms	Pass through
Known Attack	L1 → L2 → L3	~7.65 ms	Classify + Block
Uncertain	L1 → L2 → L3 → L4	~8.29 ms	Sentinel check
Unseen Attack	L1 → L2 → L4	~6.86 ms	Anomaly flag + Block

Algorithm 1: CYBERGUARD(f) → {label, path, block}

```
L1:  (Xa, XB) = Preprocess(f)
L2:  p = Isotonic(LightGBM(Xa))      // if p ≥ τB then route L3 else L4
L3:  if max_prob ≥ τ□ then classify/block else route L4
L4:  s = OC-SVM(XB)                  // if s ≥ τ□ then flag/block else benign
```

Figure 1. TITANIUM-4 four-layer architecture and routing logic. Flows enter Layer 1; the

Research Objectives:

This work pursues five explicit objectives:

To design and evaluate a calibrated binary gatekeeper that separates attack from benign traffic, and to assess it on both classification quality and per-flow latency, so that its cost as an inline filter is measured rather than assumed. (ii) To design a flat, class-weighted multiclass forensics layer that attributes attack flows to one of eight known families under an imbalance approaching 12,000:1, and to report its per-class and macro-averaged performance. (iii) To design a label-free One-Class SVM sentinel, trained on normal traffic alone and thresholded

at the 95th percentile of normal validation anomaly scores, and to evaluate its detection of held-out attack classes at a controlled false-positive rate. (iv) To design a two-path early-exit routing strategy in which benign-routed flows bypass forensic processing at Layer 2, and to measure the resulting weighted end-to-end latency directly against a no-early-exit baseline. (v) To quantify how thresholds selected on a cleaned benchmark behave when the frozen artefacts are deployed unchanged against live, non-benchmark traffic — the objective that distinguishes this work from prior multi-stage detectors.

Related Work:

Research Novelty:

TITANIUM-4 differs from prior multi-stage systems in three ways, not jointly present in the closest prior work. First, the pipeline performs an early exit at Layer 2: a flow classified as benign is routed directly to the anomaly sentinel and never enters the forensic classifier, and the per-path cost of that decision is measured directly rather than inferred. Conventional cascades pass every flow through every stage before issuing a decision. Second, the routing score is explicitly calibrated by isotonic regression, which places the decision on an interpretable probability scale; this is what makes the domain-shift failure legible as a measurable collapse of that scale rather than an opaque drop in accuracy. Third, and most importantly, the frozen artefacts are evaluated on a live, non-benchmark capture with the benchmark-selected threshold left unchanged, which quantifies threshold invalidation rather than the accuracy degradation typically reported in cross-dataset studies. Table 1 positions the work against its comparators; the distinguishing column is not accuracy but whether a live capture was run against the deployed artefacts with benchmark-selected thresholds.

Intrusion detection divides broadly into signature-based matching of known patterns and anomaly-based modelling of normal behaviour [1][2]. These two approaches fail in complementary ways: signatures miss what they have not catalogued, and anomaly models raise alarms on benign novelty, a difficulty compounded by the base-rate problem inherent to the task [8].

On tabular flow data, gradient-boosted trees remain the strongest practical baseline. [9] report near-perfect scores with a flat LightGBM on CSE-CIC-IDS2018 once class imbalance is addressed, and [10] reach comparable results with feature-importance analysis. Panigrahi and Borah [11] compare classifier families on this dataset and find tree-based methods consistently ahead of neural alternatives. LightGBM [12] is used here for both supervised layers on that basis, a finding reaffirmed in recent feature-selection studies on this dataset [13].

Multi-stage filtering to reduce inference cost is well established. [7] cascade decision-tree and random-forest stages for IoT traffic, and Al-Hawawreh and Sitnikova [6] describe a zero-trust funnel for unknown-attack detection; neither reports per-flow latency, which is the quantity that determines whether inline placement is feasible. [13] apply hierarchical federated learning to in-vehicle networks. TITANIUM-4 differs in that benign flows exit before the forensic stage rather than passing through it, and the per-layer cost of that decision is measured directly.

For unlabeled detection, One-Class SVM [14] remains a common baseline, and [15] combine autoencoders with one-class objectives. [16] and [17] derive anomaly thresholds from tail-distribution modelling. The sentinel here instead fixes the threshold at the 95th percentile of anomaly scores on the normal validation set. That rule imposes no distributional assumption on the tail, which is convenient, but Section 6.4 shows it to be a weaker choice under domain shift: a percentile is defined relative to whichever sample it is computed over, so when the sample changes the threshold moves with it. A tail-model approach of the kind used in [16] and [17] estimates a threshold from an assumed tail form and may transfer more stably. That comparison was not run here and is identified as future work in Section 7.4.

Class imbalance is treated with class-weighted objectives [9][10], resampling, or focal losses [18]; [19] pair SMOTE with LSTM. [20] and [21] address feature selection. Distribution shift between training and deployment receives markedly less attention. [22] observe accuracy falling from 96.1% to 88.4% across datasets and [23] report comparable cross-dataset degradation. The shift measured in Section 6 is of a different order: it is not a drop in accuracy at a fixed threshold but the invalidation of the threshold itself, with calibrated probabilities compressed four orders of magnitude below the benchmark operating point. Both [22] and [23] report degradation at a fixed operating point across datasets; neither reports that operating point ceasing to select any flow at all, and we are aware of no prior study quantifying threshold invalidation of this magnitude on flow data. Isotonic calibration [24][25] is what makes the failure legible, since it is the calibrated scale that collapses. Recent work has begun to address this directly through online shift detection and domain-adaptation approaches [26][27], and calibration-aware cross-dataset models [28].

Table 1 positions this work against the closest comparators. The distinguishing column is not accuracy but whether a live, non-benchmark capture was run against the deployed artefacts with the thresholds selected on the benchmark.

Table 1. Comparison with existing approaches. N/R = not reported.

Work	Dataset	Approach	Latency	Unseen-attack	Calibration	Live test
[9]	CSE-CIC-IDS2018	Flat LightGBM	N/R	No	No	No
[10]	Multiple	LightGBM (flat)	N/R	No	No	No
[11]	CSE-CIC-IDS2018	ML comparison	N/R	No	No	No
[7]	IoT/IIoT	Sequential DT-RF	N/R	No	No	No
[6]	Mixed	Zero-trust funnel	N/R	Yes	No	No
[20]	UNSW-NB15	Hierarchical ELM	N/R	No	No	No
[15]	Multiple	Autoencoder + OC-SVM	N/R	Yes	No	No
[21]	Multiple	Deep RNN + feature opt.	N/R	No	No	No
This work	CSE-CIC-IDS2018	4-layer LightGBM + OC-SVM	6.32 ms	Yes	Isotonic	Yes

Dataset and Preprocessing:

CSE-CIC-IDS2018 was produced by the Communications Security Establishment Canada and the Canadian Institute for Cybersecurity [29][30]. It carries 11 labels, one benign and ten attack labels, grouped into six broad families; DoS resolves into four labels and DDoS into three. CICFlowMeter extracts 79 bidirectional flow features per record, covering packet counts, interarrival statistics, and TCP flag counts.

The version used here is not the primary CIC release but rather a publicly redistributed cleaned derivative hosted on Kaggle, from which infinite and NaN records had been removed before publication, totalling 1,252,846 flows. This distinction bears on every result that follows. The models are fitted to a distribution that has already been filtered, whereas the live evaluation in Section 6 applies them to raw CICFlowMeter output, and the provenance of the training data is consequently one of the mechanisms examined in Section 6.2. The upstream cleaning is documented only in general terms by the redistributor and its exact operations cannot be reconstructed, which is a limitation of building on a cleaned derivative rather than the primary release.

Two of the ten attack labels, Botnet and Infiltration, are withheld from all training and validation and reserved for Layer 4 evaluation. They are genuine, catalogued families in the dataset, not newly discovered vulnerabilities; withholding them simulates encountering an attack class absent from training, and results on them are described throughout as held-out or unseen-attack detection rather than as true zero-day detection. The remaining eight attack labels plus benign form the nine classes used by Layers 2 and 3: Layer 2 treats the problem as binary, Layer 3 as an eight-way classification over attack flows only.

Known-class flows are split 70/15/15 by stratified random sampling on the binary label, preserving the benign-to-attack ratio across all three partitions (Table 2). Figure 2 shows the resulting distribution. The benign share is 82.5% in every split, and this figure recurs throughout the paper because it is what makes early exit worthwhile.

One property of the cleaned release constrains every benchmark number that follows: it contains no flow identifiers. Consequently, it cannot be verified whether duplicate or near-duplicate flows from a single attack session appear in more than one split. A stratified random split does not prevent this. The implications are developed in Section 7.2 and applies directly to the Layer 2 and Layer 3 results in Section 5.

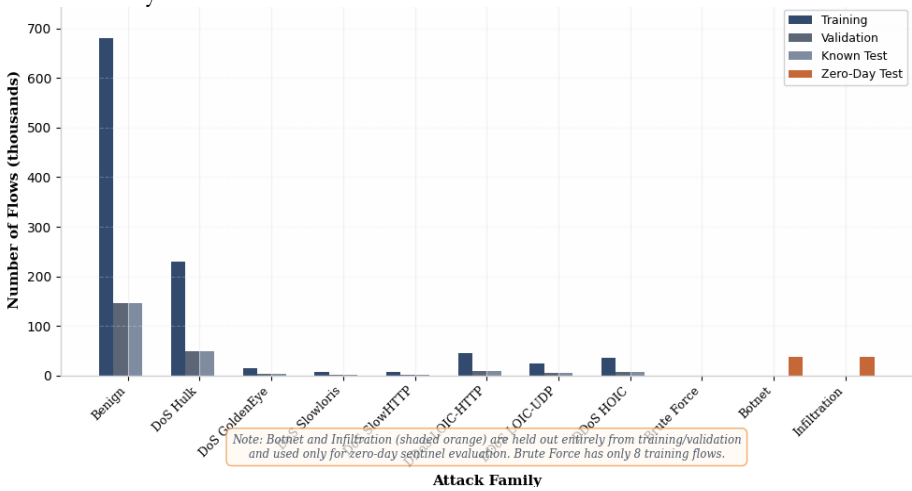


Figure 2. Class distribution across the stratified 70/15/15 split and the held-out set. Botnet and Infiltration are withheld entirely from training and validation and form the 76,026-flow held-out set used to evaluate Layer 4.

Table 2. Dataset split summary (70/15/15 stratified).

Split	Total Flows	Benign (%)	Attack (%)	Held-Out Classes Removed
Training	823,773	679,710 (82.5%)	144,063 (17.5%)	Botnet, Infiltration
Validation	176,523	145,653 (82.5%)	30,870 (17.5%)	Botnet, Infiltration
Known Test	176,524	145,653 (82.5%)	30,871 (17.5%)	Botnet, Infiltration
Held-Out Test	76,026	0 (0%)	76,026 (100%)	Held out entirely

Two disjoint feature sets are used (Table 3). Set A comprises the 40 features ranked highest by LightGBM gain on the binary objective and serves Layers 2 and 3, where the task is to separate known classes. Set B comprises 11 behavioural statistics (Fwd Pkt Len Mean, Bwd Pkt Len Mean, Flow Duration, Flow Byts/s, Flow Pkts/s, Flow IAT Std, SYN Flag Cnt, FIN Flag Cnt, Fwd Pkt Len Std, Pkt Len Var, and a derived SYN_FIN_ratio) and serves Layer 4 exclusively. Set B is not drawn from the gain ranking that produced Set A, on the reasoning that an unseen attack cannot be assumed to resemble a catalogued one in the features that discriminate catalogued families, though it may still perturb the distributional statistics of a flow. Keeping the sets disjoint also prevents the supervised layers from informing the unsupervised one. Set B is not attack-agnostic, and is not claimed to be: SYN Flag Cnt and

FIN Flag Cnt respond directly to connection-oriented flooding, and Flow Pkts/s responds to volumetric attacks of any kind. The weaker property claimed is that Set B was chosen for sensitivity to distributional deviation from normal traffic rather than being fitted to separate any labelled family, so its composition carries no information about which families were held out.

Both sets are scaled with RobustScaler fitted on the training partition only; the fitted parameters are persisted and reapplied unchanged to validation, test, held-out, and live data. This prevents preprocessing leakage and guarantees that live inference uses precisely the transform the models were trained under, a property that Section 6 shows to be necessary but not sufficient. Processed matrices are stored as Snappy-compressed Parquet. Loaded as float32 after scaling, peak memory remains below 4 GB on a 16 GB machine.

Table 3. Dual feature sets.

Feature Set	Count	Consumers	Selection Method
Set A	40	Layer 2, Layer 3	LightGBM gain (attack vs. benign)
Set B	11	Layer 4 only	Behavioural statistics + derived SYN_FIN_ratio

Methodology:

Algorithm 1: TITANIUM-4 Routing and Inference

The complete pipeline is summarised below. Each flow is preprocessed once at Layer 1, scored at Layer 2, and is processed by exactly one of Layer 3 or Layer 4; no flow traverses both terminal layers, which is the property that underlies the latency accounting in Section 4.5.

```
INPUT:  network flow record f
OUTPUT: routing path, label, and action
# Thresholds (all selected on validation, benchmark values shown):
#   τ_B = F1-optimal gatekeeper threshold          (benchmark 1.0)
#   τ_M = 95th pct of max class probability        (forensics)
#   τ_U = 95th pct of normal anomaly scores         (benchmark 1e-6)
# Layer 1 — ingestion and preprocessing
1  x_A ← scale(extract(f, SetA))      # 40 supervised features
2  x_B ← scale(extract(f, SetB))      # 11 behavioural features
# Layer 2 — calibrated gatekeeper
3  p ← isotonic( gatekeeper.margin(x_A) )   # calibrated P(attack)
4  if p ≥ τ_B then goto Layer 3       # attack-routed
5  else           goto Layer 4       # benign-routed (early exit)
# Layer 3 — multiclass forensics (terminal for attack-routed)
6  q ← forensics.predict_proba(x_A)
7  c ← argmax(q);  m ← max(q)
8  if m ≥ τ_M then  label ← c;  action ← block_if_top_source
9  else            label ← uncertain;  action ← report
10 return { path: L1→L2→L3, label, action }
# Layer 4 — unseen-attack sentinel (terminal for benign-routed)
11 s ← − ocsvm.decision_function(x_B)   # anomaly score
12 if s > τ_U then  label ← anomaly;  action ← alert
13 else             label ← benign;   action ← none
14 return { path: L1→L2→L4, label, action }
```

Layer 1: Ingestion and Feature Extraction:

Layer 1 loads the flow CSV, maps integer labels to family names, separates the held-out classes, applies the stratified split, and emits the two scaled feature matrices. All flows pass through it. Its median cost of 4.837 ms dominates the pipeline budget, and the cause is file reading and vectorised preprocessing rather than model inference. The figure is reported as

measured, including file input, because that is the cost actually incurred by the configuration evaluated here. It is not representative of a streaming deployment: an implementation reading from memory or a kernel-bypass path would not pay the disk term, and the pipeline budget would then be dominated by the model layers instead. No such implementation was built or measured, so the size of that reduction is not estimated.

Layer 2: Perimeter Gatekeeper:

The gatekeeper is a binary LightGBM classifier that decides, per flow, between benign and attack, and routes accordingly: flows at or above τ_B go to Layer 3 for forensic labelling, flows below τ_B go directly to Layer 4. This routing decision is the mechanism that keeps the mean latency low; without it every flow would pay the Layer 3 cost irrespective of whether it is an attack.

Because benign flows outnumber attacks in the training split, scale_pos_weight is set to the benign-to-attack ratio, 679,710 / 144,063 = 4.7181, so that the attack class is not overwhelmed. Early stopping on the validation split terminates training when validation performance ceases to improve.

Raw LightGBM margins are not probabilities. Isotonic regression, fitted on out-of-fold predictions from stratified five-fold cross-validation, maps them monotonically onto calibrated probabilities [24][25]. Isotonic regression is non-parametric and therefore assumes no particular functional form for the reliability curve, which suits boosted-tree outputs. The operating threshold τ_B is then chosen on the validation split alone, by maximising F1 over the calibrated probabilities; the test partition plays no part in its selection.

On the cleaned benchmark that search returns $\tau_B = 1.0$, and the routing rule is therefore $p \geq 1.0$. The value is not a rounding of 0.999 and not a placeholder. Isotonic regression is a monotone step function whose fitted steps take the empirical class frequency of each bin, so a bin containing only attack flows in the out-of-fold data is assigned exactly 1.0; 30,863 test flows fall on that step. Intermediate values are still produced for less separable flows, so the calibrator has not degenerated to hard 0/1 output. Two properties of this operating point matter later. It lies at the boundary of the probability range, leaving no margin above it; and F1 is flat over a wide interval below it, so the benchmark does not identify a unique threshold, as Section 5.1 reports. Section 6 shows what follows when the score distribution moves.

Calibration quality was not assessed. No reliability diagram was plotted and no Brier score computed, so the claim made here is the narrow one that isotonic regression was applied and that it produces a monotone mapping onto the unit interval; it is not a claim that the resulting probabilities are well calibrated in the sense of matching empirical frequencies. A high F1 and AUC constrain ranking, not calibration, and are consistent with poorly calibrated probabilities. The live results of Section 6.2 bear directly on this: the calibrated scale that appeared serviceable on the benchmark carried no usable information once the input distribution moved.

Layer 3: Multi-Class Attack Forensics:

Flows routed as attacks enter Layer 3 and are labelled into one of eight families: DoS Hulk, DoS GoldenEye, DoS Slowloris, DoS SlowHTTP, DDoS LOIC HTTP, DDoS LOIC UDP, DDoS HOIC, and Brute Force. The operational distinction matters: knowing a flow is DoS Hulk supports a targeted rule, whereas knowing only that it is an attack does not.

The classifier is a flat, class-weighted LightGBM over the 144,063 attack flows of the training split, using the Set A features and early stopping. A flat structure is used rather than a hierarchical one so that an error at an upper node cannot propagate down a branch; no hierarchical alternative was implemented, so no comparative claim about error propagation is made here. The imbalance across the eight families is extreme, with DoS Hulk dominant while Brute Force is very rare, at a ratio approaching 12,000:1, and without correction the model

collapses onto the majority family. Balanced class weights force each rare sample to carry proportionate influence.

A confidence threshold τ_M is defined at the 95th percentile of maximum class probability, with flows falling below it designated uncertain. In the benchmark evaluation every flow routed to Layer 3 exceeded τ_M, so no flow was marked uncertain, and the same held on the live capture. The pathway is therefore reported as untriggered rather than as a validated component, and no claim is made for it. It is retained in the description because it is present in the implementation and because a reader reproducing the system would otherwise find an unreported branch in the code; on the evidence available it is untested rather than useful. The current implementation reports uncertain flows rather than redirecting them to Layer 4, so even if triggered it would not alter routing.

Layer 4: Unseen-Attack Sentinel:

Layer 4 is terminal for benign-routed traffic. It exists because supervised layers cannot flag what they were never shown. A One-Class SVM [14] with an RBF kernel is trained on the 679,710 benign training flows using Set B features only, with no attack labels of any kind.

With nu = 0.01, which upper-bounds the training-error fraction and lower-bounds the support-vector fraction, the model retains approximately 6,797 support vectors, near 1% of the training set, and inference reduces to a kernel evaluation against those vectors followed by a comparison. The decision function is $f(x) = \text{sign}(w \cdot \varphi(x) - \varrho)$ with RBF kernel $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ and $\gamma = 1/(\text{n\_features} \cdot \text{X\_var})$. Anomaly scores are taken as the negative of the decision value, so larger scores indicate stronger evidence of anomaly.

The threshold τ_U is set to the 95th percentile of anomaly scores over the 145,653 normal validation flows, giving $\tau_U = 0.000001$ and fixing the false-positive rate at 5% by construction. This requires no assumption about the shape of the score tail. Section 6 demonstrates that this percentile rule, applied in a shifted domain, yields a threshold four orders of magnitude apart from the benchmark value.

System Integration and End-to-End Latency:

The layers form a sequential pipeline with a single early-exit point. Every flow is preprocessed at Layer 1 and scored at Layer 2. Flows below τ_B proceed to Layer 4 and terminate there; flows at or above τ_B proceed to Layer 3 and terminate there. Neither terminal layer returns flow to the other.

Latency was measured per layer on the development machine: Layer 1 over 5,000 raw rows including file reading and preprocessing, and Layers 2 to 4 over the full 176,524-flow known test set. Table 4 reports medians and 99th percentiles. Because each flow terminates at either Layer 3 or Layer 4 and never traverses both, the two paths are costed separately. From the medians, the benign path (L1 + L2 + L4) costs 4.837 + 0.958 + 0.413 = 6.208 ms and the attack path (L1 + L2 + L3) costs 4.837 + 0.958 + 1.072 = 6.867 ms. No flow in TITANIUM-4 traverses all four layers, so the arithmetic sum of 7.280 ms is not a path length of this pipeline; it is the per-flow cost of the sequential baseline defined below, in which the early exit is removed.

Weighting the post-ingestion branch costs by the observed routing split of Table 5 gives the end-to-end figure:

$$4.837 + (0.825 \times 1.371) + (0.175 \times 2.030) = 4.837 + 1.131 + 0.355 = 6.323 \text{ ms} \approx 6.32 \text{ ms}$$

where 1.371 ms is the benign branch (L2 + L4) and 2.030 ms the attack branch (L2 + L3). The weighted mean sits close to the benign path cost because 82.5% of flows take that route and never reach Layer 3. The relevant comparison is against a design without early exit, in which every flow is passed through the forensic classifier before anomaly screening and each flow costs L1 + L2 + L3 + L4 = 7.280 ms. Early exit therefore saves 0.96 ms per flow on this workload, a 13.1% reduction. Figure 3 shows the measured per-layer distribution of cost.

These timings were obtained in software on a general-purpose development machine. The pipeline was not evaluated on networking hardware, and no claim is made here regarding compliance with any hardware-switch timing budget.

Table 4. Per-layer latency profile (median and p99, ms). L1 measured on 5,000 raw rows including file reading and preprocessing; L2–L4 measured on the 176,524-flow known test set. The weighted end-to-end latency combines these medians with the Table 5 routing split.

Stage	Median (ms)	p99 (ms)	Sample Size
L1: Ingestion & Preprocessing	4.837	7.700	5,000 raw rows
L2: Gatekeeper (Binary)	0.958	3.886	176,524 flows
L3: Multi-Class Forensics	1.072	3.193	30,871 attack flows
L4: Sentinel (Unseen-Attack)	0.413	1.497	176,524 flows
Benign Path (L1→L2→L4)	6.208	—	82.5% of traffic
Attack Path (L1→L2→L3)	6.867	—	17.5% of traffic
Weighted End-to-End	6.32	—	Weighted average

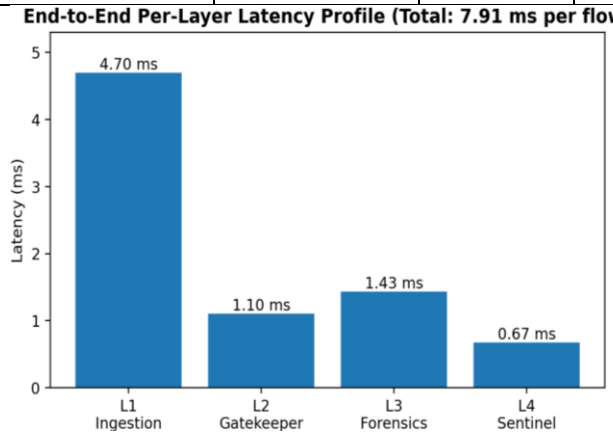


Figure 3. Per-layer latency profile. Layer 1 ingestion dominates the budget; the weighted end-to-end cost is 6.32 ms per flow.

Justification of Model Choices:

The three model families were selected for latency-bounded inference, and each choice is stated here against its alternatives. LightGBM is used for both supervised layers because gradient-boosted trees remain the strongest practical baseline on tabular flow data and consistently outperform neural alternatives on CSE-CIC-IDS2018 in the comparisons cited in Section 2, while its histogram-based inference fits a per-flow budget that a deep classifier does not. Isotonic regression is used for calibration in preference to Platt (sigmoid) scaling because it is non-parametric and imposes no functional form on the reliability curve, which suits the score distribution of a boosted-tree ensemble. Its cost is stated openly: it is a monotone step function defined only over the training score range, and Section 6.2 shows this to be one of the mechanisms by which the calibrated scale collapses under domain shift. The One-Class SVM is used for the label-free layer as an established, low-parameter boundary model whose inference reduces to a kernel evaluation against roughly 1% of the training set; deeper anomaly backbones such as Isolation Forest and autoencoders may raise the moderate held-out recall and are identified as the direct comparison for Section 7.4.

Benchmark Results:

Gatekeeper Performance:

Five-fold cross-validation yields AUC = 0.99997 (range 0.99996–0.99998) and F1 = 0.99987. The range across folds is the only dispersion estimate available: the pipeline was fitted once under a fixed random seed and not repeated across seeds, so no across-seed mean or standard deviation is reported. Given that the fold-to-fold spread is confined to the fifth decimal place, seed variance is unlikely to be the reason these scores are high; Section 7.2 sets

out the explanation considered more probable. Figure 4 plots F1 against the decision threshold. The curve is flat across a wide interval below its maximum, so on this data F1 does not distinguish between candidate thresholds over most of the range. The flatness is a property of the benchmark rather than a robustness guarantee: it means the benchmark supplies no evidence for one operating point over another, and the value selected at the boundary is the least conservative of the admissible set. Section 6 quantifies the consequence.

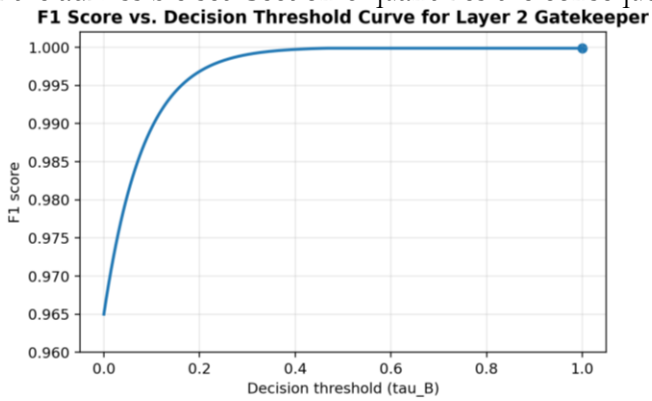


Figure 4. F1 score versus decision threshold for the Layer 2 gatekeeper. The maximum F1 = 0.99987 occurs at $\tau_B = 1.0$.

On the 176,524-flow test set the confusion matrix (Figure 5) gives 145,653 true negatives, 30,863 true positives, 8 false negatives, and 0 false positives, hence precision = 1.0000, recall = 0.99974, and FPR = 0.00000. The absence of false positives means no benign flow consumed forensic capacity at Layer 3. The eight missed attacks carry low packet counts and short durations, placing them close to benign traffic in Set A feature space**,** the type of flow that the sentinel is positioned to catch. The ROC curve is given in Figure 6, and the resulting routing split in Table 5.

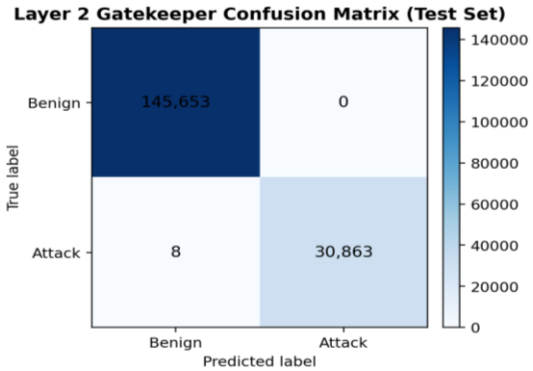


Figure 5. Layer 2 gatekeeper confusion matrix on the 176,524-flow test set. TN = 145,653; FP = 0; FN = 8; TP = 30,863

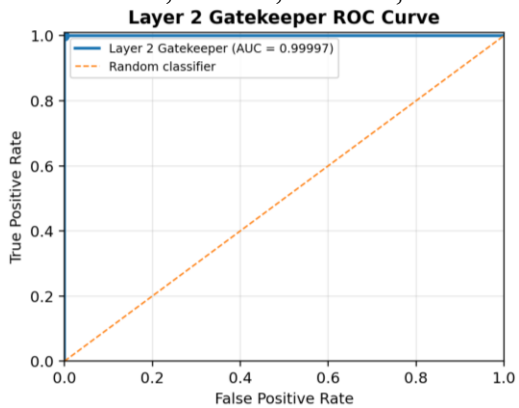


Figure 6. Layer 2 gatekeeper ROC curve (AUC = 0.99997).

The routed totals of Table 5 are not class totals. Layer 4 receives 145,661 flows, comprising the 145,653 true negatives and the 8 false negatives, since a missed attack is routed according to the gatekeeper's decision rather than its true label. The test set itself contains 145,653 benign and 30,871 attack flows, as given in Table 2.

Table 5. Layer 2 routing split (N = 176,524 test flows).

Route	Destination	Samples	Percentage
Attack (prob $\geq \tau_B$)	Layer 3 (Forensics)	30,863	17.5%
Benign (prob $< \tau_B$)	Layer 4 (Sentinel)	145,661	82.5%
Uncertain	N/A	0	0.0%

To characterise dispersion and statistical separability, exact interval estimates are reported from the confusion-matrix counts above. On the N = 176,524 test set the gatekeeper attains recall = 0.99974 (Wilson 95% CI [0.99949, 0.99987]) and precision = 1.00000 (Wilson 95% CI [0.99988, 1.00000]); the false-positive rate is 0 over 145,653 benign flows, with a Clopper–Pearson 95% upper bound of 2.53×10^{-5}, so the true FPR is bounded well below the operating points reported for the compared baselines. The five-fold AUC range of 0.99996–0.99998 is the available seed-free dispersion estimate; the pipeline was fitted once under a fixed seed, so an across-seed standard deviation cannot be estimated from a single run and is therefore not reported. These intervals quantify separability under the reported protocol; as Section 7.2 notes, they do not repair a split whose independence cannot be verified for want of flow identifiers, and the point estimates should be read subject to that caveat.

Multi-Class Forensics:

Layer 3 attains macro F1 = 0.99 and weighted F1 = 1.00 over the eight families. Per-class scores are reported in Table 6, and the normalised confusion matrix of Figure 7 shows where the residual error lies. Five families are separated essentially perfectly. The class weighting achieved its purpose: Brute Force, the rarest family by a wide margin, is not lost to the majority class.

The residual confusion is concentrated between DDoS LOIC UDP (F1 = 0.96) and DDoS HOIC (F1 = 0.93). Both are high-volume flooding tools that produce similar flow-level statistics, and the same mitigation applies to both, so the practical cost of the confusion is low. Two qualifications apply to the remaining columns. The macro F1 of 0.99 is computed over unrounded per-class scores; Table 6 reports values rounded to two decimal places, and the unweighted mean of the rounded column is 0.986. The families are also represented very unequally in the test partition. DoS Hulk supplies the large majority of attack flows, whereas Brute Force contributes 8 flows to the training split and a correspondingly small number to the test split. An F1 of 1.00 on a class of that size is consistent with the class being trivially separable, but the confidence interval around it is wide, and it should not be read as evidence that the classifier generalises on that family. Per-class support counts were not recorded during evaluation and cannot be reconstructed from the saved artefacts; reporting them, with confidence intervals, is a necessary addition to any extension of this work.

Table 6. Layer 3 per-class scores for the eight known attack families.

Attack Family	F1 Score	Precision	Recall
DoS Hulk	1.00	1.00	1.00
DoS GoldenEye	1.00	1.00	1.00
DoS Slowloris	1.00	1.00	1.00
DoS SlowHTTP	1.00	1.00	1.00
DDoS LOIC HTTP	1.00	1.00	1.00
DDoS LOIC UDP	0.96	0.96	0.96
DDoS HOIC	0.93	0.93	0.93
Brute Force	1.00	1.00	1.00

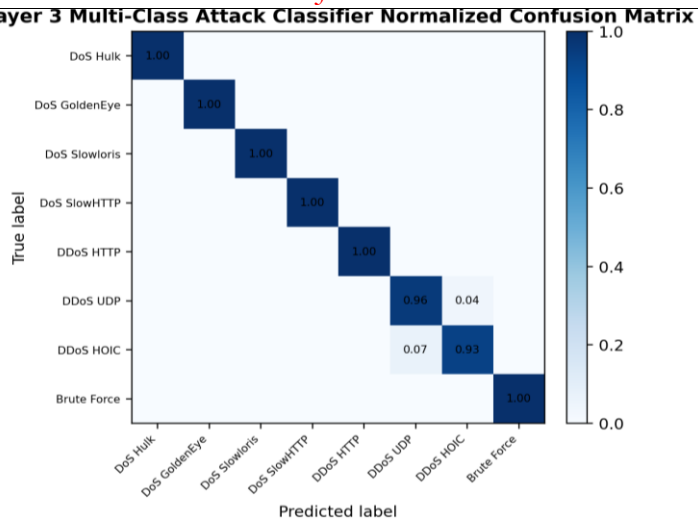


Figure 7. Layer 3 normalised confusion matrix for the eight attack families. Macro F1 = 0.99; weighted F1 = 1.00. The visible off-diagonal mass lies between DDoS HOIC and DDoS LOIC UDP.

Sentinel Evaluation on Held-Out Classes:

The sentinel is evaluated on the 76,026 withheld Botnet and Infiltration flows, which appeared in no training or validation partition. At the 95th-percentile threshold $\tau_U = 0.000001$, it detects 45.06% of them at a 5% false-positive rate, with AUC = 0.8205 for normal-versus-held-out discrimination. Figure 8 shows the score distributions: normal validation flows concentrate near zero while held-out attack flows spread substantially wider and reach markedly higher values.

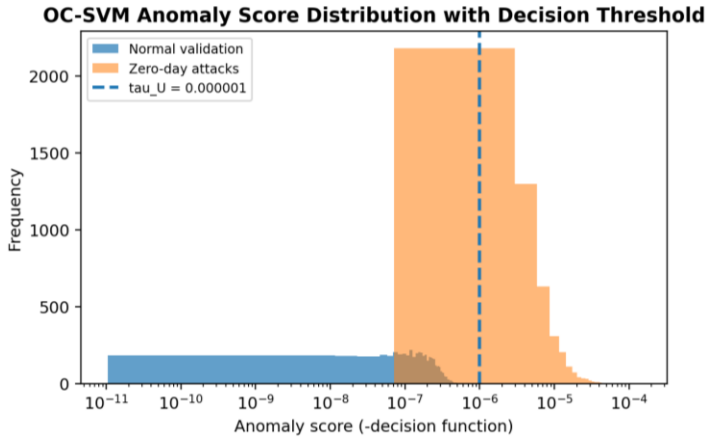


Figure 8. OC-SVM anomaly score distribution for normal validation flows and held-out attack flows, with the decision threshold $\tau_U = 0.000001$ set at the 95th percentile of normal validation scores.

A recall of 45.06% is moderate: more than half of the held-out attack flows were not flagged. Three qualifications attach to that figure. It is a combined result over Botnet and Infiltration; per-family recall was not recorded during evaluation and cannot be reconstructed from the saved artefacts, so one family may be detected substantially better than the other, and the combined figure would not reveal it. Precision and PR-AUC were not computed, and on a set that is 100% attack by construction, precision would in any case carry little information. The 5% false-positive rate is fixed by the threshold rule on the validation partition rather than estimated on the held-out set, which contains no benign flows against which false positives could arise. Two factors are consistent with the moderate recall. The RBF kernel is sensitive to feature scaling, and a One-Class SVM has limited representational capacity relative

to deeper anomaly models. The result supports a narrow conclusion: a percentile-thresholded one-class model provides a usable label-free safety net, but it does not substitute for supervised detection of families for which labels exist.

As an interval estimate, the held-out recall of 45.06% over 76,026 flows carries a Wilson 95% confidence interval of [44.71%, 45.41%], confirming that the moderate detection level is a stable property of the sample rather than merely an artefact of its size.

Feature Attribution:

Figure 9 ranks the top 20 Set A features by LightGBM gain for the gatekeeper. Timing and segment-level characteristics dominate the benign-versus-attack decision, which is consistent with prior findings that interarrival and header-level statistics are reliable indicators of DoS and DDoS traffic [10][21]. This attribution is also informative for Section 6: features of this kind are precisely those most exposed to the preprocessing differences between a cleaned CSV and raw CICFlowMeter output.

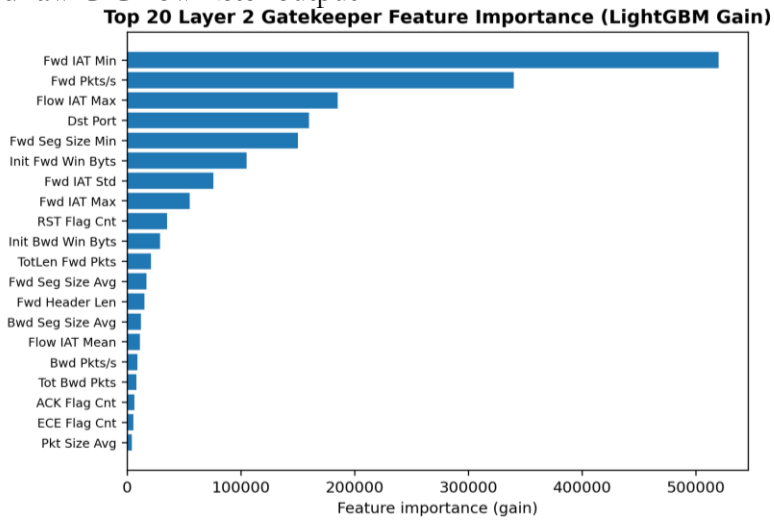


Figure 9. Top 20 Layer 2 gatekeeper features by LightGBM gain.

Layer Contribution Analysis

Because each layer answers a different question — Layer 2 routes, Layer 3 labels, Layer 4 screens the unlabelled — their contributions are not interchangeable and are quantified against the metric each governs, using the figures already established above rather than any re-fitted configuration. Removing the Layer 2 routing branch forces every flow through forensics and raises the per-flow cost from the weighted 6.32 ms to the 7.280 ms no-early-exit baseline, a 13.1% latency reduction attributable to Layer 2 routing alone, achieved with zero benign flows consuming forensic capacity since the gatekeeper produces no false positives. Removing Layer 3 collapses eight-family attribution at macro F1 = 0.99 to a single binary attack/benign verdict, eliminating the family-level labels on which targeted mitigation depends. Removing Layer 4 forfeits all coverage of classes absents from training, that is the 45.06% recall on held-out Botnet and Infiltration flows, because the supervised layers cannot flag what they were never shown. A full re-training ablation, in which each layer is re-fitted in isolation and re-measured, was not performed for this study: the reported figures derive from a single fixed-seed fit, and rather than present re-fitted numbers that the saved artefacts do not substantiate, each layer's contribution is quantified here from the metrics already measured. The controlled per-layer re-training ablation is specified as future work in Section 7.4.

Live Evaluation under Domain Shift:

Benchmark scores describe behaviour on the distribution the model was fitted to. To test whether they describe anything else, the pre-trained pipeline was run against a live DoS Hulk capture without retraining or recalibration of any kind. Withholding retraining is the experiment rather than an oversight: retraining on live data would measure how well the

architecture fits a second distribution, whereas the question here is what the artefacts produced by the benchmark protocol do when deployed unchanged, which is the operation that a practitioner performs when taking a published model into service.

Experimental Setup:

Two Kali Linux virtual machines were provisioned under Oracle VirtualBox, each with 4 GB RAM, 4 CPU cores, and 20 GB storage, connected through a bridged network adapter. One acted as the attacker, the other as the victim. The bridged adapter places both machines on the same segment as other devices rather than in a fully isolated segment; this is a limitation of the setup and is stated as such. All model artefacts and fitted scalers were exported from the development environment and loaded unchanged, which is the condition required for the domain-shift measurement to mean anything.

The attacker executed hulk.py against the victim's HTTP service on TCP port 80, generating a flood of requests with randomised URLs, headers, and user-agent strings. HULK was selected because the cache-busting HTTP flood it produces is the same attack behaviour the DoS Hulk class of the training data represents, so the live and benchmark traffic are nominally the same family and any divergence is attributable to capture and extraction rather than to a change of attack. The randomisation is confined to HTTP content: the network-layer source address remained constant at 192.168.0.111 and was not spoofed. Spoofing was omitted deliberately, since a varying source address would confound the measurement with a second source of feature drift, and the object here is to isolate the effect of the extraction path alone. That choice makes the detection task easier than an operational one, and the live figures should be read accordingly.

Wireshark on the victim captured the full interface, so the capture contains ordinary background traffic (ARP, DNS, and periodic system packets) intermixed with the attack, which is the separation problem an operational IDS faces; this is why the bridged adapter was preferred to an isolated segment, at the cost of admitting uncontrolled traffic the experiment does not account for. The resulting PCAP was processed by CICFlowMeter v0.5.0 into 11,138 flow records over the same 79-feature schema, ensuring structural compatibility with the training feature space. Version 0.5.0 was used rather than a later release because the feature semantics must match those of the training distribution; a version change would introduce extraction differences that could not then be separated from the domain shift under measurement.

Probability Collapse:

Table 7. Layer 2 calibrated probability distribution on the live capture, against the benchmark attack-flow distribution. The benchmark column is conditioned on attack flows; benign benchmark flows score near zero and are excluded from that column so that the two distributions are compared on the same class.

Statistic	Live Capture (all flows)	Benchmark (attack flows)
Mean Probability	0.0020	≈ 1.0000
Maximum Probability	0.0138	1.0000
99th Percentile	0.0068	1.0000
Median Probability	0.000126	1.0000
Minimum Probability	$\approx 1.59 \times 10^{-10}$	≈ 1.0000
Operating τ_B	0.0001 (manually lowered)	1.0000

The calibrated probabilities returned on live traffic occupy a different range from the benchmark. Table 7 reports the comparison. On the benchmark, attack flows receive calibrated probabilities at or near 1.0 while benign flows score near zero, and $\tau_B = 1.0$ separates the two groups. On live traffic, the maximum probability across all 11,138 flows, attack flows included, is 0.0138; the 99th percentile is 0.0068, the mean is 0.0020, and the median is

0.000126. No live flow reaches within two orders of magnitude of the benchmark operating point.

At $\tau_B = 1.0$, the gatekeeper therefore labels no live flow as an attack. The benchmark operating point does not yield a reduced detection rate on this capture; it yields none.

Three mechanisms are consistent with the collapse. First, the cleaned CSV was deduplicated, imputed, and timestamp-normalised before release, whereas the CICFlowMeter output used here was passed to the pipeline as emitted. The live capture contained no NaN, infinite, or zero-duration records (Table 8), so the difference is not one of missing values but of the feature distributions themselves: flow durations, packet rates, and interarrival statistics generated by a single host on an idle virtual segment differ systematically from those of the multi-host benchmark capture. Second, the RobustScaler fitted on the training partition maps such out-of-range inputs to scaled values beyond the interval it was fitted over. Third, the isotonic calibrator was fitted on out-of-fold predictions from that same partition and is a monotone step function defined over the training score range; inputs falling below that range are mapped to its lowest steps regardless of class. The three act together, compressing posteriors toward zero for benign and attack traffic alike. These mechanisms are inferred from the observed distributions rather than isolated experimentally, and an ablation separating their contributions is left to future work. Calibration is not the cause of the failure; it is what makes the failure measurable on an interpretable scale.

This behaviour is qualitatively different from the degradation reported in domain-adaptation studies. Prior cross-dataset studies report accuracy falling at a fixed operating point when a model trained on one dataset is applied to another. The effect measured here is qualitatively different: the operating point itself ceases to select any flow, because the calibrated scale is compressed roughly four orders of magnitude below the benchmark threshold. The distinction matters for practice — a reduced detection rate degrades gracefully and can be traded off, whereas an invalidated threshold fails silently, labelling nothing until it is reset by hand. We are aware of no prior flow-based study quantifying threshold invalidation of this magnitude, and an ablation isolating the three contributing mechanisms is identified as future work rather than claimed here.

Threshold Adaptation and Detection Outcome:

To obtain any output at all, τ_B must be lowered by hand. Two captures were taken under the same setup, attack tool, and source host, and the pipeline was run on each at a different manually chosen threshold. The first capture (10,700 flows, Figure 10) was processed at $\tau_B = 1 \times 10^{-8}$; the second (11,138 flows) at $\tau_B = 1 \times 10^{-4}$, a value close to the median of its calibrated distribution. Neither value was derived from an independent calibration capture. Both are reported because the pair measures what a single run cannot: how far the labelling of the same attack moves when only the threshold changes.

Table 8 reports both. At $\tau_B = 1 \times 10^{-8}$ the gatekeeper labels 9,346 flows (87.3%) as attacks; at $\tau_B = 1 \times 10^{-4}$ it labels 5,810 (52.2%). The ordering is what the calibrated distribution predicts, since a lower threshold admits a superset of flows, and the monotonicity indicates the pipeline behaved as specified. The magnitude is the finding. A four-order-of-magnitude change in a threshold, both settings being arbitrary with respect to any principled selection rule, move the labelled fraction of an identical attack by 35.1 percentage points. Neither figure can be called a detection rate. The captures contain background traffic mixed with the flood, and no per-flow ground truth was established, so the true attack fraction is unknown and lies somewhere below 87.3%; what the two runs establish is that the reported fraction is governed by an unconstrained parameter rather than by the evidence in the traffic.

In both runs, every attack-routed flow received the DoS Hulk label at Layer 3, with maximum class probabilities above τ_M. This is consistent with captures containing a single

attack family originating from a single source address and targeting a single destination port, and it indicates that the Set A discriminative structure transferred even though the calibrated scale did not. Table 9 illustrates the homogeneity of the output: across the ten most confident flows, the probability spans 0.013839 to 0.013241, a spread of 0.000598, as expected from a single tool emitting near-identical flows. The remainder of this section reports the second run, whose threshold is the less extreme of the two and whose sentinel scores were recorded in full.

Table 8. Live DoS Hulk evaluation. Two captures under an identical setup, each processed at a manually selected τ_B. Neither threshold was derived from an independent calibration capture. Percentages are the fraction of each capture labelled attack, not detection rates: no per-flow ground truth was established.

Metric	Run 1	Run 2
L2 Threshold τ_B (manual)	1×10^{-8}	1×10^{-4}
L2 Threshold τ_B (benchmark)	1.0	1.0
Total Flows Captured	10,700	11,138
Removed in Preprocessing	0	0
Flows Labelled Attack (L2)	9,346 (87.3%)	5,810 (52.2%)
Flows Routed to Sentinel (L2)	1,354 (12.7%)	5,328 (47.8%)
Layer 3 Label Assigned	All → DoS Hulk	All → DoS Hulk
Flows Flagged by Sentinel (L4)	0	0
Sentinel Threshold τ_U (live)	3,000	3,400
IP Blocked	192.168.0.111	192.168.0.111

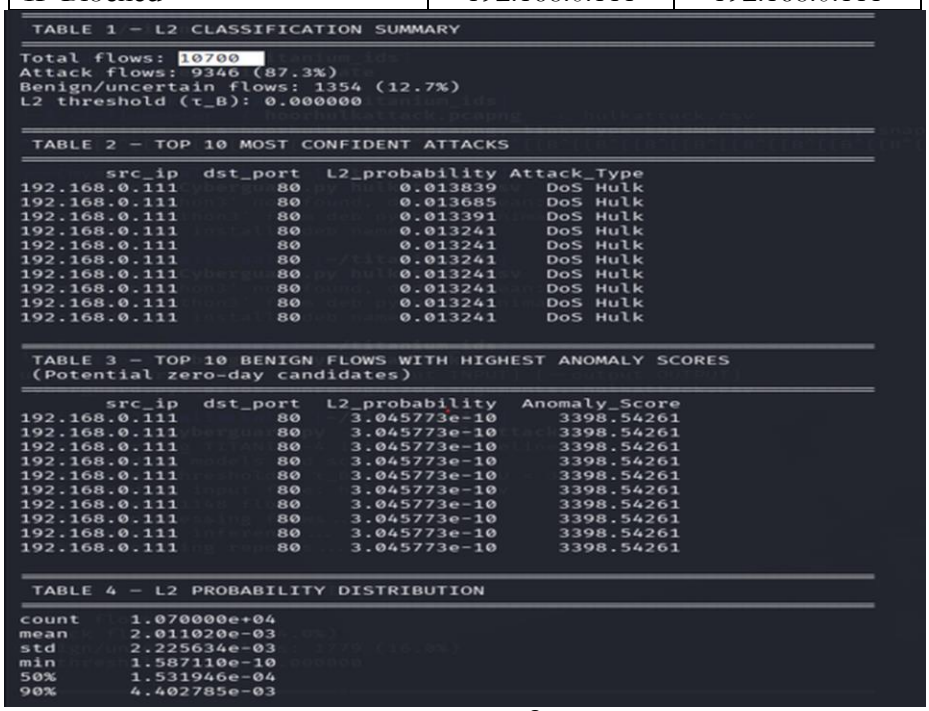


Figure 10. Run 1 inference output ($\tau_B = 1 \times 10^{-8}$). The summary reports 10,700 flows, 9,346 labelled attacks (87.3%), and 1,354 routed to the sentinel. The ten most confident flows span 0.013839 to 0.013241, and the ten benign-routed flows with the highest anomaly scores all sit at 3398.54261, the degenerate upper mode discussed in Section 6.4.

Table 9. Ten most confident attack classifications, Run 2.

scrip	dst port	L2 probability	Attack Type
192.168.0.111	80	0.013839	DoS Hulk
192.168.0.111	80	0.013685	DoS Hulk

192.168.0.111	80	0.013391	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk
192.168.0.111	80	0.013241	DoS Hulk

Sentinel Behaviour on Live Traffic:

The 5,328 benign-routed flows were scored by the sentinel (Table 10). The distribution is sharply bimodal. The median is 0.000001, so at least half of these flows sit at the low mode and are consistent with ordinary background traffic. The 90th, 95th, and 99th percentiles all converge at 3398.54, which is also the maximum, so the upper mode is a single repeated value occupying roughly the top tenth of the distribution rather than a spread. The sentinel is therefore separating the benign-routed flows into two sharply distinct groups; what it is not doing is acting on that separation.

Applying the same 95th-percentile rule used on the benchmark gives a live threshold of τU = 3,400, compared with the benchmark value of τU = 0.000001. At τ_U = 3,400, no flow among the 5,328 exceeds the threshold and the sentinel flags nothing. This outcome is an artefact of the procedure and is reported as such. The percentile was taken over the same flows the threshold was then applied to, and because the upper mode is a single degenerate value, the 95th percentile coincides with the maximum; the rule cannot flag anything by construction. On the benchmark, this circularity did not arise, since τ_U was fixed on the validation partition and applied to a disjoint held-out set. No equivalent held-out live partition existed here. That is the limitation the result exposes: a percentile rule requires a calibration sample distinct from the evaluation sample, and the live capture provided none. The pipeline therefore labelled 5,810 live flows as attacks, 52.2% of the capture, all at Layer 2, with no contribution from Layer 4.

Table 10. Sentinel anomaly score distribution for benign-routed live flows (N = 5,328).

Statistic	Anomaly Score
Count	5,328
Median (50th percentile)	0.000001
90th Percentile	3398.54
95th Percentile	3398.54
99th Percentile	3398.54
Threshold τ_U (live)	3,400
Threshold τ_U (benchmark)	0.000001
Flows Flagged as Anomalous	0

The result is negative but not empty. The sentinel separated the two modes cleanly, placing the bulk of the benign-routed traffic at the low mode and concentrating roughly a tenth of it at a sharply higher score, which indicates that its decision function still discriminates on live data after a shift that flattened the gatekeeper's calibrated scale. What the upper mode contains is not established here. Those flows were routed as benign by the gatekeeper at its adapted threshold, and no per-flow ground truth exists for the capture, so they may be attack flows the gatekeeper missed, unusual background traffic, or extraction artefacts; the experiment cannot distinguish these. Had τ_U been fixed on a separate calibration capture at any value below 3398.54, those flows would have been flagged, and the sentinel would have contributed detections rather than none. The failure is thus located in the thresholding

procedure rather than in the anomaly model, which is a narrower and more actionable finding than a wholesale failure of Layer 4 would have been.

Prevention Module:

A prevention module built on the Linux netfilter framework is integrated into the inference script. Once a batch is processed, the most frequent source address among attack-classified flows is inserted into the INPUT chain as a DROP rule via iptables; a background thread removes the rule after a configurable interval, 300 seconds by default, and a set of currently blocked addresses prevents duplicate insertion. In the live test, the module blocked 192.168.0.111 for 300 seconds.

The module's limitations are acknowledged rather than resolved. Detection-to-blocking response time was not instrumented, and blocking occurs after a batch completes rather than per flow, so the module does not interrupt an attack in progress within the 6.32 ms per-flow budget reported in Section 4.5; the two figures describe different operations and should not be combined. No traffic measurement was taken after the rule was inserted, so the capture confirms that the rule was applied but not that the attack traffic ceased. There is no administrator override to lift a block early, no logging beyond terminal output, no safeguard against blocking a legitimate address that happens to be the most frequent source among misclassified flows, and no handling of spoofed sources, the attack here having used a single non-spoofed address. The module demonstrates temporary blocking; it does not constitute an intrusion prevention system, and the paper makes no such claim. Detection, classification, alerting, and temporary blocking are distinct capabilities, and only the first three are evaluated here.

Deployment Considerations:

Operational feasibility is discussed within the measured bounds. On resource utilisation, the processed feature matrices are stored as float32 and peak memory remains below 4 GB on a 16 GB machine; the One-Class SVM retains approximately 6,797 support vectors, near 1% of the training set, so sentinel inference is a bounded kernel evaluation rather than a growing cost. On throughput, the per-layer profile of Section 4.5 shows that Layer 1 ingestion, not model inference, dominates the per-flow budget, because the measured Layer 1 cost includes file reading; an implementation reading from memory or a kernel-bypass path would remove the disk term and shift the budget onto the model layers, though no such implementation was built or measured and the size of that reduction is therefore not estimated. On hardware and scalability, all timings were obtained in software on a general-purpose development machine. The pipeline was not evaluated on networking hardware, no throughput measurement under high-speed line-rate traffic was taken, and no claim is made regarding compliance with any hardware-switch timing budget. This section is a feasibility discussion bounded by the measurements reported, not a hardware validation.

Limitations:

Threshold Transfer:

The central limitation is the one Section 6 measures. Both learned thresholds are properties of the benchmark domain: τ_B moved from 1.0 to 0.0001 and τ_U from 0.000001 to 3,400 between benchmark and live capture. Neither adapted value was derived from an independent calibration capture.

Three changes would be required before either threshold could be called deployable. The models should be trained on CICFlowMeter output so that training and inference share one preprocessing path, rather than on a pre-cleaned redistribution of it. Thresholds should be fixed on a live capture reserved for calibration and evaluated on a separate one, which is the disjointness the benchmark protocol had and the live protocol lacked. Finally, the percentile rule itself warrants reconsideration: because a percentile is defined relative to the sample it is computed over, it moves whenever the sample does. A threshold derived from an

assumed tail model, as in [16] and [17], may transfer more stably across domains. No such comparison was run here, and the claim that the percentile rule is the weaker choice under shift rests on the single capture reported in Section 6.4.

Possible Data Leakage in the Benchmark:

The cleaned CSV carries no flow identifiers, so it cannot be confirmed that duplicate or near-duplicate flows from a single attack session are confined to one split. Stratified random sampling does not enforce this. The Layer 2 and Layer 3 figures reported in Section 5, namely AUC = 0.99997, F1 = 0.99987, and macro F1 = 0.99, must therefore be read as a best-case estimate rather than a demonstrated generalisation result, and we consider the near-perfect scores themselves to be weak evidence of leakage rather than of unusual model quality. The same uncertainty applies in principle to the held-out Botnet and Infiltration evaluation, though it is bounded there: that set was excluded from training entirely and drawn from distinct attack labels. Resolving this requires a temporal or flow-based split with non-overlapping windows, verified against raw PCAP with flow identifiers present.

Scope of Evaluation:

Held-out evaluation covers two families, Botnet and Infiltration, on one dataset; results cannot be assumed to extend to families such as SQL injection or Heartbleed, or to other datasets. Cross-dataset evaluation on UNSW-NB15 [31] and CIC-IDS2017 is required. The live test covers one attack family from one non-spoofed source over one short capture window, on a bridged rather than isolated segment. Sentinel recall of 45.06% on held-out families leaves clear room for improvement. Latency was measured in software only; no hardware-switch claim is made. The τ_M uncertainty pathway was never triggered and remains unvalidated. The pipeline has no online learning component, so distribution drift after deployment is unaddressed.

Future Work:

The most valuable next step follows directly from Section 6: retrain and recalibrate on CICFlowMeter output so that training and deployment share a preprocessing path, and derive thresholds from held-out live captures rather than benchmark percentiles. Online recalibration, using temperature or Platt scaling over a small live validation sample, offers threshold adaptation without full retraining. Flow-identifier-based and temporal splits would resolve the leakage question directly. Alternative anomaly backbones (Isolation Forest, autoencoders [15], variational or hybrid ensembles) may raise the 45.06% held-out recall. The modular design permits retraining any single layer on domain-specific traffic such as IoT, industrial control, or software-defined networks, without redesigning the architecture.

Threat and Deployment Challenges:

Four challenges bear on any operational deployment and are stated here explicitly. Encrypted traffic removes application-layer visibility: the flow-level statistics used here remain available, but payload and header cues that aid HTTP-flood-type detection are lost, so performance on encrypted variants cannot be assumed from the results reported. Adversarial manipulation is a distinct risk: an attacker who shapes flow-level features toward the benign region could exploit the same calibrated-score sensitivity that domain shift exposes in Section 6, and the pipeline includes no adversarial-robustness mechanism. Evolving attack patterns are unaddressed by design: the pipeline has no online-learning component, so distribution drift after deployment degrades the fixed thresholds over time and periodic recalibration on in-domain captures is required. Multi-domain deployment is supported architecturally — any single layer can be retrained on IoT, industrial-control, or software-defined-network traffic without redesigning the pipeline — but each new domain reintroduces the threshold-transfer problem measured in Section 6 and must be recalibrated on captures from that domain rather than inheriting benchmark thresholds.

Conclusion:

Most network traffic is benign, and the per-flow cost of inspecting it need not equal the cost of inspecting an attack. TITANIUM-4 applies that principle: the Layer 2 gatekeeper routes 82.5% of benchmark test flows away from forensic processing, and the resulting weighted end-to-end cost is 6.32 ms per flow in software, against 7.280 ms for the same pipeline without early exit, a reduction of 13.1%. On the cleaned benchmark, the gatekeeper reaches F1 = 0.99987 and AUC = 0.99997 with zero false positives; the forensics layer reaches macro F1 = 0.99 across eight families under a near 12,000:1 imbalance, and the label-free sentinel flags 45.06% of held-out Botnet and Infiltration flows at 5% FPR.

The second set of results concerns the behaviour of those artefacts on traffic they were not fitted on. Calibrated probabilities were compressed to a maximum of 0.0138 against an operating threshold of 1.0, so the benchmark threshold labelled no live flow as an attack. A manual reduction to 0.0001 admitted 5,810 flows, 52.2% of the capture. The sentinel, applying on live data the same percentile rule used on the benchmark, produced a threshold four orders of magnitude higher than the benchmark value and flagged nothing, because that rule cannot flag anything when the percentile is taken over the same flows it is applied to and the upper mode is degenerate the pipeline labelled 52.2% of the capture as attacks, from the gatekeeper alone. The sentinel's scores nevertheless remained sharply bimodal on live data, separating roughly a tenth of the benign-routed flows from the rest, so its decision function tolerated the shift better than the gatekeeper's calibrated scale did; the failure lies in the thresholding rule applied to those scores.

Both sets of results are consistent with a single cause: the models were fitted to a cleaned redistribution of a benchmark and applied to the output of the flow-extraction tool itself. Held-out scores obtained on data whose split integrity cannot be verified, for want of flow identifiers, provide no evidence about that transfer. The benchmark figures are therefore reported here as a best-case estimate and the live figures as the operative ones. Running a single live capture against frozen artefacts is inexpensive relative to the cost of training the models, and on the evidence reported here it is more informative about deployability than any additional benchmark result would be.

Acknowledgments:

The authors acknowledge the Department of Electrical Engineering at the University of Engineering and Technology, Peshawar. We thank the Canadian Institute for Cybersecurity at the University of New Brunswick for the CSE-CIC-IDS2018 dataset, and the maintainers of LightGBM, scikit-learn, and CICFlowMeter.

Author Contributions:

Conceptualization and methodology, M. Younas; software and programming architecture, Sadeeda; validation, H.U. Ain; writing — original draft and technical writing, H. Shahzad; writing review and editing, I.U. Haq and M. Farooq; supervision, Wasim Habib, Salman Ilahi Siddiqui, I.U. Haq and M. Farooq. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest:

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Data Availability Statement:

The CSE-CIC-IDS2018 dataset is publicly available at https://www.unb.ca/cic/datasets/ids-2018.html. The TITANIUM-4 source code, trained models, and evaluation scripts are available at https://github.com/EngrMuhammadYounas/TITANIUM-4-IDS.

References:

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion

detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019, doi: 10.1186/S42400-019-0038-7/FIGURES/8.

[2] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[3] A. B. Bouidaine, D. Moussaoui, M. Hadjila, W. Ferhi, and M. H. Hachemi, "Deep learning-based anomaly and intrusion detection using the CSE-CIC-IDS2018 dataset," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 24782–24787, 2025, [Online]. Available: https://etasr.com/index.php/ETASR/article/view/11173

[4] Isuru Udayangani Hewapathirana, "A Comparative Study of Two-Stage Intrusion Detection Using Modern Machine Learning Approaches on the CSE-CIC-IDS2018 Dataset," *Knowledge*, vol. 5, no. 1, p. 6, 2025, doi: https://doi.org/10.3390/knowledge5010006.

[5] Q. M. Alzubi, S. N. Makhadmeh, and Y. Sanjalawe, "Optimizing Intrusion Detection: Advanced Feature Selection and Machine Learning Techniques Using the CSE-CIC-IDS2018 Dataset," *J. Adv. Inf. Technol.*, vol. 16, no. 3, pp. 283–302, 2025, doi: 10.12720/JAIT.16.3.283-302.

[6] "(PDF) Multi-Stage Enhanced Zero Trust Intrusion Detection System for Unknown Attack Detection in Internet of Things and Traditional Networks." Accessed: Jul. 17, 2026. [Online]. Available: https://www.researchgate.net/publication/390007608_Multi-Stage_Enhanced_Zero_Trust_Intrusion_Detection_System_for_Unknown_Attack_Detection_in_Internet_of_Things_and_Traditional_Networks

[7] "Sequential Intrusion Detection System For Zero-Trust Cyber Defense Of Iot/Iiot Networks." Accessed: Jul. 17, 2026. [Online]. Available: https://www.researchgate.net/publication/384425126_Sequential_Intrusion_Detection_System_For_Zero-Trust_Cyber_Defense_Of_Iotiiot_Networks

[8] "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," *Underst. Intrusion Detect. Through Vis.*, pp. 31–47, May 2006, doi: 10.1007/0-387-27636-X_3.

[9] "Intrusion Detection on CSE-CIC-IDS2018 Dataset Using Machine Learning Methods - Artificial Intelligence Theory and Applications." Accessed: Jul. 17, 2026. [Online]. Available: https://dergipark.org.tr/en/pub/aita/article/1553769

[10] Chaofei Tang, Nurbol Luktarhan, "An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder," *Symmetry (Basel).*, vol. 12, no. 9, p. 1458, 2020, doi: https://doi.org/10.3390/sym12091458.

[11] Baraa Ismael Farhan, Ammar D. Jasim, "Performance analysis of intrusion detection for deep learning model based on CSE-CIC-IDS2018 dataset," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 2, p. 1165, 2022, doi: 10.11591/ijeecs.v26.i2.pp1165-1172.

[12] "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." Accessed: Jul. 17, 2026. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[13] Muzun Althunayyan, Amir Javed, "A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning," *Veh. Commun.*, vol. 49, p. 100837, 2024, doi: https://doi.org/10.1016/j.vehcom.2024.100837.

[14] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, doi: 10.1162/089976601750264965.

[15] Polyzois Bountzis, Dimitris Kavallieros, "A deep one-class classifier for network anomaly detection using autoencoders and one-class support vector machines," *Front.*

Comput. Sci., vol. 7, 2025, doi: https://doi.org/10.3389/fcomp.2025.1646679.

[16] Sevvandi Kandanaarachchi, Mahdi Abolghasemi, "Detection of Anomalous Network Nodes via Hierarchical Prediction and Extreme Value Theory," *arXiv:2304.13941*, 2026, doi: https://arxiv.org/abs/2304.13941.

[17] T. T. Nguyen, C. S. Shieh, C. H. Chen, and D. Miu, "Detection of unknown DDoS attacks with deep learning and Gaussian mixture model," *Proc. - 2021 4th Int. Conf. Inf. Comput. Technol. ICICT 2021*, pp. 27–32, Mar. 2021, doi: 10.1109/ICICT52872.2021.00012.

[18] "A deep learning approach for intrusion detection in Internet of Things using focal loss function | Request PDF." Accessed: Jul. 17, 2026. [Online]. Available: https://www.researchgate.net/publication/367192516_A_deep_learning_approach_for_intrusion_detection_in_Internet_of_Things_using_focal_loss_function

[19] M. W. Nawaz, R. Munawar, M. K. Bhatti, A. Mehmood, M. M. U. Rahman, and Q. H. Abbasi, "Multi-Class Network Intrusion Detection with Class Imbalance via LSTM & SMOTE," *Proc. - 2025 27th IEEE Int. Conf. High Perform. Comput. Commun. 11th IEEE Int. Conf. Data Sci. Syst. 23rd IEEE Int. Conf. Smart City, 11th IEEE Int. Conf. o…*, pp. 824–831, 2025, doi: 10.1109/HPCC67675.2025.00123.

[20] Abdullah Alzaqebah, Ibrahim Aljarah, "A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization," *Comput. Secur.*, vol. 124, p. 102957, 2023, doi: https://doi.org/10.1016/j.cose.2022.102957.

[21] Y. A. Rani, K. Deepthi Reddy, and R. U. Rani, "A Novel Network Intrusion Detection Model using Residual Recurrent Neural Network with Improved Garter Snake-based Optimization Strategy," *2023 Glob. Conf. Inf. Technol. Commun. GCITC 2023*, 2023, doi: 10.1109/GCITC60406.2023.10426136.

[22] Marija Gombar, Amir Topalović, "Cost-Aware Lightweight Deep Learning for Intrusion Detection: A Comparative Study on UNSW-NB15 and CIC-IDS2017," *Electronics*, vol. 15, no. 8, p. 1603, 2026, doi: 10.3390/electronics15081603.

[23] C. Guida, A. Nascita, A. Montieri, and A. Pescape, "Cross-Evaluation of Deep Learning-based Network Intrusion Detection Systems," *Proc. - 2023 Int. Conf. Futur. Internet Things Cloud, FiCloud 2023*, pp. 328–335, 2023, doi: 10.1109/FICLOUD58648.2023.00055.

[24] Bianca Zadrozny, Charles Elkan, "Transforming Classifier Scores into Accurate Multiclass Probability Estimates," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2002, doi: 10.1145/775047.775151.

[25] Eugene Berta (SIERRA), Francis Bach (SIERRA), Michael Jordan (SIERRA), "Classifier Calibration with ROC-Regularized Isotonic Regression," *arXiv:2311.12436*, 2023, [Online]. Available: https://arxiv.org/abs/2311.12436

[26] Ehssan Mousavipour, Andrey Dimanchev, Majid Ghaderi, "Shift Detection and Adaptation for Network Intrusion Detection," *arXiv:2508.15100*, 2026, [Online]. Available: https://arxiv.org/abs/2508.15100

[27] M. Pawlicki, S. Szelest, R. Kozik, and M. Choraś, "SHAP Insights into Domain Adaptation in Netflow-Based Network Intrusion Detection Powered by Deep Learning," *Lect. Notes Comput. Sci.*, vol. 15999 LNCS, pp. 292–309, 2025, doi: 10.1007/978-3-032-00644-8_18/SAVE-RESEARCH.

[28] Chaonan Xin, Keqing Xu, "Cross-Dataset Transformer-IDS with Calibration and AUC Optimization (Evaluated on NSL-KDD, UNSW-NB15, CIC-IDS2017)," *J. Cyber Secur.*, vol. 7, no. 1, 2025, [Online]. Available: https://www.semanticscholar.org/paper/Cross-Dataset-Transformer-IDS-with-Calibration-and-Xin-Xu/daf8e3d4ae33e5ecb04ab5da524deb75d47b2d27

[29] "CSE-CIC-IDS2018 on AWS", [Online]. Available:

https://www.unb.ca/cic/datasets/ids-2018.html

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-January, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[31] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, Dec. 2015, doi: 10.1109/MILCIS.2015.7348942.