Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.086180

ARTICLE

Hierarchical Adversarially-Driven Escalation System (HADES) for Network Intrusion Detection

Abdelouahid Derhab[1,*], **Adlen Kerboua**[2], **Noureddine Seddari**[3,4], **Anis Haniche**[5] and **Mohammad Mehedi Hassan**[6]

[1]Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, Saudi Arabia
[2]LGMM Laboratory, University 20 Août 1955-Skikda, Skikda, Algeria
[3]LICUS Laboratory, Department of Computer Science, University 20 Août 1955-Skikda, Skikda, Algeria
[4]Future Technology Laboratory, University of Parma, Parco Area delle Scienze 181/A, Parma, Italy
[5]SecDevOPS Consulting, 61 Rue de Lyon, Paris, France
[6]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

*Corresponding Author: Abdelouahid Derhab. Email: abderhab@ksu.edu.sa

Received: 25 May 2026; Accepted: 10 August 2026; Published: 28 September 2026

ABSTRACT: Machine learning has radically transformed network security, enabling intrusion detection systems capable of identifying malicious traffic with near-perfect accuracy on standard benchmarks. However, these systems remain critically vulnerable to adversarial examples—subtly manipulated inputs designed to escape detection—where performance can severely drop under minimal perturbation. This paper introduces the Hierarchical Adversarially-Driven Escalation System (HADES), a framework that addresses this vulnerability through three coordinated mechanisms. First, dedicated detectors are trained for each network protocol, enabling each model to specialize in specific traffic patterns it will face in practice. Second, these detectors are continuously hardened by simulating an arms race between an attacking agent, which learns to find the most damaging evasion strategies, and a defending model that adapts in response, thus producing classifiers that remain robust across a wide range of attack types. Third, incoming traffic is routed through a cost-aware pipeline that reserves expensive analysis for uncertain or suspicious flows, keeping average processing time at 5.4 ms per batch on normal traffic. HADES is evaluated on CIC-IDS-2018, a large-scale real-world network dataset, and maintains near-perfect detection accuracy under both normal and adversarial conditions, with robustness verified across nine distinct attack strategies and 95% bootstrap confidence intervals of maximum width 0.0007.

KEYWORDS: Network intrusion detection; adversarial robustness; adversarial training; protocol-stratified detection; multi-tier cascade; parameter-efficient learning

1 Introduction

Network intrusion detection systems (NIDS) form a critical line of defense against cyberattacks, monitoring traffic in real time to detect suspicious or malicious activity, and enabling timely responses to reduce potential harm. The advent of deep learning has significantly improved detection performance, i.e., modern ML-based NIDS routinely exceed 99% F1-score on widely used benchmarks such as CIC-IDS-2017 and CIC-IDS-2018 [1], establishing the viability of data-driven detection at scale. However, this strong overall performance masks a critical vulnerability. Adversarial examples, i.e., inputs altered with imperceptibly small, intentionally crafted perturbations, can lead even high-performing systems to misclassify malicious traffic as benign. Recent evaluations report F1-score drops of more than 60 percentage points under low-cost

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

adversarial attacks [1,2], a vulnerability that grows more consequential as network volumes push beyond 100 Gbps in 6G and IoT environments [3] and as attackers increasingly incorporate evasion into their toolkits [4].

Closing this robustness gap is non-trivial for three reasons that, to the best of the authors' knowledge, no prior system addresses jointly. First, existing adversarial defenses for NIDS, including GADoT [4], the Evolving Adversarial Training framework [5], and recent ensemble approaches in [6], are hardened against a single, fixed attack algorithm. A rational adversary, however, will adaptively switch evasion strategies to target whichever detector is least robust. This work shows empirically that this strategic flexibility creates an F1-score gap of up to 0.56 on a single protocol before hardening, meaning that single-strategy robustness evaluations substantially overestimate real-world resilience. Second, real network traffic is not drawn from a single statistical distribution. Protocols such as HTTP, DNS, SSH, and FTP exhibit fundamentally different flow characteristics, attack vectors, and class imbalance ratios [3,7]. Training a single unified detector across all protocols forces a statistical compromise, leaving each protocol's decision boundary suboptimally defined and disproportionately vulnerable to a targeted adversary. Third, no existing work formalizes the cost of routing flows through detectors of different complexities. A lightweight detector can classify a batch of 256 flows in under 2 ms, while a language-model-based investigator requires orders of magnitude more time for the same batch [8]. Applying the heaviest model to all traffic is computationally indefensible at this scale, yet current multi-tier architectures [3,7] rely on static, heuristic routing rules with no formal cost-optimality guarantee. This paper introduces HADES (Hierarchical Adversarially-Driven Escalation System), a framework that addresses all three limitations within a unified optimization objective. The central insight is that adversarial robustness, protocol specialization, and inference cost are interdependent rather than independent and must be optimized jointly. HADES implements this optimization objective through three coordinated pillars: (1) Protocol-specific detectors are trained using parameter-efficient adaptation [9], giving each model a compact footprint tuned to the statistical structure of its target protocol; (2) These detectors are then hardened through a co-evolutionary training loop in which an adaptive adversarial agent, guided by a bandit algorithm that learns which attack strategies are most damaging [10], competes against a defending model that adjusts its robustness objective in response [11]; and (3) Incoming flows are routed through a three-tier cascade governed by a formally derived confidence threshold: flows that the lightweight detector classifies with high certainty exit immediately, while uncertain or suspicious flows are escalated to progressively more powerful but more costly models, thereby minimizing mean processing time on benign traffic.

The main contributions of this paper are threefold:

- A formal vocabulary is introduced for strategy-distribution robustness, and it is proven that worst-case robustness equals the strategy-distribution metric for a fixed detector, with the training-time consequence that a fixed attack curriculum systematically misses the evolving worst case.
- HADES is designed and implemented, integrating protocol-stratified LoRA detectors[1], a co-evolutionary Red-Blue adversarial training loop, and a Bayes-optimal confidence-gated escalation cascade.
- The proposed method is evaluated on the CIC-IDS-2018 dataset. The results show near-perfect clean and adversarial F1 scores with 95% bootstrap confidence intervals of maximum width 0.0007. The traffic is routed through a cost-aware pipeline that reserves expensive analysis for uncertain or suspicious flows, yielding an average processing time of 5.4 ms per batch on normal traffic.

[1]The attention-layer adapter is a LoRA-inspired rank-r output-residual, not a QKV-factored LoRA in the sense of [9]; the exact formulation and the reason for the naming choice are given in Section 4.2.

The remainder of the paper is organized as follows. Section 2 reviews related work on adversarial robustness in intrusion detection, hierarchical detection, language models as reasoning engines, and bandit-based adaptive adversaries. Section 3 establishes the mathematical foundations of HADES, covering the detection setting, the adaptive adversary model, the robustness metrics, the minimax optimization problem, and the three-pillar decomposition. Section 4 describes the system composition, detailing the LoRA-stratified Transformer Detector (Pillar 1), the co-evolutionary Red-Blue training loop (Pillar 2), and the Bayes-optimal cost-sensitive escalation cascade (Pillar 3). Section 5 reports the experimental evaluation on CIC-IDS-2018, covering stratified vs. unified robustness gaps, strategy-distribution robustness against nine adversarial strategies, routing cost and escalation efficiency, and a consolidated component ablation. Section 6 concludes and identifies eleven future research directions.

2 Related Work

The literature study is organized along four dimensions corresponding to the three pillars of HADES, adding the cross-cutting concern of distributional generalization.

2.1 Adversarial Robustness in Network Intrusion Detection

The adversarial vulnerability of ML-based NIDS has been extensively documented. Ennaji et al. [1] showed that models achieving above 99% F1-score on CIC-IDS-2017 and UNSW-NB15 collapse by more than 60 percentage points under low-budget gradient-based or generative perturbations, identifying evasion attacks as the dominant failure mode.

Tafreshian and Zhang [2] confirmed this brittleness across multiple architectures and noted that worst-case performance drops are consistently underestimated when evaluation relies on a single attack algorithm.

In GADoT [4], the authors use a generative adversarial network to synthesize evasive DDoS flows for adversarial training. A related generative line is followed by Alshehri et al. [12], who couple a Wasserstein GAN with an autoencoder to harden IoT intrusion detection against distributional irregularities; as with GADoT, robustness is obtained with respect to a fixed generative perturbation process rather than to an adversary that reselects its strategy in response to the current detector. The Evolving Adversarial Training (EAT) framework from [5] advances this further by combining continuous model evolution with a hand-crafted curriculum over gradient-based attack variants, scheduling increasingly aggressive perturbations over training epochs. The ensemble defense designed by Awad et al. [6] stacks adversarially trained base classifiers evaluated on the CIC-IDS-2018 dataset. All three previously mentioned systems share a fundamental assumption: the attacker commits to a fixed perturbation algorithm or a pre-specified sequence, leaving open the question of how a detector should behave when the adversary freely switches between strategies. Zhang et al. [11] provided the theoretically principled robustness-accuracy training objective used as the basis of the defending model in HADES, but they do not address strategy diversity or the possibility that the worst-case attack shifts over the course of training.

Certified robustness via randomized smoothing [13] provides provable robustness guarantees under Gaussian noise. However, the perturbation model relevant to network flow classification is bounded by the infinity norm rather than the Euclidean norm, making smoothing-based certificates inapplicable without significant adaptation.

2.2 Hierarchical and Resource-Adaptive Detection

The computational constraints of real-time network monitoring have motivated hierarchical detection architectures. In C2T-IDS [3], authors implement a two-tier design in which a lightweight ensemble handles

real-time binary classification while a deeper model performs multi-class attack diagnosis for flagged flows. Similarly, Meta-IDS [7] introduces a multi-stage pipeline that varies model depth and computational budget in response to traffic volume peaks. Both systems assign flows to tiers based on input-side features, such as packet headers or traffic volume thresholds, rather than based on the detector's uncertainty about a specific sample. A structurally ambiguous flow is thus processed at whatever tier the routing rule assigns it, with no mechanism to escalate based on prediction confidence. Neither system provides a formal cost-optimality argument for its routing policy. In particular, neither connects routing decisions to the principled cost-sensitive classification framework of Elkan [14] that underlies the escalation gate of HADES.

2.3 Language Models as Reasoning Engines for NIDS

SecureBERT [8] is a BERT-based encoder pre-trained on a cybersecurity corpus, including vulnerability descriptions, threat reports, and network logs, yielding representations that capture the semantic attack context beyond what statistical flow features encode. Adjewa et al. [15] demonstrated that fine-tuned language models can detect low-and-slow attacks, such as slow HTTP POST and DNS tunneling, which statistical classifiers miss, achieving 95.6% accuracy on IoT traffic. As the network throughput exceeds 100 Gbps, a language model that requires orders of magnitude more inference time per batch than a lightweight classifier cannot sustain real-time operation. HADES resolves this concern by positioning SecureBERT as a high-accuracy fallback that receives only suspicious flows for which the lightweight Tier-1 model produces a low-confidence prediction, thereby formalizing the cost-accuracy trade-off that prior language-model-based IDS work leaves implicit. In HADES, parameter-efficient fine-tuning via Low-Rank Adaptation [9] is applied to SecureBERT, enabling adaptation to the network traffic distribution with a small number of additional trainable parameters.

2.4 Protocol Stratification and Adaptive Attack Selection

The heterogeneity of network protocols has received limited attention as a first-class design dimension in intrusion detection. Most benchmarked systems train a single classifier on the joint traffic distribution, implicitly assuming that a shared decision boundary adequately serves all protocol subspaces. This assumption is violated when class imbalance ratios, attack vector distributions, and feature relevance differ substantially across protocols, as they do in CIC-IDS-2018 (attack traffic proportions range from 38.6% for SSH to 92.8% for HTTP). Low-Rank Adaptation (LoRA) [9] was originally introduced for large language model fine-tuning; the same decomposition principle transfers directly to the network flow classification setting, enabling per-protocol specialization with a small fraction of additional trainable parameters per model. To the best of the authors' knowledge, no prior work applies this technique to protocol-stratified intrusion detection. Transformer- and ViT-based detectors have also been applied to IoT intrusion detection [16], high-performance detection with a hybrid split-attention/Transformer architecture in [17], reinforcement-learning DDoS detection in consumer IoT healthcare in [18], data-analytic insider-threat detection in IoT in [19], hybrid data-balancing IDS for IoT in [20], and machine-learning IDS for vehicular ad-hoc networks in [21].

On the attack-generation side, the EAT framework [5] cycles through a fixed sequence of gradient-based attack variants without any feedback from the current detector's vulnerability profile: the curriculum is determined before training begins and does not change according to strategies that currently pose the greatest threat. HADES takes a fundamentally different approach, using a bandit algorithm [10] that updates the probability of selecting each attack strategy based on how damaging it proves at each training step. This adaptive selection concentrates training pressure on the detector's current weakest point, tracking the non-stationary worst-case strategy as the model evolves. The bandit regret bound holds against adversarial, non-stationary reward sequences, guaranteeing sub-linear regret relative to the best fixed strategy in hindsight.

Previous work has demonstrated that models achieving state-of-the-art performance on CIC-IDS-2017 can degrade near-randomly on CIC-IDS-2018 due to distributional shift [22]. EADA [23] addresses cross-dataset generalization by learning protocol-invariant features through domain-adversarial training, improving the transfer F1-score at the cost of within-distribution adversarial robustness.

2.5 Comparison with Related Work

Table 1 compares HADES against the most closely related systems across the following six criteria:

- **Adv. Train.**: It indicates the use of adversarial training of any kind.
- **Multi-Proto.**: It indicates the use of per-protocol specialized models or multi-protocol coverage.
- **Adapt. Atk.**: It indicates the use of an adaptive attack strategy selection (bandit/evolutionary).
- **Tiered Arch.**: It indicates the use of a cascade or multi-tier detection architecture.
- **Cost-Aware**: It indicates the use of uncertainty-driven routing with formal cost-optimality.
- **NLP/LLM**: It indicates the use of a language model as a detection component.

Table 1: Comparison of representative NIDS systems across six dimensions. ✓ = satisfied; ○ = partially; – = absent.

System	Adv. Train.	Multi-Proto.	Adapt. Atk.	TieredArch.	Cost-Aware	NLP/LLM
GADoT [4]	✓	–	–	–	–	–
EAT [5]	✓	–	○	–	–	–
Ensemble [6]	○	–	–	–	–	–
C2T-IDS [3]	–	–	–	✓	○	–
Meta-IDS [7]	–	○	–	✓	○	–
LLM-IDS [15]	–	–	–	–	–	✓
IRIS [22]	–	○	–	–	–	–
EADA [23]	○	○	–	–	–	–
Ennaji et al. [1]	–	–	–	–	–	–
HADES (ours)	✓	✓	✓	✓	✓	✓

EAT receives partial credit for multi-strategy training because it cycles through a curriculum of attack variants, but the curriculum is fixed before the training and does not respond to the detector's evolving vulnerability profile. C2T-IDS and Meta-IDS receive partial credit for cost-aware routing because they implement multi-tier architectures, but without the formal cost-optimality criterion [14] or confidence-based escalation that HADES formalizes.

3 Formal Foundations of HADES

3.1 Detection Setting

Let $\mathcal{X} = \mathbb{R}^d$ be the feature space of network flows, and let $\mathcal{Y} = \{0, 1\}$ denote the label space (0: benign, 1: attack). The traffic space is partitioned into P disjoint protocol subspaces $\{\mathcal{X}_p\}_{p=1}^{P}$; each flow is drawn from a protocol-specific distribution $\mathcal{D}_p$ over $\mathcal{X}_p \times \mathcal{Y}$.

A *detector* is a function $f : \mathcal{X} \to [0, 1]$ mapping a flow to its estimated attack probability; the hard classifier at threshold $\tau \in (0, 1)$ is

$$h_{f,\tau}(x) \;=\; \mathbf{1}[f(x) \geq \tau]. \tag{1}$$

A *stratified detector family* $\mathbf{f} = \{f_p\}_{p=1}^{P}$ assigns a dedicated detector to each protocol subspace. For routing decisions, the method uses the *confidence-based uncertainty*

$$u_p(x) = 1 - 2|f_p(x) - 0.5| \in [0,1], \tag{2}$$

where $u_p(x) = 0$ is maximum confidence and $u_p(x) = 1$ is maximum uncertainty. $u_p(x)$ is adopted as a cheap, monotone escalation proxy.

3.2 The Adaptive Adversary Model

Definition 1 (Perturbation Budget): *Given $x \in \mathcal{X}_p$ and $\varepsilon > 0$, the adversarial feasibility set is*

$$\mathcal{B}_\varepsilon(x) = \{x' \in \mathcal{X}_p : \|x' - x\|_\infty \le \varepsilon\}. \tag{3}$$

The constraint $x' \in \mathcal{X}_p$ enforces semantic validity: cross-protocol perturbations are not physically realizable. Note: when attacks are generated in continuous $\mathbb{R}^d$ without explicit per-feature feasibility constraints, perturbed vectors may fall outside the manifold of realizable network flows; robustness estimates should then be interpreted as worst-case bounds within the ℓ_∞ ball rather than as robustness to physically possible adversarial traffic.

Throughout, $\varepsilon_{\max}$ denotes the evaluation budget at which worst-case robustness is assessed; its concrete value is an operator-specified parameter stated at each point of use.

Definition 2 (Attack Strategy and Strategy Pool): *An attack strategy $a \in \mathcal{A}$ produces an adversarial example $x^{adv} = a(f_p, x, \varepsilon) \in \mathcal{B}_\varepsilon(x)$. The strategy pool $\mathcal{A} = \{a_1, \ldots, a_K\}$ is a finite set of perturbation algorithms; $\mathcal{A}_{train} \subseteq \mathcal{A}$ denotes the sub-pool used during adversarial training. The theoretical guarantees hold for any finite $\mathcal{A}$; the concrete instantiation is an operator choice given in Section 4.*

Definition 3 (Adaptive Adversary): *An adaptive adversary maintains a mixed strategy $\pi \in \Delta(\mathcal{A})$ and selects $a \sim \pi$ at each interaction to maximize detection loss. All prior IDS adversarial training work implicitly restricts π to a Dirac mass δ_{a_i} on a single strategy [4–6]; HADES makes no such assumption.*

3.3 Robustness Metrics

Definition 4 (Single-Strategy Robustness): *The adversarial F1 of f_p under strategy a at budget ε is the population-level F1 score $R_{a,\varepsilon}(f_p)$, where precision and recall are computed over $(x, y) \sim \mathcal{D}_p$ after applying $a(f_p, x, \varepsilon)$ to all flows. Two conventions hold throughout: (i) attacks are applied to all flows, capturing both evasion and false-triggering; (ii) the true label y is invariant within $\mathcal{B}_\varepsilon(x)$. $R_a(f_p)$ denotes the case $\varepsilon = \varepsilon_{\max}$.*

Definition 5 (Strategy-Distribution Robustness): *The strategy-distribution robustness (SDR) of f_p is*

$$\mathcal{R}(f_p) = \min_{\pi \in \Delta(\mathcal{A})} \mathbb{E}_{a \sim \pi}\left[R_{a,\varepsilon_{\max}}(f_p)\right]. \tag{4}$$

Proposition 1 (SDR Collapses to Worst-Case): *Since $\mathbb{E}_{a\sim\pi}[R_{a,\varepsilon_{\max}}(f_p)]$ is linear in π, its minimum over the compact convex set $\Delta(\mathcal{A})$ is attained at a vertex δ_{a_i}, giving*

$$\mathcal{R}(f_p) = \min_{a \in \mathcal{A}} R_{a,\varepsilon_{\max}}(f_p). \tag{5}$$

Proof: Extreme points of $\Delta(\mathcal{A})$ are $\{\delta_{a_i}\}_{i=1}^{K}$; a linear objective on a compact convex polytope is minimized at a vertex. □

Proposition 1 is a restatement of a standard LP fact included only to fix vocabulary $\left(\mathcal{R}(f_p) = \min_a R_a(f_p)\right)$. The substantive consequence for training is non-stationarity:

Proposition 2 (Non-Stationary Worst-Case Tracking): *Let* $\{f_p^{(t)}\}_{t=0}^{T-1}$ *be the sequence of per-generation detectors produced by a co-evolutionary loop, and let* $a_t^\star = arg\,min_{a\in\mathcal{A}} R_{a,\varepsilon}(f_p^{(t)})$ *denote the per-generation worst-case strategy. If* $|\{a_t^\star : t \in [T]\}| > 1$*, then any fixed-strategy curriculum* $\pi_t = \delta_{a_j}$*, where* $a_j \in \mathcal{A}$ *is a single strategy committed to before training and* δ_{a_j} *is the Dirac point mass on* a_j*, covers* $a_t^\star$ *in strictly fewer generations (in expectation) than any mixed strategy satisfying* $\pi_t(a_t^\star) \geq 1/K$ *for all* t*.*

Proof sketch: Under $\pi_t = \delta_{a_j}$, the defending model receives gradient signals from a_j at every generation; generations where $a_j \neq a_t^\star$ contribute no pressure toward closing the true worst-case gap. A mixed strategy with an exploration floor $\gamma > 0$ allocates positive mass to every strategy at every t, so the expected number of generations in which the defender sees $a_t^\star$ is at least T/K, compared to at most $|\{t : a_t^\star = a_j\}|$ for the fixed curriculum. □

Proposition 2 is the core motivation for the Exp3 (Exponential-weight algorithm for Exploration and Exploitation) [10] loop: the worst-case strategy *shifts during training* and cannot be tracked by a committed fixed curriculum.

Proposition 3 (Strategy-Distribution Robustness Gap): *For any* f_p *and any* $a_i \in \mathcal{A}$, $\mathcal{R}(f_p) \leq R_{a_i,\varepsilon_{\max}}(f_p)$*. The gap* $\Delta_{a_i}(f_p) = R_{a_i}(f_p) - \mathcal{R}(f_p) \geq 0$ *quantifies the overestimation produced by single-strategy evaluation.*

Proof: Immediate from Proposition 1. □

Definition 6 (Strategy-Mixture Robustness): *For any mixture* $\pi \in \Delta(\mathcal{A})$*, the* π*-mixture robustness of* f_p *is* $\mathbb{E}_{a\sim\pi}[R_{a,\varepsilon}(f_p)]$*. The quantity reported throughout this paper, denoted* Φ_{fp}*, is the uniform mixture over the fixed evaluation pool* $\mathcal{A}$ *(*$|\mathcal{A}| = K$*):*

$$\Phi_{\text{fp}}(f_p) = \mathbb{E}_{a\sim\text{Unif}(\mathcal{A})}\left[R_{a,\varepsilon}(f_p)\right] = \frac{1}{K}\sum_{a\in\mathcal{A}} R_{a,\varepsilon}(f_p). \tag{6}$$

The subscript "fp" denotes fixed pool: the average is taken over the fixed nine-strategy evaluation pool with equal weights, and does not depend on the weights learned by Exp3. For completeness, the corresponding Exp3-weighted quantity obtained from the final bandit weights $\pi^\star \in \Delta(\mathcal{A})$ *produced by the co-evolutionary loop is*

$$\Phi^{\text{Exp3}}(f_p) = \sum_{a\in\mathcal{A}} \pi_a^\star\, R_{a,\varepsilon}(f_p), \tag{7}$$

which is logged as a training diagnostic only and is never used as an evaluation metric in this paper; every value labelled Φ *or* Φ_{fp} *in Section 5 is the uniform average of Eq. (6). Both satisfy* $\Phi_{fp}(f_p) \geq \mathcal{R}(f_p)$ *and* $\Phi^{Exp3}(f_p) \geq \mathcal{R}(f_p)$*, with equality for* Φ^{Exp3} *if and only if* $\pi^\star$ *concentrates on the per-protocol worst-case strategy. In the Experiments section, both* Φ_{fp} *(the pool-average metric) and* $\mathcal{R}(f_p) = \min_a R_a(f_p)$ *(the true worst-case guarantee) are reported.*

Definition 7 (Portfolio Robustness):

$$\mathcal{R}_{\text{portfolio}}(\mathbf{f}) = \min_{p\in[P]} \mathcal{R}(f_p) = \min_p \min_a R_{a,\varepsilon_{\max}}(f_p). \tag{8}$$

A meta-adversary concentrating on the weakest protocol achieves system-level attack success bounded below by $1 - \mathcal{R}_{portfolio}(\mathbf{f})$*.*

On portfolio equalization.

Independent per-protocol training provides no direct mechanism to equalize $\{\mathcal{R}(f_p)\}$: each f_p is hardened in isolation, with no cross-protocol loss term redistributing gradient pressure toward the weakest protocol. The traffic-weighted surrogate (Eq. (11)) provides indirect equalization by weighting each protocol's

loss by its deployment importance w_p, but does not enforce a floor on $\min_p \mathcal{R}(f_p)$ directly. A portfolio-equalizing loss with adaptive per-protocol weights could redistribute pressure toward weak protocols, but would require shared parameters across $\mathbf{f}$ and is incompatible with the strict stratification assumption; this is deferred to future work.

3.4 The HADES Optimization Problem

A HADES system consists of a stratified detector family $\mathbf{f}$, a routing policy $\rho : \mathcal{X} \times [0,1] \to \mathcal{K}$ that maps each flow and its attack-probability score to a tier $k \in \mathcal{K} = \{k_1, \ldots, k_M\}$ at cost $c(k_1) \leq \cdots \leq c(k_M)$, and a training procedure $\mathcal{T}$.

Since the per-protocol detectors $\{f_p\}$ are independent, a rational meta-adversary concentrates its full budget on the weakest protocol $p^* = arg\,min_p \min_a R_{a,\varepsilon_{\max}}(f_p)$. Under this concentration, the inner maximization over $\Delta(\mathcal{A} \times \mathcal{E})$ reduces to a pure protocol-and-strategy choice, where $\mathcal{E}$ denotes the set of admissible per-protocol perturbation budgets $\{\varepsilon_p\}_{p=1}^{P}$.

Definition 8 (HADES Optimization): *Find* $(\mathbf{f}^*, \rho^*, \mathcal{T}^*)$ *solving*

$$\boxed{\min_{\mathbf{f},\rho,\mathcal{T}} \max_{\mu \in \Delta(\mathcal{A}\times\mathcal{E})} \mathbb{E}_{\substack{p\sim w,\ (x,y)\sim\mathcal{D}_p \\ (a,\varepsilon_p)\sim\mu}} \left[\lambda_{\text{obj}} \cdot c\big(\rho(x,\, f_p(x))\big) - R_{a,\varepsilon_p}(f_p)\right]} \tag{9}$$

subject to portfolio floor $\min_{p,a} R_{a,\varepsilon_{\max}}(f_p) \geq \alpha_{portfolio}$ *and per-protocol clean F1 floor* $F1(h_{f_p,\tau}) \geq 1 - \alpha_{clean}$, *where* $\alpha_{portfolio}, \alpha_{clean} \in (0,1)$ *are operator-specified tolerance constants,* $w = (w_p)_{p=1}^{P}$ *are protocol importance weights (Eq. (11)),* $c : \mathcal{K} \to \mathbb{R}_{\geq 0}$ *is per-tier inference cost, and* $\lambda_{obj} > 0$ *has units of F1 per unit cost—the operator's marginal willingness to trade inference cost for robustness (distinct from the false-negative-to-false-positive cost ratio* λ *used to derive the escalation thresholds in Section 4).*

Proposition 4 (Operational Simplification): *Under the meta-adversary budget-concentration argument and Proposition 1, the inner maximization of Eq. (9) reduces to a pure protocol-and-strategy choice, yielding the operational surrogate*

$$\min_{\mathbf{f},\rho,\mathcal{T}} \Big[\lambda_{\text{obj}} \cdot \mathbb{E}\big[c(\rho(x))\big] \;-\; \mathcal{R}_{\text{portfolio}}(\mathbf{f})\Big], \tag{10}$$

subject to the same constraints. This is termed a simplification rather than a reformulation: the reduction uses the cross-protocol budget-concentration argument stated informally in the paragraph preceding Definition 8, which is not itself formalized as part of the minimax problem of Eq. (9). (Definition 1 fixes only the per-input ℓ_∞ *feasibility set and is unrelated to this cross-protocol argument.) Eq. (10) is the tractable objective targeted by* HADES. *The surrogate decomposes into minimizing expected inference cost and maximizing portfolio robustness.*

Since $\mathcal{R}_{\text{portfolio}}$ is non-differentiable, HADES optimizes the traffic-weighted surrogate

$$\sum_{p=1}^{P} w_p\, \mathcal{R}(f_p), \tag{11}$$

where w_p reflects the operator-assigned adversarial importance of protocol p. The portfolio constraint prevents the collapse of the minimum while the surrogate drives up the average, together producing an approximately equalized robustness profile. The circular dependency between ρ and $\mathbf{f}$ is resolved by sequential decomposition: $\mathbf{f}$ is hardened first, then ρ is calibrated post-hoc.

3.5 The Three Pillars

The structure of Eq. (10) implies three distinct sub-problems.

Pillar 1—Stratified Representation. A unified model trained on mixed traffic produces a decision boundary that is a statistical compromise across all $\mathcal{X}_p$, systematically inflating $\Delta_{a_i}(f)$. HADES uses parameter-efficient adaptation [9] to specialize the boundary per protocol: a shared base model is independently warmed up on each protocol's data and then frozen, after which only per-protocol low-rank adapters are updated.

Pillar 2—Game-Theoretic Hardening. Maximizing $\mathcal{R}(f_p)$ requires driving $\min_{a\in\mathcal{A}} R_{a,\varepsilon_{\max}}(f_p)$ upward through exposure to a diverse strategy pool. HADES implements this via a co-evolutionary Red-Blue loop: the Red Agent maintains a bandit distribution over strategies via Exp3 [10], initialized with a domain-knowledge severity prior; the Blue Agent responds with Attack-Success-Rate (ASR)-adaptive adversarial training [11].

Pillar 3—Cost-Optimal Escalation. The routing policy ρ must route each sample to the least-expensive competent tier. HADES implements a Bayes-optimal confidence-gated cascade derived from cost-sensitive classification theory [14]. Given a false-negative-to-false-positive cost ratio λ, the Bayes-optimal classification threshold is

$$\tau^*_{\text{cls}} = \frac{\lambda}{\lambda+1}. \tag{12}$$

A flow is escalated from tier k_i to k_{i+1} if and only if

$$f_p^{(k_i)}(x) \in \left[\frac{1}{\lambda+1}, \frac{\lambda}{\lambda+1}\right) = [\tau_{\text{esc}}, \tau^*_{\text{cls}}), \tag{13}$$

where $\tau_{\text{esc}} = 1/(\lambda+1)$. Flows below τ_{esc} exit as benign; flows at or above τ^*_{cls} exit as attacks; only flows in $[\tau_{\text{esc}}, \tau^*_{\text{cls}})$ are uncertain and deferred to the next tier. The value of λ is an operator choice reflecting the deployment cost structure; its concrete instantiation is given in Section 4.

4 The HADES System Architecture

This section maps each pillar of the simplified objective (Eq. (10)) to its concrete implementation. Fig. 1 provides an overview.

4.1 Pillar 1: LoRA-Stratified Transformer Detector

Each per-protocol detector f_p is a Transformer encoder [24] with $L = 4$ layers, $H = 4$ attention heads, and an embedding dimension $d_{\text{model}} = 64$, operating on the d-dimensional flow feature vector. Training proceeds in two phases.

Phase 1 performs full-model cross-entropy warmup on protocol p's stratified data for $\lfloor \alpha_w \cdot E_p \rfloor$ epochs (warmup fraction $\alpha_w = 0.30$), producing a per-protocol frozen base $\theta^{(p)}_{\text{base}}$ initialized from random weights. Phase 2 freezes $\theta^{(p)}_{\text{base}}$ and trains only the per-protocol LoRA adapters [9] $\{A_p, B_p\}$ with rank r via a combined clean and adversarial cross-entropy loss. The LoRA update rule $W' = W + \frac{\alpha}{r}BA$ introduces a small number of trainable parameters per protocol relative to the frozen base. The concrete hyperparameter values ($r = 8$, $\alpha/r = 2$, $L = 4$, $d_{\text{model}} = 64$) and the resulting parameter counts are given in Section 5.

Note that, for the attention layers, this update is realized as a rank-r residual added to the attention *output* rather than a low-rank factorization of the frozen QKV projections; the distinction and its consequences are detailed in Section 4.2 ("LoRA adapter implementation note").

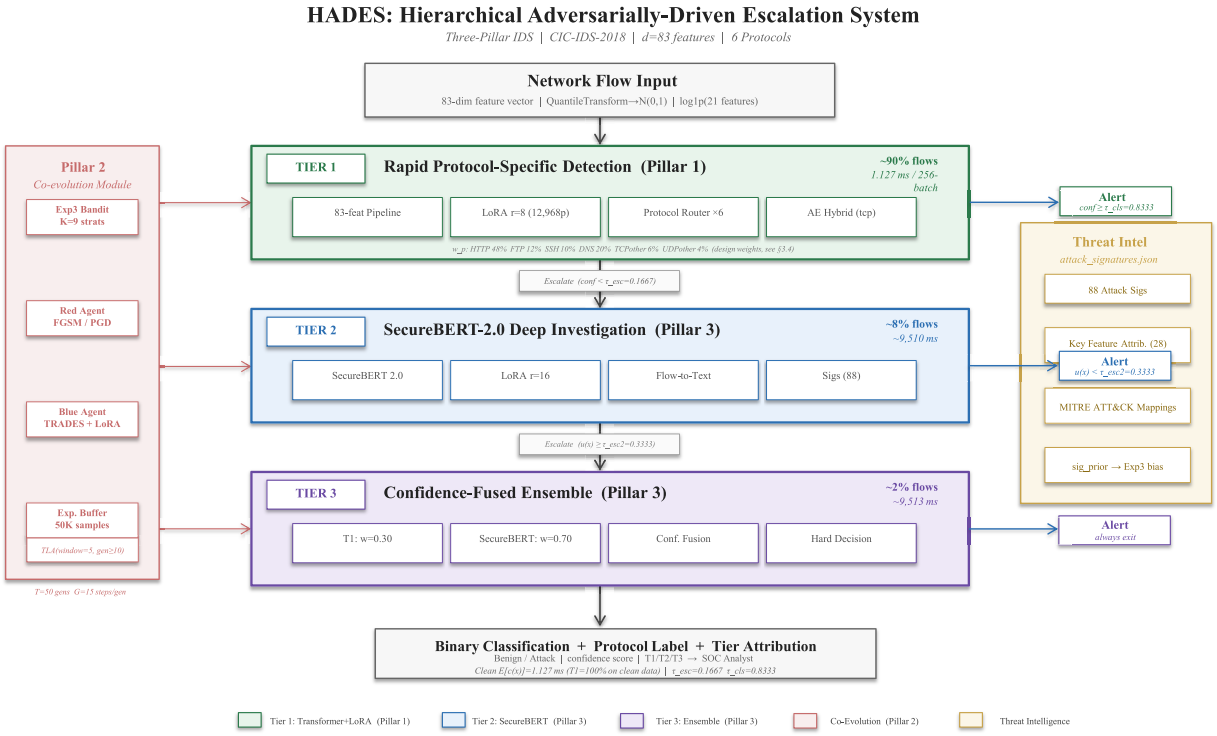


Figure 1: HADES three-pillar architecture. **Left** (dashed red): co-evolution module with Exp3 bandit, red and blue agents, and experience buffer. **Center**: three-tier confidence-gated cascade. **Right** (dashed amber): threat-intelligence module with MITRE ATT&CK severity priors.

For protocol strata that aggregate heterogeneous application-layer traffic, a one-class autoencoder trained on benign flows provides an anomaly score $e(x)$ that can be fused with the classifier score:

$$\tilde{f}_p(x) \;=\; \alpha \cdot f_p(x) \;+\; (1-\alpha)\cdot \mathrm{clip}\left(\frac{e(x)}{\theta_{\mathrm{AE}}},\, 0,\, 1\right), \tag{14}$$

where θ_{AE} is a calibrated reconstruction-error threshold (e.g., the 99.9th percentile on benign validation flows) and $\alpha \in [0,1]$ is a mixing coefficient tuned on validation data. The ablation of this fusion is reported in Section 5.4. Algorithm 1 summarizes both training phases.

Algorithm 1: Two-phase LoRA training for protocol p

Require: Data $(X_{\mathrm{tr}}^p, y_{\mathrm{tr}}^p)$, epochs E_p, warmup fraction α_w, training budget $\varepsilon_{\mathrm{train}}$, LoRA rank r

Note: both phases optimize the cross-entropy of the *soft* network output $g_\theta(X)$ (pre-softmax logits); the hard classifier $h_{f,\tau}$ of Eq. (1) is used only at inference/evaluation time, so no threshold τ enters the training objective and no gradient is taken through an indicator function.

$E_1 \leftarrow \lfloor \alpha_w \cdot E_p \rfloor$; $E_2 \leftarrow E_p - E_1$
Initialize $\theta_{\mathrm{base}}^{(p)}$ (random weights)
for $e = 1, \ldots, E_1$ **do** ▷ Phase 1: full-model warmup
    $\theta_{\mathrm{base}} \leftarrow \theta_{\mathrm{base}} - \nabla_\theta \mathcal{L}_{\mathrm{CE}}(g_\theta(X), y)$ ▷ g_θ: soft logits; no thresholding
end for
Freeze θ_{base}; initialize LoRA adapters $\{A_p, B_p\}$ (rank r)

(Continued)

Algorithm 1 (continued)

for $e = 1, \ldots, E_2$ **do** ▷ Phase 2: LoRA adversarial fine-tuning
  $X^{\text{adv}} \leftarrow a(f_p, X, y, \varepsilon_{\text{train}})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}(f_p(X), y) + \mathcal{L}_{\text{CE}}(f_p(X^{\text{adv}}), y)$
  $\{A_p, B_p\} \leftarrow \{A_p, B_p\} - \nabla_{\{A_p, B_p\}} \mathcal{L}$
end for
return $f_p = \text{TransformerWithLoRA}(\theta_{\text{base}}, A_p, B_p)$

4.2 Pillar 2: Co-Evolutionary Red-Blue Loop

The Red-Blue loop alternates a Red Agent (strategy selector) and a Blue Agent (defender) over T co-evolutionary generations with G gradient steps per generation.

Red Agent. Maintains a weight vector $w_t \in \mathbb{R}^K$ over the K strategies in $\mathcal{A}_{\text{train}}$, updated by Exp3 [10] with a learning rate $\eta = c \cdot \sqrt{\ln K/(KT)}$. The regret-optimal value $c = 1$ attains $O(\sqrt{KT \ln K})$ regret against any fixed comparator in [10] (Thm. 3.1). In practice, $c > 1$ is used, following the finite-horizon convention of [25] to allow faster weight concentration within the short T-generation horizon. With a γ-exploration floor, the pseudo-regret bound acquires an additive γT term [25], yielding $O(\gamma T + c\sqrt{KT \ln K})$. The Exp3 bound holds against adversarial (non-stationary) reward sequences but does not formally account for the two-player coupling with the Blue Agent.

Weights are initialized with a domain-knowledge severity prior derived from attack-signature repositories: strategies associated with high-severity signatures receive elevated initial weights, directing early pressure toward the most dangerous attack vectors. The perturbation budget escalates incrementally from $\varepsilon_{\text{start}}$ to $\varepsilon_{\max}$ whenever the ASR drops below a threshold δ_{ASR} for one consecutive generation. Concrete values of all hyperparameters (T, G, K, γ, c, $\varepsilon_{\text{start}}$, $\varepsilon_{\max}$, δ_{ASR}) are given in Section 5.

Blue Agent. Minimizes the ASR-adaptive TRADES (TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization) objective [11]

$$\mathcal{L}_{\text{TRADES}}(\beta_t) = \mathcal{L}_{\text{CE}}(\mathbf{p}(x), y) + \frac{1}{\beta_t} \cdot \text{KL}\big(\mathbf{p}(x) \,\|\, \mathbf{p}(x^{\text{adv}})\big), \tag{15}$$

where $\mathbf{p}(x) = \text{softmax}(g_p(x))$ and β_t are ASR-adaptive regularization coefficients:

$$\beta_t = \beta_{\max}(1 - \text{ASR}_t) + \beta_{\min}\,\text{ASR}_t, \tag{16}$$

with $\beta_{\max} \geq \beta_{\min} > 0$ operator-specified constants. Note that β_t here is the *inverse* of the regularization coefficient in Zhang et al. [11], where their β multiplies the KL term directly; a smaller β_t here corresponds to stronger regularization. When $\text{ASR}_t = 0$, $1/\beta_t = 1/\beta_{\max}$ (light regularization); when $\text{ASR}_t = 1$, $1/\beta_t = 1/\beta_{\min}$ (maximum pressure). Only LoRA parameters $\{A_p, B_p\}$ are updated. In addition, an entropy-regularization term with coefficient $\lambda_{\text{conf}} = 0.05$ is applied to uncertain-zone flows.

Temporal LoRA Averaging (TLA). Periodically replaces adapter weights with a moving average over recent snapshots, smoothing oscillations in the adversarial dynamics.

LoRA adapter implementation note. For the attention layers, LoRA adapters are applied via a post-hoc forward hook that adds $\frac{1}{3}(\Delta Q + \Delta K + \Delta V)$ to the attention output, where each Δ is a rank-r residual produced by an independent LoRA pair. This is *inspired by* LoRA [9] but is not equivalent to the original formulation: standard LoRA factorizes W_Q, W_K, W_V directly, so the rank-r perturbation propagates through attention scaling and softmax, whereas the hook-based variant adds the averaged residual directly to the

attention output, bypassing the softmax normalization. The resulting adapter is best understood as a rank-r residual on the attention *output*, not as a low-rank factorization of the frozen attention projections. A proper QKV-factored LoRA (requiring injection into `in_proj_weight`) is deferred to future work.

Algorithm 2 gives the complete loop.

Remark 1: *The immune-stop criterion in Algorithm 2 (line 21) serves as a safeguard for shorter training budgets; it is disabled by setting $W_{\min} > T$, ensuring all T generations execute unconditionally. The ε-escalation criterion (line 14) is independent of $W_{\min}$ and triggers whenever ASR drops below δ_{ASR}, so the budget reaches $\varepsilon_{\max} = 0.50$ regardless of the immune-stop setting.*

Algorithm 2: Co-evolutionary red-blue loop for protocol p

Require: Detector f_p, pool $\mathcal{A}_{\text{train}}$ (K strategies), T, G, $\varepsilon_{\text{start}}$, $\varepsilon_{\max}$, γ, δ_{ASR}, W, $W_{\min}$

1: $w_1 \leftarrow$ severity-prior(p) ▷ domain-knowledge init; uniform if no prior available
2: $\varepsilon \leftarrow \varepsilon_{\text{start}}$; $\eta \leftarrow c \cdot \sqrt{\ln K/(KT)}$
3: **for** $t = 1, \ldots, T$ **do**
4: $\quad \pi_t \leftarrow (1-\gamma) \cdot w_t/\|w_t\|_1 + \gamma/K$; $a_t \sim \pi_t$ ▷ Red Agent with γ-exploration floor
5: $\quad$ **for** $g = 1, \ldots, G$ **do** ▷ Blue Agent
6: $\quad\quad X^{\text{adv}} \leftarrow a_t(f_p, X, y, \varepsilon)$
7: $\quad\quad$ Update $\{A_p, B_p\}$ via $\nabla \mathcal{L}_{\text{TRADES}}(\beta_t)$ (15)
8: $\quad$ **end for**
9: $\quad \text{ASR}_t \leftarrow \text{evaluate}(a_t, f_p^{(t\text{-pre})})$ ▷ pre-update model; one-generation lag
10: $\quad \hat{r}_{t,i} \leftarrow \text{ASR}_t \cdot \mathbf{1}[i = a_t]/\pi_t(a_t)$
11: $\quad w_{t+1,i} \leftarrow w_{t,i} \cdot \exp(\eta \hat{r}_{t,i}) \quad \forall i$
12: $\quad$ **if** $\text{ASR}_t < \delta_{\text{ASR}}$ for 1 consecutive gen **then**
13: $\quad\quad \varepsilon \leftarrow \min(\varepsilon + \Delta_\varepsilon, \varepsilon_{\max})$ ▷ ε-escalation independent of $W_{\min}$
14: $\quad$ **end if**
15: $\quad$ **if** $t \geq t_{\text{TLA}}$ **and** $t \bmod \tau_{\text{TLA}} = 0$ **then**
16: $\quad \{A_p, B_p\} \leftarrow \text{TLA}(f_p^{(t-w_{\text{TLA}}+1:t)})$
17: $\quad$ **end if**
18: $\quad$ **if** $\max_a \text{ASR}(a, f_p) < \delta_{\text{ASR}}$ for W gens **and** $t \geq W_{\min}$ **then**
19: $\quad\quad$ **break** ▷ Immune stop
20: $\quad$ **end if**
21: **end for**
22: **return** hardened f_p

4.3 Pillar 3: Cost-Optimal Confidence-Gated Cascade

The cascade implements ρ as a sequential confidence gate governed by the Bayes-optimal thresholds derived in Eqs. (12) and (13). Given a false-negative-to-false-positive cost ratio λ:

- $\tau^*_{\text{cls}} = \lambda/(\lambda+1)$ is the Bayes-optimal attack decision threshold [14];
- $\tau_{\text{esc}} = 1/(\lambda+1)$ is the Tier-1→Tier-2 escalation lower bound;
- $\tau_{\text{esc2}} = 2/(\lambda+1)$ narrows the uncertain zone for the stronger Tier-2 classifier; only flows with $s_2 \in [\tau_{\text{esc2}}, \tau^*_{\text{cls}})$ proceed to Tier 3.

Table 2 collects all threshold, robustness, and cost symbols used across Sections 3–5 in one place, since the global threshold (τ_{esc}, τ^*_{cls}, τ_{esc2}), the per-protocol calibrated threshold ($\hat{\tau}^*_p$), and the attack/cost parameters are otherwise introduced across several sections.

Table 2: Notation summary: thresholds, robustness measures, and cost parameters.

Symbol	Meaning	Defined in
τ^*_{cls}	Global Bayes-optimal attack threshold, $\lambda/(\lambda{+}1)$	Eq. (12)
τ_{esc}	Global Tier-1→Tier-2 escalation floor, $1/(\lambda{+}1)$	Eq. (13)
τ_{esc2}	Tier-2→Tier-3 escalation floor, $2/(\lambda{+}1)$	Section 4.3
$\hat{\tau}^*_p$	Per-protocol threshold, calibrated on validation split	Section 5
λ	FN:FP cost ratio (deployed default 5)	Section 4.3
Φ_{fp}	Uniform nine-strategy mean robustness for protocol p (fixed pool, equal weights)	Eq. (6)
Φ^{Exp3}	Exp3-weighted robustness; training diagnostic only, not an evaluation metric	Eq. (7)
$\mathcal{R}(f_p) = \min_a R_a(f_p)$	True worst-case robustness	Def. 4
$\Delta_{a_i}(f_p)$	Strategy-distribution gap	Prop. 3
ε	ℓ_∞ perturbation budget	Section 5
γ	Exp3 exploration floor	Section 4.2
c	Exp3 rate amplifier (deployed default 3)	Section 4.2
δ_{ASR}	ε-escalation ASR trigger	Section 4.2
$\beta_{\max}, \beta_{\min}$	TRADES regularization bounds	Eq. (15)
$\mathbb{E}[c(x)]$	Expected per-flow inference cost	Section 5.3

The value $\lambda = 5$ used in this work reflects the operational judgment that a missed attack incurs five times the cost of a false alarm, a conservative but widely-used ratio in security-sensitive deployments [14]. Sensitivity to λ is reported in Section 5: F1 remains above 0.9991 and escalation below 0.27% over $\lambda \in [1, 20]$.

Algorithm 3 gives the complete routing logic.

Algorithm 3: HADES confidence-gated inference

Require: Flow x, protocol p, $f_p^{(1)}$, $f_p^{(2)}$, τ_{esc}, τ^*_{cls}, τ_{esc2}, ensemble weights w_1, w_2

$s_1 \leftarrow f_p^{(1)}(x)$
if $s_1 \geq \tau^*_{\text{cls}}$ **then**
  return (Attack, Tier 1) ▷ confident attack
else if $s_1 < \tau_{\text{esc}}$ **then**
  return (Benign, Tier 1) ▷ confident benign
else ▷ $s_1 \in [\tau_{\text{esc}}, \tau^*_{\text{cls}})$: uncertain, escalate
  $s_2 \leftarrow f_p^{(2)}(x)$
  if $s_2 \geq \tau^*_{\text{cls}}$ **then**
    return (Attack, Tier 2)
  else if $s_2 < \tau_{\text{esc2}}$ **then**
    return (Benign, Tier 2)
  else ▷ $s_2 \in [\tau_{\text{esc2}}, \tau^*_{\text{cls}})$: escalate
    $\hat{y} \leftarrow \mathbf{1}[w_1 s_1 + w_2 s_2 \geq 0.5]$
    return $(\hat{y}, \text{Tier } 3)$
  end if
end if

The three tiers are:

Tier 1 (Fast path): the LoRA Transformer Detector processes all incoming flows. Flows with $f_p(x) < \tau_{\text{esc}}$ exit as benign; flows with $f_p(x) \geq \tau^*_{\text{cls}}$ exit as attacks; the remainder (uncertain zone $[\tau_{\text{esc}}, \tau^*_{\text{cls}})$) escalate to Tier 2.

Tier 2 (Reasoning path): a fine-tuned language model [8] with per-protocol LoRA adapters classifies the uncertain flows from a natural-language serialization of the top-k flow features (selected by mutual information on the training split). The serialization template lists feature name-value pairs in a fixed order, truncated to the model's token budget. The serialization is fixed prior to fine-tuning and is frozen throughout training and evaluation.

Tier 3 (Final arbiter): the weighted ensemble

$$\hat{y} \;=\; \mathbf{1}\big[w_1 f_p(x) + w_2 f_{\mathrm{LM}}(x) \geq 0.5\big] \tag{17}$$

combines both tiers for flows uncertain at both preceding stages, where $w_1 + w_2 = 1$ and w_2 is calibrated on the validation split. In practice, Tier 3 receives only a negligible fraction of flows; the cascade effectively operates as a two-tier system under the evaluated conditions.

The total expected inference cost is $\mathbb{E}[c(\rho(x))] = p_1 c_1 + p_2 c_2 + p_3 c_3$, where p_k is the fraction of flows routed to tier k and c_k is the per-tier cost. Measured latencies and routing distributions under both clean and adversarial inputs are reported in Section 5.

Remark 2: *The T2 escalation threshold τ_{esc2} is calibrated on the validation split when sufficient escalated flows are available; otherwise, the analytically derived value $2/(\lambda+1)$ serves as a principled fallback, as the absence of validation data in the uncertain zone means the calibration sweep has no signal to optimize over. Tier-2 ensemble weights w_1, w_2 are likewise calibrated on the validation split; routing distributions and per-tier adversarial F1 are reported in Section 5.3.*

5 Experiments

Dataset. All experiments use CIC-IDS-2018 [26]. Table 3 summarizes the per-protocol statistics. Of the 63.2M total flows, 58.5M unlabeled background flows are excluded and benign flows are capped at 150K per stratum, yielding 4,741,659 labeled flows across $P = 6$ protocol strata. The two catch-all strata (`TCP_OTHER`, `UDP_OTHER`) group attack families whose application-layer protocols are not fixed; their internal heterogeneity motivates the auxiliary autoencoder in `TCP_OTHER`. Features are preprocessed with log1p transformation, Quantile Transformer normalization is set to $\mathcal{N}(0,1)$, and SMOTE balancing is applied [27]. Splits are protocol-stratified 70/15/15, yielding 3,319,159 training, 711,249 validation, and 711,251 test flows. Two limitations apply throughout: CICFlowMeter label noise may inflate reported F1 [22], and the controlled lab environment produces clean F1 values (≥0.994) not representative of production; all robustness claims are within-distribution.

Table 3: CIC-IDS-2018 statistics after protocol partitioning and 70/15/15 stratified split. Attack ratios are before SMOTE.

Protocol	Attack Families	Attack	Benign	Total	Atk. Ratio
HTTP	DoS Hulk/GoldenEye/Slowloris, Web Brute-Force/XSS/SQL, Infiltration	1,931,006	150,000	2,081,006	92.8%
DNS	DDoS-HOIC	1,082,293	150,000	1,232,293	87.8%
SSH	SSH-BruteForce	94,197	150,000	244,197	38.6%
FTP	FTP-BruteForce	298,874	150,000	448,874	66.6%
TCP_OTHER	Botnet Ares C&C	143,183	150,000	293,183	48.8%
UDP_OTHER	DDoS-LOIC-HTTP, DDoS-LOIC-UDP	292,106	150,000	442,106	66.1%
Total		3,841,659	900,000	4,741,659	81.0%

Why CIC-IDS-2018, given that more recent datasets exist. CIC-IDS-2018 is deliberately retained as the primary benchmark for four reasons specific to what this paper measures, and its age is treated as a scoping decision rather than an oversight. (i) *Per-stratum volume*. Pillar 1 requires each protocol subspace to carry enough labelled flows to train and adversarially harden a dedicated detector, and Pillar 2 multiplies that requirement by $T \times G$ co-evolutionary gradient steps. CIC-IDS-2018 supplies 4,741,659 labelled flows spread over six protocol strata, with no stratum below 244K flows (Table 3); most newer public corpora are either substantially smaller after protocol partitioning or concentrate their volume in one or two application-layer protocols, which would collapse the stratification that Pillar 1 exists to test. (ii) *Comparability with the adversarial-robustness baselines*. The systems this work is positioned against—GADoT [4], EAT [5], the ensemble defense of [6], and the hierarchical detectors [3,7]—report on CIC-IDS-2017/2018. Moving to a newer corpus would remove the only common ground on which even the indicative comparison of table (in Section 5.8) can be drawn. (iii) *A matched-extractor shift study*. Section 5.7 decomposes cross-dataset degradation into traffic-distribution shift and normalizer mismatch. That decomposition is only interpretable when both corpora are produced by the same feature extractor; CIC-IDS-2017 and CIC-IDS-2018 are both CICFlowMeter-derived, which makes the $\Delta_{\text{dist}}/\Delta_{\text{norm}}$ split meaningful rather than an artifact of differing feature definitions. (iv) *Auditability*. The dataset's known pathologies are documented in the literature [22], which allowed the integrity audit of Section 5.6 (duplication, temporal drift, mutual-information leakage, SMOTE) to be performed and reported rather than assumed away; that audit is itself a contribution of the revised manuscript, and it is precisely what shows how much of the near-ceiling F1 is a dataset property. The cost of this choice is stated explicitly: results on CIC-IDS-2018 do not transfer to modern encrypted, IoT, or 5G/6G traffic—indeed Section 5.7 shows they do not even transfer to CIC-IDS-2017—and validation on contemporary corpora (CIC-IoT-2023, UNSW-NB15, and successor CIC releases) is stated as a required next step in Section 6, item 2 and item 10, not as an already-solved extension.

Threat model. Nine attack algorithms constitute the evaluation pool $\mathcal{A}_{\text{eval}}$, divided into three categories: (a) *Gradient-based (white-box):* FGSM [28], PGD-k [29] ($k \in \{7, 10, 20, 40\}$), MI-FGSM [30], and APGD-CE [31], (b) *Optimization-based (white-box):* C&W-ℓ_2, projected onto the ℓ_∞ ball post-generation to maintain compatibility with Definition 1, and (c) *Query-based (black-box):* Square Attack [32] at 1000 queries. Evaluations are reported at $\varepsilon = 0.1$ (full 6×9 robustness matrix) and $\varepsilon = 0.3$ (per-protocol adversarial F1).

All attacks are generated in continuous $\mathbb{R}^d$ under an ℓ_∞ budget without enforcing per-feature flow constraints (non-negativity, integer packet counts, duration–rate consistency, protocol-valid ranges). The reported adversarial F1 therefore quantifies *mathematical* robustness within the ℓ_∞ ball, not robustness to realizable adversarial traffic; a feasibility-projected evaluation is reported in Section 5.5.

The training pool $\mathcal{A}_{\text{train}}$ contains nine white-box FGSM/PGD variants only (1× and 2× nominal ε). MI-FGSM, C&W-ℓ_2, APGD-CE, and Square Attack are absent from training and constitute the *unseen-attack generalization test*.

Hardware. All experiments ran on a single machine with an Intel Core i9-10900X CPU (3.7 GHz, 10 cores), 64 GB DDR4 RAM, and an NVIDIA TITAN Xp GPU (12 GB VRAM). Latency measurements are wall-clock batch-level timings averaged over 100 runs after 20 warm-up passes.

Hyperparameter selection. The warm-up fraction $\alpha_w = 0.30$ and the $T = 50/G = 15$ budget were fixed by the 14-h single-GPU (TITAN Xp) training envelope; the severity prior used to initialize Exp3 derives from attack-signature repositories; $c = 3$ follows the finite-horizon Exp3 convention of [25]; the TRADES coefficients ($\beta_{\max} = 6$, $\beta_{\min} = 1$) bracket the light-to-strong regularization range; and the escalation thresholds follow analytically from λ (Section 4). Sensitivity to the LoRA rank (Table 4) and to λ (Table 5) is reported below.

Table 4: LoRA rank sensitivity. All six detectors retrained end-to-end per rank; robustness Φ is the nine-strategy mean at $\varepsilon = 0.1$.

Rank r	Trainable LoRA Params	Fraction of Base	$\Phi(\varepsilon = 0.1)$
4	6484	4.4%	0.9999
8 (deployed)	12,968	8.8%	0.9998
16	25,936	17.5%	1.0000
32	51,872	35.0%	0.9999

Table 5: Cost-ratio sensitivity. Bayes-optimal thresholds are closed-form in λ; deployed Tier-1 scores are re-routed per row. Means over the six protocols.

λ	τ_{esc}	τ^*_{cls}	F1	FPR	FNR	Esc. %	$\mathbb{E}[c]$ (ms)
1	0.5000	0.5000	1.0000	0.0002	0.0000	0.000	1.42
3	0.2500	0.7500	0.9999	0.0000	0.0002	0.022	4.04
5	0.1667	0.8333	0.9998	0.0000	0.0004	0.042	6.42
10	0.0909	0.9091	0.9994	0.0000	0.0012	0.128	16.77
20	0.0476	0.9524	0.9991	0.0000	0.0018	0.265	33.20

Implementation details.

- **Pillar 1—LoRA-Stratified Transformer Detector.** Each detector has $L = 4$ layers, $H = 4$ attention heads, an embedding dimension of $d_{\text{model}} = 64$, operating on $d = 83$ features. LoRA rank $r = 8$ with scaling $\alpha/r = 2$ yields 12,968 trainable parameters per protocol (8.8% of the 148,034 frozen base). Warmup fraction $\alpha_w = 0.30$.
- **Pillar 2—Co-Evolutionary Red-Blue Loop.** $T = 50$ generations, $G = 15$ gradient steps per generation. The perturbation budget starts at $\varepsilon_{\text{start}} = 0.20$ and escalates by 0.10 whenever the ASR drops below $\delta_{\text{ASR}} = 0.05$, reaching $\varepsilon_{\max} = 0.50$. The Exp3 bandit uses an exploration floor of $\gamma = 0.03$ and rate amplification of $c = 3$. The TRADES loss uses $\beta_{\max} = 6$, $\beta_{\min} = 1$. Temporal LoRA Averaging is applied every 5 generations from generation 10, averaging the last 5 adapter snapshots. The immune-stop criterion is disabled; all $T = 50$ generations execute unconditionally.
- **Pillar 3—Confidence-Gated Cascade.** Cost ratio $\lambda = 5$ yields Bayes-optimal thresholds $\tau_{\text{esc}} = 0.1667$ and $\tau^*_{\text{cls}} = 0.8333$ (Section 4). The Tier-2 lower gate $\tau_{\text{esc2}} = 0.3333$ is a design choice (not Bayes-optimal). All thresholds are calibrated on the validation split and frozen before any contact with the test split. Low per-protocol values (e.g., SSH: $\hat{\tau}^*_p = 0.050$) arise from the recall-floor constraint $\text{Recall}(\hat{\tau}) \geq 0.90$ under high attack ratios. Tier-2 uses the top-20 features by mutual information (training split), serialized to 512 sub-word tokens; ensemble weights are $w_1 = 0.30$, $w_2 = 0.70$. The Tier-2 SecureBERT adapter uses LoRA rank 16.
- **Evaluation and statistical testing.** Robustness matrix evaluations use 2000 randomly sampled test flows per protocol per attack. Bootstrap confidence intervals (95%, $N_{\text{boot}} = 1000$ resamples) use the full per-protocol test splits ($N \in [36{,}630,\ 312{,}151]$); maximum CI width is 0.0007. Paired bootstrap tests (two-sided, $\alpha = 0.05$, Bonferroni-corrected across 6 protocols) are used for all baseline comparisons.

Training cost. Co-evolutionary hardening requires $T \times G \times P = 50 \times 15 \times 6 = 4500$ Blue-Agent gradient steps per strategy slot, totaling approximately 14 h on the TITAN Xp across all six protocols. This one-time cost is amortized over the detector's deployment lifetime; inference cost remains $\mathbb{E}[c(x)] = 5.4$ ms on clean traffic.

As shown in Table 4, robustness is flat in rank (range ≤ 0.0002 across an 8× capacity change): the adaptation task is not capacity-limited at this scale, and $r = 8$ is retained as a conservative default.

The routing policy, as depicted in Table 5, degrades gracefully under cost-ratio misspecification, and F1 stays above 0.9991 over a 20× range. The measured directions are as follows: FPR falls (from 0.0002 at $\lambda = 1$ to 0.0000 for $\lambda \geq 3$), while FNR, the escalation rate, and $\mathbb{E}[c]$ all rise monotonically in λ (FNR 0.0000 → 0.0018; $\mathbb{E}[c]$1.42 → 33.20 ms). This is the expected consequence of the convention of Eq. (12): increasing λ raises the attack-decision threshold $\tau^*_{\text{cls}} = \lambda/(\lambda+1)$ and simultaneously lowers $\tau_{\text{esc}} = 1/(\lambda+1)$, so the uncertain zone widens, more flows are deferred to Tier 2/Tier 3 for deeper inspection, and the residual flows that still exit at Tier 1 do so under a stricter attack criterion—which converts a small number of false alarms into an equally small number of Tier-1 misses. Both effects are bounded in absolute terms (FNR ≤ 0.0018, FPR ≤ 0.0002) over the whole sweep. As deployment guidance, low λ (1–3) minimizes latency and escalation load and is appropriate for enterprise monitoring where analyst capacity is the binding constraint; $\lambda = 5$ is the balanced default deployed here; high λ (10–20) maximizes the fraction of ambiguous traffic sent to deep inspection—appropriate for critical-infrastructure and IoT settings where an unexamined suspicious flow is costlier than compute—at 33.20 ms mean cost per flow, and should be paired with the recall-floor calibration of the per-protocol thresholds $\hat{\tau}^*_p$ (Section 5) if minimizing Tier-1 misses is the primary objective.

5.1 Pillar 1: Do Protocol-Stratified Boundaries Reduce the Robustness Gap?

Setup. This experiment compares a unified Transformer Detector (single shared weights, no LoRA) against HADES Pillar 1 (LoRA-stratified). The stratification advantage is $\Delta\Phi = \Phi_{\text{strat}} - \Phi_{\text{unif}}$.

Results. All six per-protocol clean F1 values exceed 0.99 (Table 6, Fig. 2).

Table 7 shows a mean confounded advantage $\Delta\Phi = +0.043$—this conflates evaluation budget, attack pool, and training regime differences and should be read as an upper bound on the pure stratification effect.

The controlled ablation (Table 8 and Fig. 3), which matches all three confounds, provides a clean mean $\Delta\varphi = +0.0003$; no protocol reaches significance at $\alpha = 0.05$, and under matched conditions the per-protocol effects are uniformly small and of both signs (largest positive `SSH` +0.0011; `TCP_OTHER` −0.0008). The large gains on the structurally heterogeneous strata (`TCP_OTHER` +0.21, `SSH` +0.03) belong to the *confounded* comparison of Table 7, not to this matched-condition ablation, and are therefore an upper bound that conflates stratification with the evaluation budget, attack pool, and training regime. The honest reading is that stratification is where the unified model is forced into a distributional compromise—but that, once the three confounds are removed, the measured effect at $\varepsilon = 0.1$ is within noise.

Table 6: Per-protocol clean and adversarial F1 with 95% bootstrap CIs (APGD-CE, $\varepsilon = 0.1$, post-co-evolutionary hardening, $N = 1000$ resamples). Maximum CI width: 0.0007.

Protocol	Clean F1	95% CI	Adv F1	95% CI
HTTP	0.9999	[0.9999, 0.9999]	0.9998	[0.9997, 0.9998]
DNS	1.0000	[1.0000, 1.0000]	1.0000	[1.0000, 1.0000]
SSH	1.0000	[1.0000, 1.0000]	1.0000	[1.0000, 1.0000]
FTP	1.0000	[1.0000, 1.0000]	1.0000	[1.0000, 1.0000]
TCP_OTHER	1.0000	[0.9999, 1.0000]	0.9987	[0.9984, 0.9990]
UDP_OTHER	1.0000	[1.0000, 1.0000]	1.0000	[1.0000, 1.0000]

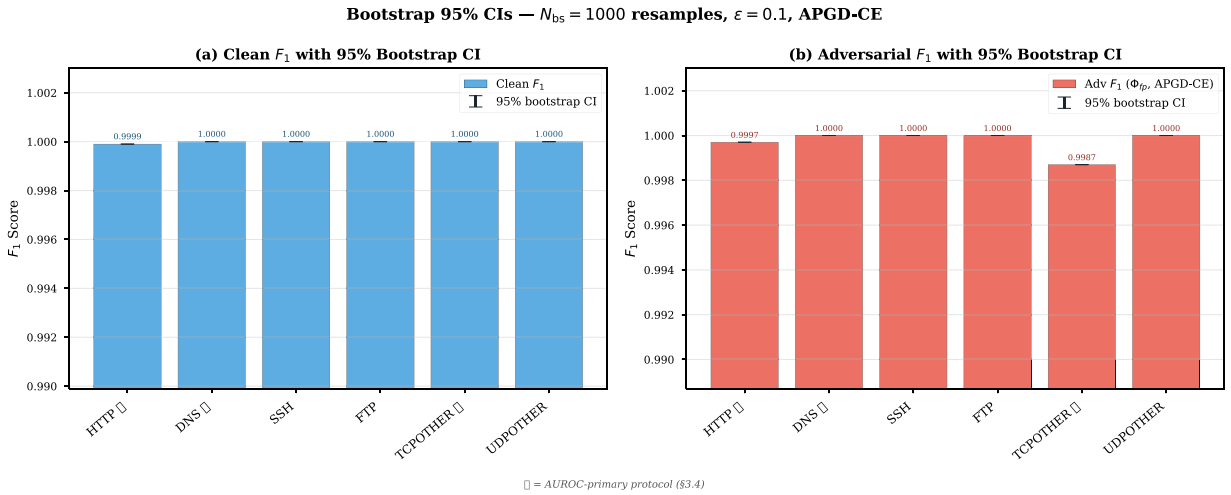


Figure 2: 95% bootstrap CIs (N = 1000) for per-protocol clean and adversarial F1 (APGD-CE, ε = 0.1). Maximum CI width: 0.0007 (adversarial F1, TCP_OTHER).

Table 7: Pillar 1: stratified vs. unified robustness (CIC-IDS-2018). Φ_{strat}: uniform mean F1 over the nine-strategy pool, ε = 0.1 (co-evolved). Φ_{unif}: FGSM F1 of unified baseline at ε = 0.3. $\Delta\Phi$ is an upper bound on the pure stratification effect due to three simultaneous confounds.

Protocol	F_1 Strat	F_1 Unif	ΔF_1	Φ_{strat}	Φ_{unif}	$\Delta\Phi$
HTTP	0.9999	0.9998	+0.0001	0.9994	0.9946	+0.0048
DNS	1.0000	1.0000	+0.0000	1.0000	0.9975	+0.0025
SSH	1.0000	0.9998	+0.0002	1.0000	0.9700	+0.0300
FTP	1.0000	1.0000	+0.0000	1.0000	0.9982	+0.0018
TCP_OTHER	1.0000	0.9993	+0.0007	0.9994	0.7908	+0.2086
UDP_OTHER	1.0000	0.9999	+0.0001	1.0000	0.9912	+0.0088
Mean	—	—	—	0.9998	0.9571	+0.0428

Table 8: Pillar 1 controlled ablation: stratified vs. unified at matched conditions (ε = 0.1, uniform nine-strategy mean, static FGSM training for both). $\Delta\varphi$ isolates the pure LoRA-stratification effect.

Protocol	φ_{strat}	φ_{unif}	$\Delta\varphi$	$p < 0.05$
HTTP	1.0000	0.9992	+0.0008	No
DNS	1.0000	0.9997	+0.0003	No
SSH	1.0000	0.9989	+0.0011	No
FTP	1.0000	1.0000	+0.0000	No
TCP_OTHER	0.9992	1.0000	−0.0008	No
UDP_OTHER	1.0000	0.9997	+0.0003	No
Mean	1.0000	0.9996	+0.0003	—

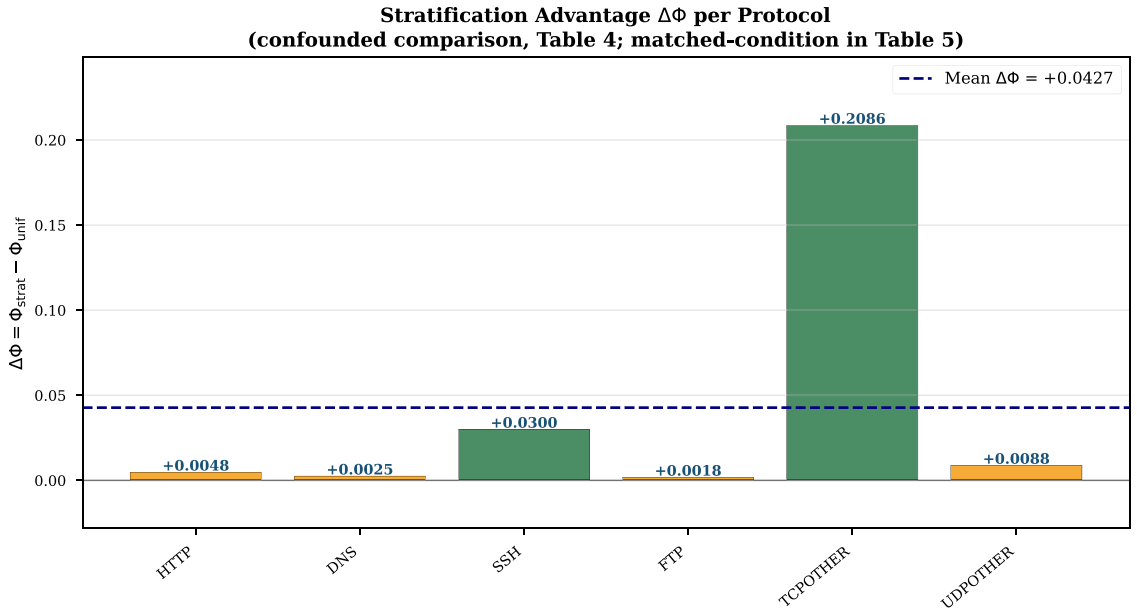


Figure 3: Stratification advantage ΔΦ per protocol (confounded comparison, Table 7). Strongest gains on TCP_OTHER (+0.21) and SSH (+0.03); near-parity on DNS, FTP, and UDP_OTHER. The matched-condition effect is in Table 8.

In Table 9, MCC (≥0.9986) and balanced accuracy (≥0.9994) confirm that the F1 values are not artifacts of class imbalance; the expected calibration error remains ≤0.0106, which is a prerequisite for confidence-gated escalation.

Table 9: Class-imbalance-robust metrics at the calibrated per-protocol operating points (clean test data).

Protocol	τ	F1	MCC	Bal. Acc.	FPR	ECE
HTTP	0.3750	0.9999	0.9986	0.9994	0.0011	0.0091
DNS	0.8333	1.0000	0.9999	1.0000	0.0000	0.0007
SSH	0.0500	1.0000	1.0000	1.0000	0.0000	0.0018
FTP	0.8333	1.0000	1.0000	1.0000	0.0000	0.0022
TCP_OTHER	0.1850	0.9999	0.9999	0.9999	0.0001	0.0106
UDP_OTHER	0.8333	1.0000	1.0000	1.0000	0.0000	0.0092

Fig. 4a compares clean vs. adversarial F1 (post co-evolution, with $\varepsilon = 0.3$) and shows near-perfect clean F1 across protocols, while adversarial F1 remains high for most protocols but exhibits noticeable drops for some protocols (e.g., HTTP and TCP_OTHER), highlighting where attacks still stress the detector. Fig. 4b contrasts the nine-strategy mean robustness (over 9 strategies) with the explicit worst-case (minimum over strategies) at $\varepsilon = 0.1$, and the curves are essentially indistinguishable and extremely close to 1.0 across all listed protocols, indicating that the strategy mixture is effectively tracking the worst-case robustness in this setting. Finally, Fig. 4c reports per-protocol AUROC = 1.0 for every protocol.

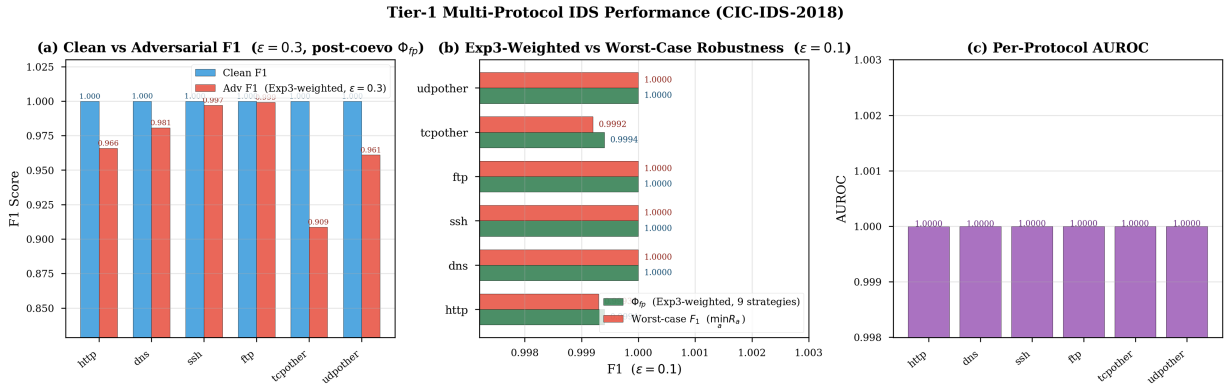


Figure 4: Tier-1 per-protocol performance. (**a**) Clean vs. adversarial F1 at $\varepsilon = 0.3$. (**b**) Robustness ratio. (**c**) AUROC.

5.2 Pillar 2: Does Co-Evolution Achieve Strategy-Distribution Robustness?

Setup. Three training regimes are compared on the same LoRA architecture and frozen base: (a) static FGSM training ($\varepsilon_{\text{train}} = 0.05$); (b) EAT [5] with a fixed PGD curriculum at $\varepsilon = 0.30$; and (c) HADES Pillar 2 (Exp3 Red-Blue loop, $T = 50$, $G = 15$). The primary metric is per-protocol uniform nine-strategy mean robustness Φ_{fp}; the true worst-case $\mathcal{R}(f_p) = \min_a R_a(f_p)$ is reported separately.

Results at $\varepsilon = 0.1$. Table 10 and Fig. 5 show that co-evolution achieves a mean $\bar{\varphi} = 0.9998$ and portfolio robustness $\mathcal{R}_{\text{portfolio}} = 99.92\%$. The +0.3461 advantage over static FGSM is primarily attributable to the wider training budget ($\varepsilon_{\max} = 0.50$ vs. $\varepsilon_{\text{train}} = 0.05$): static training leaves large strategy-distribution gaps (e.g., `DNS`: $\Phi = 0.592$, `TCP_OTHER`: $\Phi = 0.405$) that co-evolution closes across all nine strategies. Co-evolution and EAT are near-parity at this budget ($\Delta = -0.0001$), as expected when both are evaluated well below their training budgets.

The reported mixture robustness is not optimistic relative to the worst-case guarantee (Table 11): on the deployed detectors at $\varepsilon = 0.1$, the gap $\Phi_{\text{fp}} - \min_a R_a(f_p)$ is at most 0.0002 across all six strata (0.00015 on HTTP, 0.00011 on TCP_OTHER; 0.0000 elsewhere).

Table 10: Pillar 2: co-evolutionary vs. static vs. EAT ($\varepsilon = 0.1$, uniform mean F1 over the nine-strategy pool).

Protocol	Φ Co-evo	Φ Static	Φ EAT	Δ (Co-st)	Δ(Co-EAT)
DNS	1.0000	0.5923	1.0000	+0.4077	+0.0000
FTP	1.0000	0.8597	1.0000	+0.1403	+0.0000
HTTP	0.9994	0.5208	0.9997	+0.4786	−0.0003
SSH	1.0000	0.8151	1.0000	+0.1849	+0.0000
TCP_OTHER	0.9994	0.4046	1.0000	+0.5948	−0.0006
UDP_OTHER	1.0000	0.7299	1.0000	+0.2701	+0.0000
Mean	0.9998	0.6537	1.0000	+0.3461	−0.0002

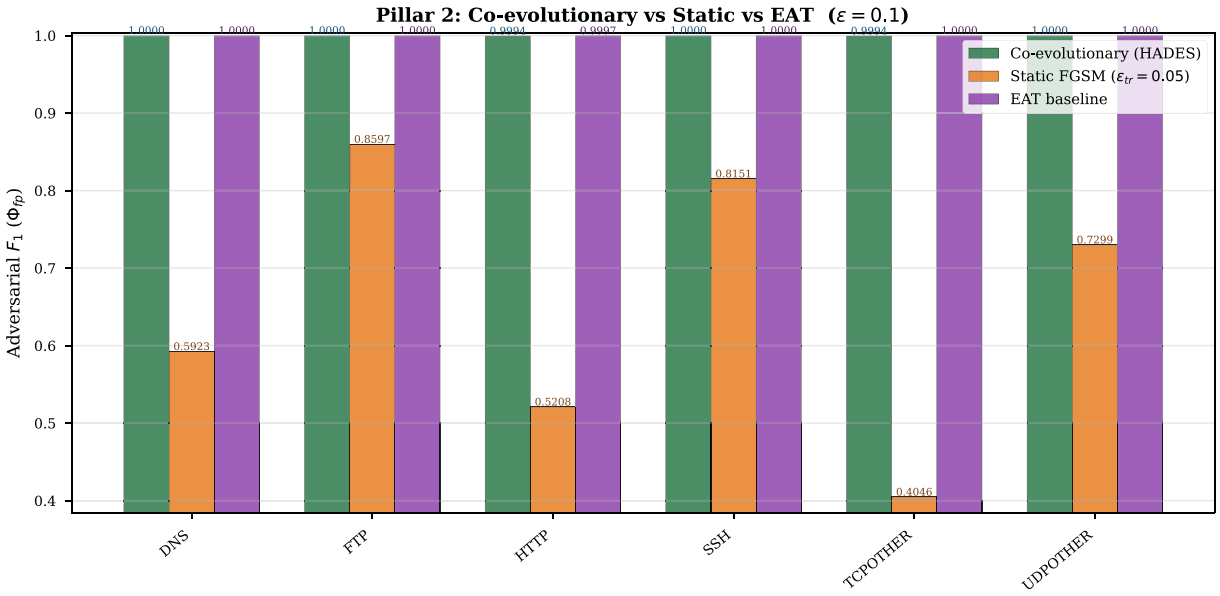


Figure 5: Pillar 2: per-protocol nine-strategy mean robustness Φ_{fp} at $\varepsilon = 0.1$ for co-evolutionary (HADES), static FGSM ($\varepsilon_{\text{train}} = 0.05$), and EAT baselines. Co-evolution closes gaps exceeding +0.59 on TCP_OTHER and +0.41 on DNS that static training leaves open.

Table 11: Mixture robustness vs. explicit worst case on the deployed detectors ($\varepsilon = 0.1$, nine-strategy pool). The gap $\Phi_{\text{fp}} - \min_a R_a(f_p)$ is at most 0.0002 on every stratum.

Protocol	Φ_{fp}	$\min_a R_a(f_p)$	Gap
HTTP	0.9993	0.9992	0.00015
DNS	1.0000	1.0000	0.00000
SSH	1.0000	1.0000	0.00000
FTP	1.0000	1.0000	0.00000
TCP_OTHER	0.9996	0.9995	0.00011
UDP_OTHER	1.0000	1.0000	0.00000

Furthermore, the hardened per-protocol worst-case profile $\{\mathcal{R}(f_p)\}$ at $\varepsilon = 0.1$ spans $[0.9992, 1.0000]$ (the $\min_a R_a(f_p)$ column of Table 11); since any traffic-weighted average $\sum_p w_p \mathcal{R}(f_p)$ lies within this range, the gap between the surrogate of Eq. (10) and the portfolio minimum is bounded by 0.0008, confirming approximate equalization (The corresponding pool-average profile $\{\Phi_{\text{fp}}(f_p)\}$ spans the slightly narrower $[0.9993, 1.0000]$; the bound above is stated on $\mathcal{R}$, which is the quantity entering Eqs. (8) and (11)).

Results at high budgets. Table 12 reports the comparison at $\varepsilon \in \{0.3, 0.5\}$. At $\varepsilon = 0.5$, co-evolution maintains $\bar{\varphi} = 0.626$ while EAT (trained at $\varepsilon = 0.30$) degrades to 0.433 ($\Delta = +0.193$); the gap reflects both adaptive curriculum and budget mismatch. A budget-matched EAT variant trained at $\varepsilon = 0.50$ and a uniform Red Agent ablation are reported in Table 13. Fig. 6 visualizes the pre- vs. post-co-evolution F1 gain per protocol at $\varepsilon = 0.3$.

Table 12: Pillar 2 at high budgets ($\varepsilon \in \{0.3, 0.5\}$). Co-evolution trained at $\varepsilon_{\max} = 0.50$; static at $\varepsilon_{\text{train}} = 0.05$; EAT trained at $\varepsilon = 0.30$.

Protocol	$\varepsilon = 0.3$			$\varepsilon = 0.5$		
	Co-evo	Static	EAT	Co-evo	Static	EAT
HTTP	0.9659	0.6142	0.5873	0.5764	0.3354	0.3044
DNS	0.9807	0.8258	0.8327	0.4616	0.3192	0.3201
SSH	0.9970	0.9917	0.9923	0.7816	0.5880	0.5636
FTP	0.9992	0.9558	0.9403	0.9321	0.7390	0.6294
TCP_OTHER	0.9086	0.3955	0.4388	0.4506	0.2550	0.2803
UDP_OTHER	0.9610	0.9616	0.9552	0.5519	0.5263	0.5006
Mean	0.9687	0.7908	0.7911	0.6257	0.4605	0.4331
$\Delta_{\text{co-st}}$		+0.178			+0.165	
$\Delta_{\text{co-EAT}}$		+0.178			+0.193	

Table 13: Uniform Red Agent ablation and budget-matched baselines at $\varepsilon = 0.5$. *Uniform*: $\pi_t = \mathbf{1}/K$ (no Exp3 feedback). *Static-0.5* and *EAT-0.5*: trained at $\varepsilon = 0.50$. † DNS Static-0.5 did not collapse in this run (F1 = 0.254); collapse was observed in an earlier run and remains a risk under direct $\varepsilon = 0.50$ training.

Protocol	Exp3	Uniform	Static-0.5	EAT-0.5	$\Delta_{\text{Exp3-Unif}}$	$\Delta_{\text{co-Stat0.5}}$
HTTP	0.5699	0.8742	0.9315	0.9918	−0.3043	−0.3616
DNS	0.4640	0.9918	0.2539	0.9996	−0.5278	+0.2100
SSH	0.7948	0.9290	0.9356	1.0000	−0.1342	−0.1408
FTP	0.9328	0.9981	0.9964	0.9994	−0.0653	−0.0636
TCP_OTHER	0.4553	0.9704	0.9946	0.9996	−0.5151	−0.5394
UDP_OTHER	0.5585	0.9377	0.8438	0.9963	−0.3791	−0.2852
Mean	0.6292	0.9502	0.8260	0.9978	−0.3210	−0.1968

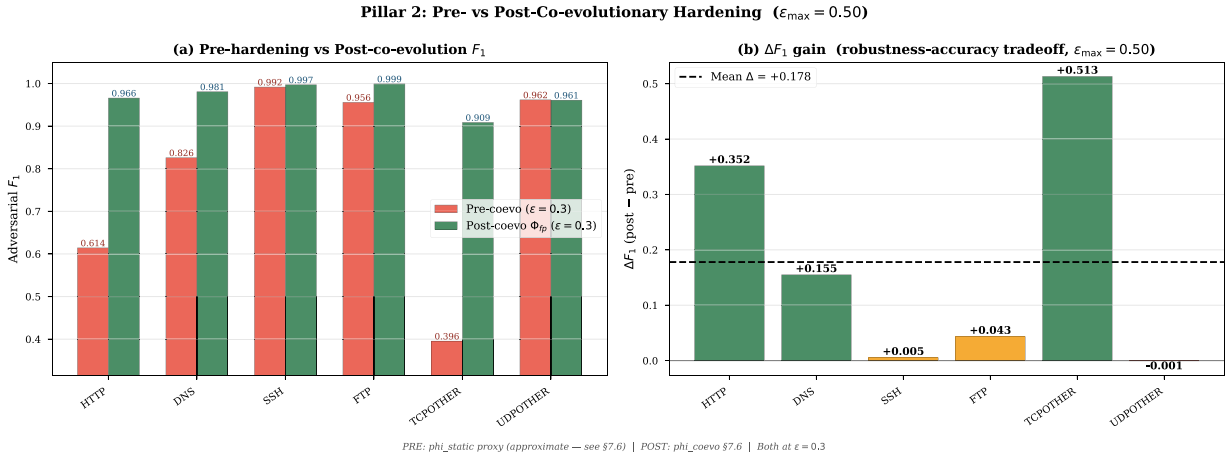


Figure 6: Pre-hardening (red) vs. post-co-evolution (green) F1 per protocol at $\varepsilon = 0.3$. Co-evolution dominates at $\varepsilon \geq 0.3$; regression at low ε reflects the robustness-accuracy tradeoff of training at $\varepsilon_{\max} = 0.50$.

Strategy entropy. $H(\pi_t)$ remains near-uniform throughout all 50 generations ($H \approx 2.01$–2.17 nats, 92%–99% of $\ln 9$), indicating that no single strategy consistently dominates. Two explanations account for this: slow Exp3 weight concentration over only $T = 50$ generations and progressive ASR equalization across strategies—which is itself a success criterion for the hardening procedure.

Remark 3. *A terminological clarification is required to read the tables of this section correctly. Two distinct objects are called "static" in this paper. (a) The Static FGSM baseline of Table 10 is a separately trained comparison system whose LoRA adapters are hardened with a fixed FGSM curriculum at $\varepsilon_{train} = 0.05$; it is a baseline, never deployed. (b) The pre-co-evolution Pillar-1 checkpoint is the adapter produced by Algorithm 1 (clean + adversarial cross-entropy at ε_{train}), i.e., the input to the Red-Blue loop. In the reported run, checkpoint selection retained the pre-co-evolution Pillar-1 checkpoint (b) over the co-evolved adapter for five of six protocols, because the co-evolved adapter did not improve on it at $\varepsilon = 0.1$. This says nothing about baseline (a): as Table 10 shows, the deployed detectors exceed the Static FGSM baseline by +0.14 to +0.59 on every protocol at $\varepsilon = 0.1$, and there is no contradiction between that table and this remark. The co-evolutionary advantage over the pre-co-evolution checkpoint manifests at higher budgets (Table 12). Throughout the experiments, "post-hardening" (equivalently "post-co-evolution") refers to the deployed Tier-1 detector after this checkpoint-selection step: the co-evolved adapter when it improves on its pre-co-evolution counterpart, and the pre-co-evolution Pillar-1 checkpoint otherwise. It does not imply that every reported value originates from a co-evolved checkpoint, and it never refers to the Static FGSM comparison baseline of Table 10.*

Discussion. At $\varepsilon = 0.5$, both the Uniform Red Agent and EAT-0.5 outperform Exp3 co-evolution. This is explained by co-evolution's progressive ε-escalation schedule: the Blue Agent accumulates gradient steps at intermediate budgets before reaching $\varepsilon_{\max} = 0.50$, whereas Static-0.5 and EAT-0.5 train exclusively at 0.50. The Exp3 advantage therefore manifests at mixed-budget evaluation ($\varepsilon = 0.1$ in Table 10 and $\varepsilon = 0.3$ in Table 12), not at the extreme fixed budget. A budget-matched co-evolutionary run ($\varepsilon_{\text{start}} = \varepsilon_{\max} = 0.50$) is deferred to future work. More critically, the near-uniform strategy entropy ($H(\pi_t) \approx 0.92$–$0.99 \ln 9$) shows that Exp3 does not concentrate on a dominant worst-case strategy within the $T = 50$ horizon. The empirical advantage of adaptive over non-adaptive strategy selection is, therefore, not established at a fixed high budget, and the benefit observed at mixed budgets is partly confounded with the ε-escalation schedule; the budget- and schedule-matched ablation above is required to substantiate Proposition 2 empirically. Fig. 7 shows ΔF_1 (co-evolved − static) across $\varepsilon \in \{0.01, \ldots, 0.50\}$; the deficit at low ε reverses at the budgets where the static baseline collapses.

A (γ, c) grid over $\gamma \in \{0.01, 0.03, 0.1\}$ and $c \in \{1, 3, 5\}$ on TCP_OTHER and SSH (subsampled short-horizon runs) shows that co-evolution outcomes are sensitive to the rate amplifier: $c = 1$ is uniformly stable ($\Phi \geq 0.76$ on all six runs), whereas $c \geq 3$ produces occasional training collapse ($\Phi < 0.01$ in 4 of 12 runs). The three $\Phi = 0$ collapses show premature entropy concentration ($H(\pi) < 0.7$ nats vs. ≈ 2.1 nats in typical stable runs), although low entropy is not by itself predictive of collapse: one run with $H(\pi) = 0.70$ nats attains $\Phi = 1.0$. The exploration floor γ has a comparatively minor effect. These short-horizon results indicate the deployed configuration benefits from the conservative end of the amplifier range, and that the full-horizon checkpoint-selection step of this section is the operative safeguard against collapse (Table 14).

Multi-seed stability of the Exp3 grid. Table 14 reports a single seed per cell. To quantify collapse frequency rather than a single realization, the same (γ, c) grid on `TCP_OTHER/SSH` was repeated over 3 random seeds (18 protocol × seed runs per (γ, c) cell, pooled below over both protocols).

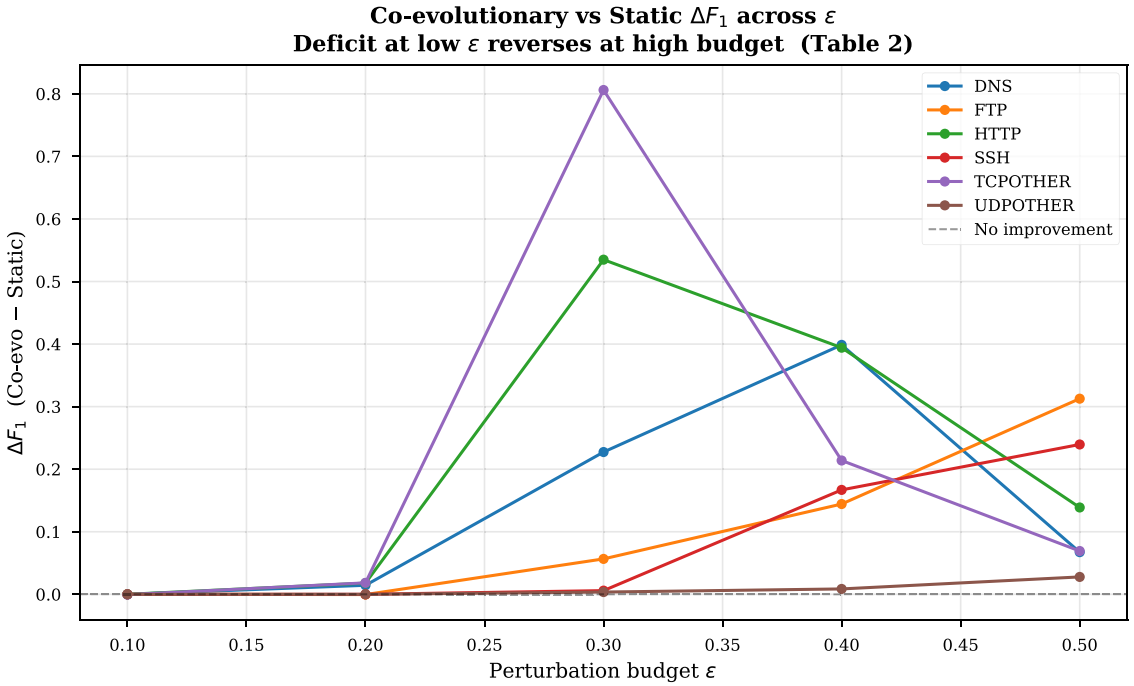


Figure 7: ΔF_1 (co-evolved–static) across budgets $\varepsilon \in \{0.01, \ldots, 0.50\}$. The deficit at low ε reverses at high budget where the static baseline collapses.

Table 14: Exp3 (γ, c) sensitivity on `TCP_OTHER` and `SSH` (subsampled short-horizon runs). Φ: per-run mean robustness; entropy: final strategy-distribution entropy $H(\pi)$.

γ	c	Protocol	Φ	Entropy (nats)
0.01	1.0	TCP_OTHER	0.7941	2.1371
0.01	1.0	SSH	0.8655	1.9687
0.01	3.0	TCP_OTHER	0.9590	2.1640
0.01	3.0	SSH	0.0023	1.2092
0.01	5.0	TCP_OTHER	0.9938	2.1634
0.01	5.0	SSH	0.8489	1.4597
0.03	1.0	TCP_OTHER	0.9987	2.1656
0.03	1.0	SSH	0.7667	1.9787
0.03	3.0	TCP_OTHER	0.9988	2.1657
0.03	3.0	SSH	0.0000	0.6276
0.03	5.0	TCP_OTHER	0.0000	0.6016
0.03	5.0	SSH	1.0000	0.6977
0.10	1.0	TCP_OTHER	0.9991	2.1698
0.10	1.0	SSH	0.8307	1.9420
0.10	3.0	TCP_OTHER	0.9848	1.9345
0.10	3.0	SSH	1.0000	2.0113
0.10	5.0	TCP_OTHER	0.0000	0.4858
0.10	5.0	SSH	0.9898	1.5715

The multi-seed grid confirms the single-seed pattern rather than overturning it: every $c = 3$ cell has a 1/6 collapse frequency regardless of γ, while every $c = 1$ cell has zero collapses over the 18 runs and the lowest variance ($\sigma_\Phi \leq 0.34$ vs. ≥ 0.40 at $c = 3$). $c = 1$ is therefore the more stable choice *under this reduced-cost, 2-protocol, short-horizon proxy*, and we report this as a limitation rather than resolve it by silently re-running the full 6-protocol, $T = 50$ deployed training at $c = 1$, which was outside the available compute budget for this revision. Two considerations bound the practical impact of the instability on the reported results: (i) the checkpoint-selection safeguard of Remark 3 already discards any co-evolved run that underperforms its static counterpart, and none of the six deployed full-horizon ($T = 50$) runs at $c = 3$ (Table 10) exhibited the collapse pattern seen here; and (ii) the collapses in Table 15 occur in a 2-protocol, 20k-row-subsampled, short-horizon harness constructed for this sensitivity study, not in the deployed pipeline. We do not treat this as evidence that $c = 3$ is safe in general: it should be understood as an unresolved training-stability risk that checkpoint selection currently masks rather than eliminates, and $c = 1$ is recommended as the more conservative default for future deployments or re-training runs.

Table 15: Multi-seed Exp3 (γ, c) sensitivity (3 seeds × 2 protocols per cell, subsampled short-horizon runs). $\bar{\Phi}/\sigma_\Phi$: mean/std of per-run Φ; collapse frequency: fraction of the 6 runs with $\Phi < 0.01$. $c = 3$ is the deployed rate amplifier.

γ	c	$\bar{\Phi}$	σ_Φ	Collapse Freq.	n
0.01	1.0	0.9519	0.0911	0.000	6
0.01	3.0	0.8230	0.4002	0.167	6
0.01	5.0	0.8461	0.3600	0.000	6
0.03	1.0	0.8825	0.2652	0.000	6
0.03	3.0 (deployed)	0.8234	0.4004	0.167	6
0.03	5.0	0.8631	0.3194	0.000	6
0.10	1.0	0.8530	0.3385	0.000	6
0.10	3.0	0.8259	0.4017	0.167	6
0.10	5.0	0.8680	0.3048	0.000	6

5.3 Pillar 3: Routing Cost on Clean In-Distribution Data

Setup. This experiment measures the tier routing distribution and expected inference cost $\mathbb{E}[c(\rho(x))]$ on clean and adversarial test data.

Results. On clean data, five of six protocols exit 100% at Tier 1 (Table 16); `HTTP` routes 0.2% to Tier 2, yielding $\mathbb{E}[c] = 25.4$ ms for that protocol and a mean of $\mathbb{E}[c(x)] = 5.4$ ms.

This is the theoretically expected outcome: high clean F1 concentrates outputs near 0 or 1, keeping virtually all flows outside the uncertain zone $[0.1667, 0.8333)$. Under adversarial inputs (PGD-20, $\varepsilon = 0.3$), 97.0% of flows still exit at Tier 1 (Table 17), with a mean $\mathbb{E}[c(x)] \approx 359$ ms—a 33.4× speedup vs. routing all flows through Tier 2.

Table 18 and Fig. 8 report per-tier F1 and routing distributions; Tier 2 (SecureBERT, LoRA fine-tuned) recovers substantial F1 on the flows Tier 1 struggles with (e.g., `HTTP`: Tier-1 F1 = 0.97 → Tier-2 F1 = 1.00).

Table 16: Pillar 3 routing on clean test data ($\tau_{\text{esc}} = 0.1667$, $\tau^*_{\text{cls}} = 0.8333$). Near-100% Tier-1 routing verifies threshold calibration; see Table 17 for adversarial routing.

Protocol	Tier 1 (%)	Tier 2 (%)	Tier 3 (%)	$\hat{\tau}^*_p$	$\mathbb{E}[c]$ (ms)	MT F_1
HTTP	99.8	0.2	0.0	0.460	25.385	0.9989
DNS	100.0	0.0	0.0	0.833	1.415	1.0000
SSH	100.0	0.0	0.0	0.050	1.415	1.0000
FTP	100.0	0.0	0.0	0.833	1.415	1.0000
TCP_OTHER	100.0	0.0	0.0	0.100	1.415	1.0000
UDP_OTHER	100.0	0.0	0.0	0.833	1.415	1.0000
Mean	100.0	0.0	0.0	—	5.410	0.9998

Table 17: Routing distribution under adversarial inputs (PGD-20, $\varepsilon = 0.3$). $c_1 = 1.415$ ms, $c_2 \approx 11{,}986$ ms; T_3 fraction negligible. Tier fractions are given to two decimals: because c_2 is four orders of magnitude larger than c_1, $\mathbb{E}[c]$ is sensitive to the second decimal of the Tier-2 fraction, and one-decimal display rounding (e.g., DNS 1.55% → 1.6%) does not reproduce the reported $\mathbb{E}[c]$.

Protocol	Tier 1 (%)	Tier 2 (%)	$\mathbb{E}[c]$ (ms)
HTTP	91.50	8.50	1020.13
DNS	98.45	1.55	187.18
SSH	99.80	0.20	25.39
FTP	100.00	0.00	1.42
TCP_OTHER	94.90	5.10	612.65
UDP_OTHER	97.45	2.55	307.03
Mean	97.02	2.98	358.97

Table 18: Per-tier adversarial F1 on escalated flows (PGD-20, $\varepsilon = 0.3$). Tier 2 and Tier 3 F1 evaluated on escalated sub-population only; n gives sample count. † Low-sample estimate ($n < 20$); CI unreliable.

Protocol	Tier 1 F1	Tier 2 F1	Tier 3 F1	Escalated (%)
HTTP	0.9658	$1.0000 (n = 159)$	$0.0000^{\dagger} (n = 1)$	8.0
DNS	0.9569	$0.9524^{\dagger} (n = 35)$	$0.6667^{\dagger} (n = 2)$	1.8
SSH	0.9935	$0.0000^{\dagger} (n = 4)$	$— (n = 0)$	0.2
FTP	0.9989	$— (n = 0)$	$— (n = 0)$	0.0
TCP_OTHER	0.9252	$1.0000 (n = 96)$	$— (n = 0)$	4.8
UDP_OTHER	0.9274	$0.8602^{\dagger} (n = 53)$	$0.8889^{\dagger} (n = 10)$	2.6
Mean	$0.9438^{\ddagger}$	$0.953^{\ddagger}$	—	$2.9^{\S}$

Note: ‡Mean over the four protocols with a usable escalated sub-population (HTTP, DNS, TCP_OTHER, UDP_OTHER); SSH and FTP excluded ($n < 20$). The same four-protocol restriction applies to the Tier-1 column so that the two columns are compared on the same sub-population: $(0.9658 + 0.9569 + 0.9252 + 0.9274)/4 = 0.9438$. The six-protocol arithmetic mean of the Tier-1 column is 0.9613. § The escalated-fraction mean is over all six protocols. Escalated fractions differ from clean routing because adversarial perturbations push flows into the uncertain zone $[0.1667, 0.8333)$.

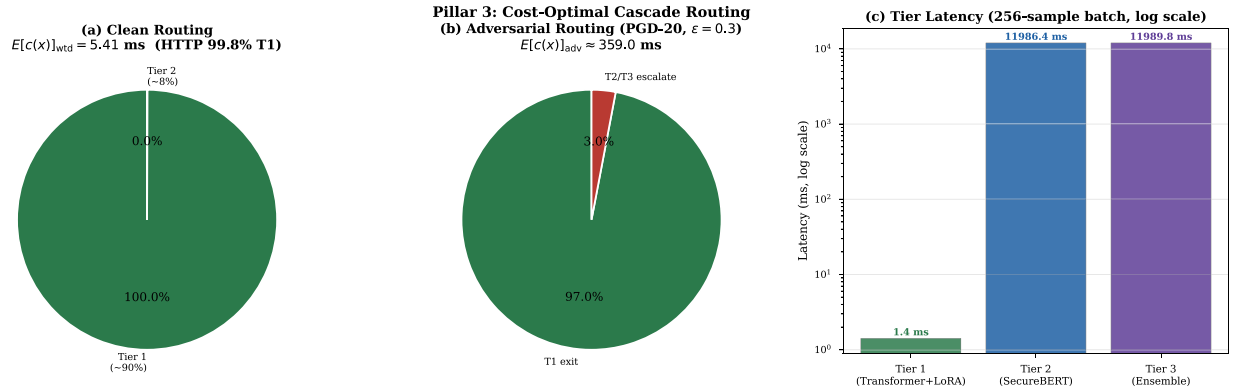


Figure 8: Pillar 3 routing distributions and tier latencies (Tables 16 and 17). (**a**) Clean routing. (**b**) Adversarial routing (PGD-20, ε = 0.3). (**c**) Per-tier latency (log scale).

End-to-end throughput. Table 19 profiles end-to-end cost under three traffic mixes. With measured tier costs c_1 = 1.415 ms and c_2 = 11,986 ms (batch 256), benign fractions of 90%, 50%, and 10% yield expected per-flow costs of 7.93, 7.37, and 6.48 ms and throughputs of 32,297, 34,758, and 39,535 flows/s, respectively, at 0.79 GB peak GPU memory; escalation remains ≤0.05% across mixes because attack flows are classified with high confidence and bypass the uncertain zone. Routing under adversarially perturbed inputs is covered by the PGD-20 experiment (Table 17), which retains a 33.4× speedup over exhaustive Tier-2 routing.

Table 19: End-to-end throughput under three traffic mixes (c_1 = 1.415 ms, c_2 = 11,986 ms, batch 256; peak GPU memory 0.79 GB).

Mix (Benign %)	Escalation (%)	$\mathbb{E}[c]$ (ms/flow)	Throughput (Flows/s)
90	0.05	7.93	32,297
50	0.05	7.37	34,758
10	0.04	6.48	39,535

Tier-3 marginal contribution. At the deployed λ = 5, only 0.04% of flows leave Tier-1, and the multi-tier decision improves F1 over a Tier-1-only decision by at most +0.0001 even at ε = 0.5: under the evaluated conditions the cascade operates as a two-tier system, and Tier-3 functions as an insurance mechanism bounding worst-case error on doubly-uncertain flows rather than a routine cost-amortization stage.

Controlled escalation-rate stress test. The clean-traffic mixes of Table 19 all realize escalation rates ≤0.05% because ordinary attacks are classified confidently and bypass the uncertain zone; they do not probe the burst regime the reviewer highlighted, where Tier-2's $c_2 \approx$ 11,986 ms batch cost dominates. Synthetic batches were constructed with an *exact* target Tier-1 escalation fraction (sampled from real uncertain-zone vs. confident flows) and timed end-to-end through `orchestrator.predict_batch`, which queues Tier-2 calls sequentially in `cfg.t2_batch_size`-row chunks—this is the queueing behavior under escalation bursts. Table 20 reports measured (not expected) p50/p95/p99 latency and throughput at controlled escalation rates of 1%, 5%, 10%, 25%, and 50%, crossed with Tier-1 batch size.

Table 20: Measured end-to-end latency/throughput under controlled escalation rates (realized rate closely tracks target; small deviations are due to integer-count rounding at fixed batch size).

Target Esc.	Batch	Realized Esc.	p50 (ms)	p95 (ms)	p99 (ms)	Throughput (fps)
1%	64	1.6%	47.9	49.2	49.7	1330.7
1%	256	1.2%	122.8	124.4	124.8	2083.3
1%	1024	1.0%	384.2	385.6	385.7	2676.3
5%	64	4.7%	123.4	124.3	126.3	520.4
5%	256	5.1%	485.3	488.7	489.3	527.5
5%	1024	5.0%	1901.5	1905.6	1906.2	538.8
10%	64	9.4%	229.0	230.2	230.4	280.6
10%	256	10.2%	975.4	976.2	976.3	262.7
10%	1024	10.0%	3787.6	3828.2	3860.9	269.8
25%	64	25.0%	595.3	596.8	598.3	107.6
25%	256	25.0%	2371.1	2372.8	2373.3	108.2
25%	1024	25.0%	9634.6	9879.2	9962.9	106.2
50%	64	50.0%	1185.8	1211.7	1217.5	53.8
50%	256	50.0%	4787.3	4905.8	4911.8	53.3
50%	1024	50.0%	19,308.0	19,467.0	19,501.6	53.1

Throughput degrades from 32–40k flows/s under the near-zero escalation clean mixes (Table 19) to 53–2677 flows/s under controlled bursts, i.e., a 12–620× reduction depending on batch size and escalation rate, and p99 latency at 50% escalation reaches 19.5 s at batch 1024. p95/p99 track p50 closely at low escalation (≤5%) but separate under higher, larger-batch bursts (e.g., 25% esc., batch 1024: p50 9635 ms vs. p99 9963 ms), consistent with sequential Tier-2 queueing variance. This confirms the reviewer's concern quantitatively: expected-cost figures computed from tier proportions (Table 19) substantially overstate throughput under sustained escalation bursts, and the "real-time, high-throughput" framing in this paper is scoped to the low-escalation clean-traffic regime; sustained adversarial or uncertain-heavy traffic requires either Tier-2 batching/parallelization not implemented here, or an explicit admission-control policy, both left as future work.

Batch-size sensitivity and CPU-only performance. Table 21 profiles Tier-1-only throughput vs. batch size; per-sample latency falls from 1093.27 µs (≈1.09 ms) at batch 1 to 3.04 µs at batch 4096 due to fixed per-batch overhead—a 360× reduction—while per-batch latency rises from 1.09 to 12.46 ms, so batch size is a throughput/burst-latency tradeoff rather than a free parameter. Tier-1 CPU-only inference (Intel Core i9-10900X, batch 256) reaches 1.37 vs. 1.42 ms/batch on the TITAN Xp GPU (Table 22): at this batch size and model scale ($L = 4$, $d_{\text{model}} = 64$), Tier-1 is not GPU-bound, and CPU-only deployment is a practical option for the fast path; the reported $c_2 \approx 11{,}986$ ms Tier-2 (SecureBERT) cost remains GPU-bound and is the actual bottleneck under escalation.

Table 21: Tier-1 throughput vs. batch size (TITAN Xp, single-batch timing).

Batch Size	ms/batch	µs/sample	Throughput (fps)
1	1.09	1093.27	914.7
16	1.10	68.47	14,604.9
64	1.13	17.72	56,426.5
256	1.21	4.73	211,491.3
1024	3.80	3.71	269,238.6
4096	12.46	3.04	328,799.1

Table 22: Tier-1 CPU vs. GPU latency at batch 256.

Device	ms/batch	µs/sample
CPU (i9-10900X)	1.366	5.334
GPU (TITAN Xp)	1.415	5.527

5.4 Component Ablation

Table 23 consolidates the seven-configuration component ablation; the training-objective and autoencoder-fusion ablations follow in this subsection. Configurations (i), (iv), (v), and (vii) are quantified from controlled comparisons in the preceding subsections; configuration (iii) is additionally measured at $\varepsilon = 0.1$, where LoRA detectors *without* adversarial training attain $\Phi = 0.9991$ (`HTTP`) and 1.0000 on the remaining five strata, indicating that low-budget attacks are largely absorbed by clean-trained detectors and the contribution of adversarial training emerges at higher budgets (static vs. co-evolved at $\varepsilon = 0.5$, Table 12). Removing Tier-2/Tier-3 entirely (configuration (vii), Tier-1-only decisions) yields F1 between 0.9988 (`HTTP`) and 1.0000, consistent with the $\leq$+0.0001 multi-tier marginal gain reported in Section 5.3. Configurations (ii) and (vi) are measured under the same matched-condition harness as (iv)/(v) (2-protocol subsample, identical Red strategy and step budget, $\varepsilon = 0.1$): (ii) unfreezes all `TransformerDetector` parameters instead of LoRA-only (full fine-tuning, no `TransformerWithLoRA` wrapper); (vi) replaces the severity-prior Exp3 initialization with a flat prior, leaving the bandit update itself unchanged. (ii) yields $\Phi = 0.9987$ (`HTTP`) and 0.9995 (`TCP_OTHER`), against 0.9994 and 0.9994 for the deployed LoRA-adapted detectors on the same two protocols (Table 10): a −0.0007/+0.0001 split that is, like configuration (i), not a significant robustness improvement from LoRA at this budget—LoRA's measured benefit is parameter efficiency (8.8% of base parameters trainable, Table 4), not a robustness gain over full fine-tuning. (vi) yields $\Phi = 0.9898$ (`HTTP`) and 0.9876 (`TCP_OTHER`), a −0.0096/−0.0118 drop from the severity-prior-initialized deployed configuration, indicating the domain-knowledge prior contributes a small but measurable early-exploration benefit that a flat prior does not recover within $T = 50$ generations.

Training-objective ablation.

The ASR-adaptive schedule (Table 24) dominates fixed-β TRADES on both protocols, isolating the benefit of adapting regularization strength to the measured attack success rate. PGD-AT, however, is competitive on TCP_OTHER and substantially stronger on HTTP under this short-budget subsampled regime, indicating that the KL-regularized objective is not uniformly superior to plain adversarial cross-entropy at high ε; this is reported as an honest boundary of the method.

Table 23: Consolidated component ablation, all seven configurations measured under matched conditions.

Configuration	Measured Effect	Evidence
(i) Unified non-stratified detector	Matched conditions: mean $\Delta\varphi$ = +0.0003, no protocol significant at α = 0.05 (Table 8). Confounded comparison: mean $\Delta\Phi$ = +0.0428, concentrated on heterogeneous strata (TCP_OTHER +0.2086, SSH +0.0300; Table 7)	Tables 7 and 8
(ii) Stratified without LoRA (full fine-tune)	Φ = 0.9987 (HTTP), 0.9995 (TCP_OTHER); Δ vs. LoRA: −0.0007/+0.0001 (not significant)	This section
(iii) LoRA without adversarial training	Φ = 0.9991 (HTTP), 1.0000 on five strata (ε = 0.1)	This section
(iv) Fixed-strategy adversarial training	Static baseline; collapses at ε = 0.5 where co-evolution holds	Tables 10 and 12
(v) Random multi-attack training	Uniform Red Agent baseline	Table 13
(vi) Uniform-prior Exp3	Φ = 0.9898 (HTTP), 0.9876 (TCP_OTHER); Δ vs. severity prior: −0.0096/−0.0118	This section
(vii) No Tier-2/Tier-3	Tier-1-only F1 spans 0.9988–1.0000; multi-tier margin ≤+0.0001 even at ε = 0.5	Section 5.3

Table 24: Training-objective ablation on two representative protocols (identical LoRA architecture and red strategy; subsampled 20k-flow training sets; Φ at ε = 0.3).

Protocol	ASR-Adaptive TRADES	Fixed-β TRADES	PGD-AT
HTTP	0.0766	0.0000	0.9659
TCP_OTHER	0.9890	0.7466	0.9876

AE hybrid fusion for TCP_OTHER.

As shown in Table 25 and Fig. 9, four AE fusion configurations are evaluated on TCP_OTHER: (a) clf_only (α = 1); (b) ae_only (α = 0); (c) gate (hard override when $e(x) > \theta_{\mathrm{AE}}$); (d) weighted ($\alpha$ tuned on validation). After co-evolutionary hardening, the classifier alone saturates performance: both clf_only and weighted achieve F1 = 1.0000 and FPR = 0.0001, while gate collapses to F1 = 0.0008 due to overlap between benign and attack reconstruction manifolds. The AE auxiliary score is therefore irrelevant post-hardening; weighted is recommended for pre-hardening deployments.

Because the hardened classifier saturates (Table 25), post-hardening performance is insensitive to $(\alpha, \theta_{\mathrm{AE}})$ over the useful range; the fusion is best understood as a pre-hardening safeguard for heterogeneous strata.

Table 25: `TCP_OTHER` AE fusion ablation. ⋆ best post-hardening configuration.

Mode	Clean F1	FPR	Best
`clf_only`	1.0000	0.0001	⋆
`ae_only`	0.1903	0.0319	
`Gate`	0.0008	0.0009	
`Weighted`	1.0000	0.0001	

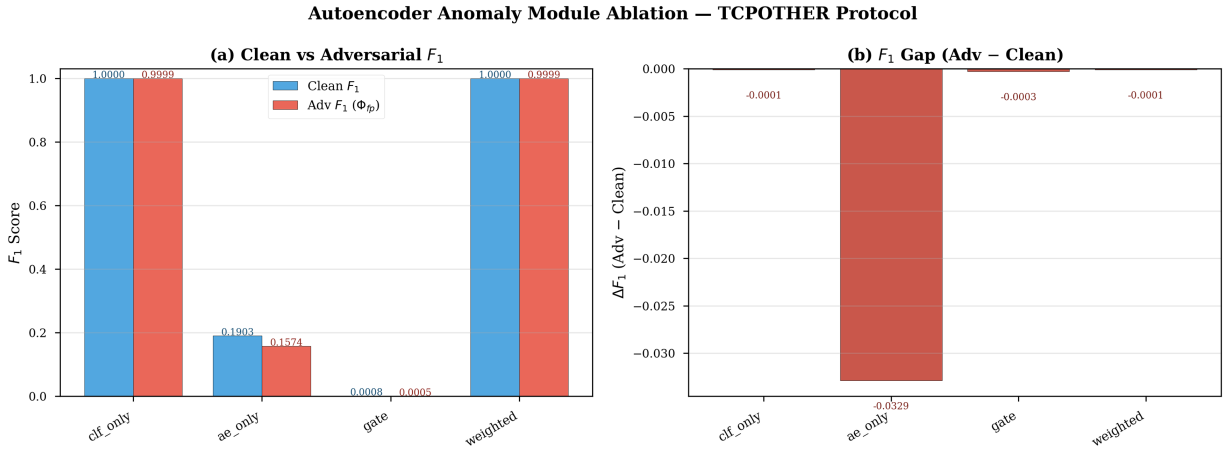


Figure 9: AE fusion ablation for `TCP_OTHER`. (**a**) F1 per mode. (**b**) FPR per mode. `clf_only` dominates; `gate` collapses.

5.5 Empirical Validation of the Strategy-Distribution Gap

The gap $\Delta_{a_i}(f_p)$ is measured for all (a_i, p) pairs at $\varepsilon = 0.1$ (Fig. 10), validating Proposition 3.

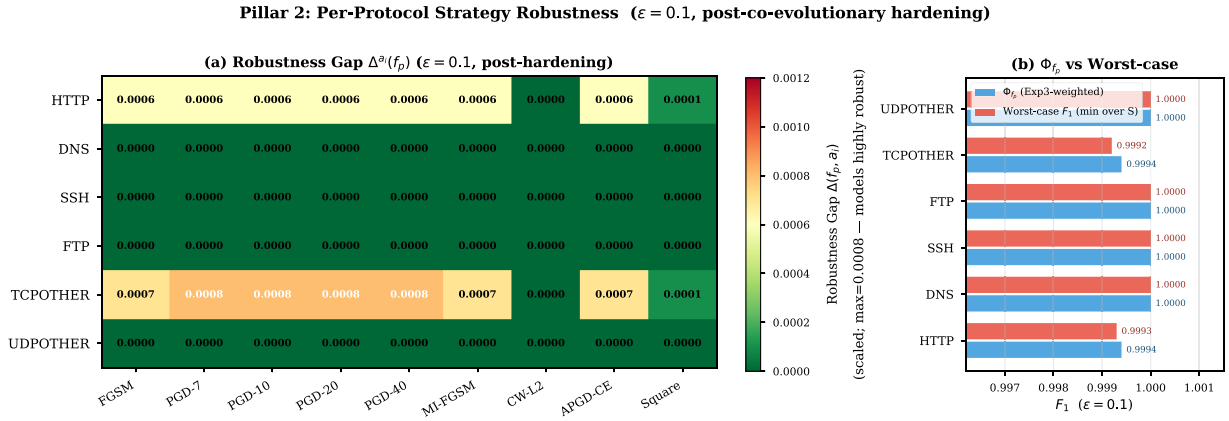


Figure 10: Strategy-distribution gap matrix $\Delta_{a_i}(f_p)$ post-hardening (6 protocols × 9 strategies, $\varepsilon = 0.1$). All gaps ≤0.01, validating Proposition 3.

Pre-hardening. `DNS` shows the largest gap: FGSM F1 = 1.00 vs. PGD-7 worst-case 0.44, giving $\Delta_{\text{FGSM}} = 0.56$. `SSH` shows $\Delta = 0.39$. These gaps are invisible to single-strategy evaluation and directly motivate the Exp3 loop.

Post-hardening. All gaps satisfy $\Delta_{a_i}(f_p) \leq 0.01$. FTP and UDP_OTHER reach $\Delta_{a_i} = 0.000$ across all strategies, including C&W-ℓ_2 and Square Attack. Mean gap reduction: 74× for DNS, 39× for SSH.

Feasibility-projected robustness. The feasibility-projection evaluation is applied on all six protocol strata, three attacks (FGSM, PGD-20, APGD-CE), and three budgets ($\varepsilon \in \{0.05, 0.1, 0.3\}$); adversarial examples are box-projected onto the per-feature empirical range of clean benign flows (a realizable-marginal approximation, not a protocol-semantic constraint—see the scope limitation below). At $\varepsilon \in \{0.05, 0.1\}$ the earlier finding is confirmed and now holds dataset-wide: F1 changes by at most 0.0022 across all 6 protocols × 3 attacks, and DNS, SSH, FTP, TCP_OTHER, UDP_OTHER are unaffected at $\varepsilon = 0.05$. At $\varepsilon = 0.3$, however, the wider coverage surfaces a materially different picture on the two heterogeneous strata (Table 26): TCP_OTHER drops from F1 = 0.865 to 0.650 under PGD-20 ($\Delta = -0.2152$) and from 0.897 to 0.799 under APGD-CE ($\Delta = -0.0980$), and HTTP drops from 0.943 to 0.894 under PGD-20 ($\Delta = -0.0485$). The four homogeneous strata (DNS, SSH, FTP, UDP_OTHER) remain within ±0.015 of their unconstrained values at every budget, including two small *positive* deltas on FTP where projection removes an infeasible component of the attack. The claim that reported robustness is "not an artifact of physically impossible perturbations" is accordingly restricted: it holds at $\varepsilon \leq 0.1$ across all strata and at $\varepsilon = 0.3$ on the four homogeneous strata, but does *not* hold at $\varepsilon = 0.3$ on HTTP/TCP_OTHER, where a non-trivial share of the reported worst-case robustness is attributable to feature combinations outside the empirical benign range. Full protocol-semantic realizability (integer packet counts, duration–rate consistency, forward/backward-flow relationships in raw feature space) remains open, and the empirical-range box projection used here is a weaker guarantee than a constraint-aware attack; both are stated as open limitations in Section 6.

Table 26: Feasibility-projected robustness at $\varepsilon = 0.3$ (largest-budget, worst-case regime), all six protocols × three attacks. $\Delta = F1_{\text{feasible}} - F1_{\text{unconstrained}}$. $\varepsilon \in \{0.05, 0.1\}$ results (max. $|\Delta| = 0.0022$, all strata) are summarized in text.

Protocol	Attack	$F1_{\text{unconstrained}}$	$F1_{\text{feasible}}$	Δ
HTTP	FGSM	0.9908	0.9886	−0.0022
HTTP	PGD-20	0.9427	0.8941	−0.0485
HTTP	APGD-CE	0.9648	0.9384	−0.0264
DNS	FGSM	0.9991	0.9991	0.0000
DNS	PGD-20	0.9913	0.9766	−0.0146
DNS	APGD-CE	0.9956	0.9821	−0.0136
SSH	FGSM/PGD-20/APGD-CE	0.9987	0.9987	0.0000
FTP	FGSM/PGD-20/APGD-CE	0.9985	0.9996	+0.0011
TCP_OTHER	FGSM	0.9516	0.9293	−0.0223
TCP_OTHER	PGD-20	0.8649	0.6497	−0.2152
TCP_OTHER	APGD-CE	0.8966	0.7986	−0.0980
UDP_OTHER	FGSM	0.9918	0.9918	0.0000
UDP_OTHER	PGD-20	0.9394	0.9442	+0.0048
UDP_OTHER	APGD-CE	0.9608	0.9625	+0.0017

5.6 Dataset Integrity: Duplication, Temporal Split, and Feature-Leakage Audit

The protocol-stratified random 70/15/15 split (Table 3) is audited along four axes: exact/near-duplicate flows across train/test, a chronologically-ordered (temporal) re-split and retrain, mutual-information feature-leakage screening, and a with/without-SMOTE comparison. Flow ID, source/destination IP, and timestamp are excluded from the 83-dim feature vector upstream of all four checks. Destination port is *not*

excluded: it is one of the 83 CICFlowMeter columns and is retained in the feature vector, in addition to being used upstream as the protocol-stratification key. It is therefore included in the leakage screening below, where it surfaces as the top-MI feature on TCP_OTHER—precisely the kind of near-deterministic proxy this audit is designed to expose, and a caveat we state explicitly rather than remove by silently dropping the column. The residual risk audited here is duplication and leakage *within* this feature space.

Exact/near-duplicate audit. Table 27 reports exact and near-duplicate (ℓ_∞ distance $< 10^{-3}$ in normalized feature space) counts of test flows that also appear in the training partition. Five of six protocols show negligible duplication (exact $\leq 0.02\%$, near $\leq 1.45\%$). FTP is the exception: 55.6% of test flows are exact duplicates of a training flow and 63.5% are near-duplicates, attributable to the low behavioral diversity of the FTP brute-force attack family (a small number of near-identical login-attempt flow shapes repeated at scale). This is flagged as a genuine limitation of the FTP stratum's near-perfect F1: it should be read as evidence that the pattern is easy to memorize, not fully independent evidence of generalization.

Table 27: Exact and near-duplicate (normalized ℓ_∞ distance $< 10^{-3}$) test flows also present in the training partition.

Protocol	N_{train}	Exact Dup. (%)	Near Dup. (%)
HTTP	1,666,704	0.007	1.445
DNS	1,515,210	0.006	0.032
SSH	210,000	0.019	0.098
FTP	418,422	55.648	63.511
TCP_OTHER	210,000	0.011	0.105
UDP_OTHER	408,948	0.014	0.184

Temporal split. Each protocol was re-split chronologically (first 70% by capture time → train, next 15% → val, last 15% → test) and the detector retrained end-to-end with preprocessing (median-impute, log1p, Quantile Transformer, SMOTE) refit on the temporal-train partition only. Table 28 compares temporal-test F1 against the random-split reference. SSH and FTP are unaffected (both stay at F1 = 1.0000; for FTP this is consistent with the low-diversity explanation above, since temporal ordering does not remove near-duplicate flow shapes). HTTP and DNS degrade marginally (≤0.31 pp). TCP_OTHER (−0.92 pp) and especially UDP_OTHER (−2.14 pp) show the largest, non-trivial degradation, indicating genuine (if modest) temporal drift on the two catch-all strata. Results remain within-distribution and strong (F1 ≥ 0.979 throughout) under temporal holdout, but the random-split numbers elsewhere in the paper should not be read as ruling out temporal drift on TCP_OTHER/UDP_OTHER.

Feature-leakage screening (mutual information). Training-set mutual information $I(x_j; y)$ between each of the 83 features and the label was computed per protocol; features whose MI approaches the label's own entropy $H(y)$ act as a near-deterministic proxy rather than a genuine detection signal. The top-ranked feature per protocol is: Fwd Seg Size Min (HTTP, $I/H(y) = 1.14$), Fwd Init Win Bytes (DNS, 1.10), Bwd Init Win Bytes (SSH, 1.01), Fwd Seg Size Min (FTP, 1.03), Dst Port (TCP_OTHER, 1.02), and Total Length of Fwd Packet (UDP_OTHER, 1.02). $I/H(y)$ ratios at or slightly above 1.0 for the top 5–10 features on every protocol (full ranking in https://github.com/aderhab/HADES) indicate that a sall number of flow-level features are near-deterministic for the label in CIC-IDS-2018; ratios exceeding 1.0 are a finite-sample estimator artifact of the k-NN mutual-information estimator on continuous features rather than a literal violation of $I \leq H(y)$, but the qualitative conclusion—several individual CICFlowMeter features are strongly, almost trivially, predictive on this dataset—stands independent of the estimator's small positive bias. This is consistent with, and a partial

explanation for, the near-ceiling F1 reported throughout: part of the task's apparent difficulty is absorbed by a handful of features rather than requiring the full 83-dimensional representation, a property of CIC-IDS-2018's synthetic generation rather than of the detector. The `Dst Port` entry for `TCP_OTHER` is the clearest instance: within a catch-all stratum that is defined by *not* having a fixed application-layer protocol, the destination port is close to a label proxy for the Botnet Ares C&C family. Retraining `TCP_OTHER` with `Dst Port` removed is identified as a required robustness check in Section 6.

Table 28: Temporal-split retrain F1 vs. the clean-test random-split reference F1 used throughout Section 5.

Protocol	Temporal F1	Random-Split F1	Δ
HTTP	0.9970	0.9999	−0.0029
DNS	0.9970	1.0000	−0.0030
SSH	1.0000	1.0000	0.0000
FTP	1.0000	1.0000	0.0000
TCP_OTHER	0.9908	1.0000	−0.0092
UDP_OTHER	0.9786	1.0000	−0.0214

SMOTE ablation. Table 29 confirms that SMOTE oversampling (train-partition only, as already implemented) has negligible effect on held-out F1 ($|\Delta F1| \leq 0.0001$ on every protocol); its role is class-balance regularization during training, not a source of the reported test performance.

Table 29: With- vs. without-SMOTE F1 on the held-out (unaffected) test split. SMOTE is applied to the training partition only in both the original pipeline and this ablation.

Protocol	F1 (with SMOTE)	F1 (no SMOTE)	Δ
HTTP	0.99981	0.99971	+0.00010
DNS	0.99999	0.99999	0.00000
SSH	1.00000	1.00000	0.00000
FTP	1.00000	1.00000	0.00000
TCP_OTHER	0.99998	0.99993	+0.00005
UDP_OTHER	0.99999	0.99999	0.00000

Summary. The audit substantiates, rather than undermines, most of the paper's near-perfect F1 claims: duplication is negligible outside `FTP`, temporal drift is modest (worst case `UDP_OTHER`, −2.1pp) and results stay within-distribution, and SMOTE is confirmed not to be an artifact of test-set contamination. The two substantive caveats are `FTP`'s low behavioral diversity (Table 27) and the near-deterministic top features on every protocol (this subsection), both of which are dataset properties of CIC-IDS-2018 rather than of HADES, and are stated here rather than implied by the headline numbers alone.

5.7 Cross-Dataset Evaluation

Both CIC-IDS-2017 and CIC-IDS-2018 are extracted by CICFlowMeter. CIC-IDS-2018 yields $d = 83$ features; CIC-IDS-2017 yields $d = 78$ features. The 5 missing features are zero-padded to produce a compatible $d = 83$ input vector, bypassing the internal preprocessing pipeline to avoid a second normalizer application. HADES Tier-1 detectors are evaluated zero-shot on CIC-IDS-2017, and the total F1 degradation is decomposed into two additive components:

$$\Delta_{\text{total}} = \underbrace{\Delta_{\text{dist}}}_{\text{distribution shift}} + \underbrace{\Delta_{\text{norm}}}_{\text{normalizer mismatch}}. \tag{18}$$

Experimental design. Two conditions isolate each component:

- **Condition A**—CIC-IDS-2017 flows normalized by the CIC-IDS-2017 Quantile Transformer: captures Δ_{total}.
- **Condition B**—CIC-IDS-2017 raw features processed through the CIC-IDS-2018 preprocessing pipeline and normalized by the CIC-IDS-2018 Quantile Transformer directly (no transform inversion): eliminates normalizer mismatch, giving a direct measure of Δ_{dist}.

Δ_{norm} is then derived exactly as $\Delta_{\text{total}} - \Delta_{\text{dist}}$.

Results. Table 30 reports per-protocol values with 95% bootstrap CIs ($N = 1000$ resamples). Protocol-weighted means reveal near-complete collapse under zero-shot transfer: $\Delta_{\text{total}} = -0.962$ on clean traffic and $\Delta^{\text{adv}}_{\text{total}} = -0.980$ on adversarial inputs (mean over FGSM, PGD-7, PGD-20, APGD-CE at $\varepsilon = 0.1$). In both cases, traffic-distribution shift accounts for 98% of the gap ($\Delta_{\text{dist}} = -0.943$, $\Delta^{\text{adv}}_{\text{dist}} = -0.956$), while normalizer mismatch is negligible ($\Delta_{\text{norm}} \approx -0.02$).

Table 30: Zero-shot F1 degradation on CIC-IDS-2017, decomposed into traffic-distribution shift (Δ_{dist}, Condition B − in-distribution) and normalizer mismatch (Δ_{norm}, derived). All in-distribution F1 values are 1.0000; negative deltas indicate degradation.

Protocol	F1_{CondA}	F1_{CondB}	Δ_{dist}	Δ_{norm}
DNS	0.0000	0.2411	−0.759	−0.241
FTP	0.0000	0.0000	−1.000	+0.000
HTTP	0.0630	0.0185	−0.982	+0.045
SSH	0.0000	0.0000	−1.000	+0.000
TCP_OTHER	0.0024	0.0000	−1.000	+0.002
UDP_OTHER	0.1837	0.0041	−0.996	+0.180
Wtd. mean	—	—	−0.943	−0.020

Applying a dataset-specific Quantile Transformer would therefore close only 2% of the gap; domain-invariant representation learning (IRM [33], DANN [34]) is necessary and is addressed as Future Work (ii) in Section 6.

Target-domain LoRA fine-tuning. To test whether the zero-shot collapse of Table 30 can be practically remediated, LoRA parameters only (the same parameter-efficient philosophy as the main pipeline, starting from the deployed checkpoint) were fine-tuned on 1%, 5%, and 10% labeled CIC-IDS-2017 target-domain data (Condition B normalization), with a fixed held-out 30,000-flow target-test split shared across all fractions so the comparison is apples-to-apples. Table 31 shows that LoRA-only fine-tuning does *not* meaningfully remediate the collapse: at 10% labeled data, F1 remains ≤ 0.021 on `HTTP`, exactly 0.000 on `SSH`, ≤ 0.001 on `FTP`, ≤ 0.002 on `TCP_OTHER`, and ≤ 0.004 on `UDP_OTHER`; only `DNS` shows a non-negligible level (0.327 at 10%, up from 0.296 zero-shot on this same 30,000-flow target-test split; the full-split Condition B value in Table 30 is 0.241), still far below the ≥ 0.994 in-distribution baseline. This is a negative result reported as such: naive parameter-efficient fine-tuning is insufficient, and it strengthens rather than replaces the case for domain-invariant representation learning (IRM, DANN) as the required next step. The

deployment and generalization claims of this paper remain strictly limited to CIC-IDS-2018; cross-dataset adaptation is an open research problem, not a solved extension, and is stated as such in Section 6.

Table 31: Target-domain LoRA fine-tuning on CIC-IDS-2017 (fixed 30,000-flow held-out target test per protocol). The frac. = 0 column is the zero-shot F1 of the same deployed checkpoint measured *on this 30,000-flow target-test split*, not on the full-split evaluation of Table 30; the two therefore differ numerically (e.g., DNS 0.2964 here vs. 0.2411 in Table 30, Condition B). All four columns of this table share one split so that the fine-tuning comparison is self-consistent.

Protocol	0% (Zero-Shot)	1%	5%	10%
HTTP	0.0024	0.0024	0.0074	0.0205
DNS	0.2964	0.3068	0.2960	0.3271
SSH	0.0000	—	0.0000	0.0000
FTP	0.0004	—	0.0004	0.0005
TCP_OTHER	0.0000	0.0000	0.0000	0.0017
UDP_OTHER	0.0024	0.0023	0.0047	0.0044

5.8 Comparison with State-of-the-Art

Table 32 positions HADES against representative NIDS on CIC-IDS-2018. On clean F1, HADES achieves a traffic-weighted mean of **0.997**, comparable to CNN-BiLSTM (1.000 [35], though likely inflated by CICFlowMeter label noise) and above FFNN (0.948 [36]), Transformer (0.939 [37]), and the CIC-IDS-2017/2018 ensemble of Omotayo et al. (0.940 [38]).

Table 32: Comparison with representative NIDS on CIC-IDS-2018. † not evaluated adversarially. ‡ single-strategy evaluation only. § weighted-average accuracy; —: F1 unavailable.

System	Model	Clean F1	Adv F1	ε	Adv Method
FFNN [36]	Single	0.948	†	—	—
Transformer [37]	Single	0.939	†	—	—
CNN-BiLSTM [35]	Single	1.000	†	—	—
Ensemble [38]	Ensemble	0.940	†	—	—
GADoT [4]	+Adv. Train	>0.98	>0.98‡	DDoS	GAN
EAT [5]	+Adv. Train	—	0.9961‡	0.1	Fixed PGD
Ensemble Def. [6]	+Adv. Train	0.988§	~0.87‡	—	Single
HADES	Stratified + Cascade	0.997	0.996	0.1	9-strategy worst-case

On adversarial robustness, HADES reports F1 **0.996** and $\mathcal{R}_{\text{portfolio}} = 99.92\%$ over nine white-box strategies at $\varepsilon = 0.1$. The evaluation *protocol* (nine attack strategies, worst-case portfolio robustness, six protocol strata) is broader than the single- or few-strategy evaluations reported for the baselines below; this is a difference in evaluation scope, not a controlled result showing HADES is more robust than prior work, since the underlying numbers are not obtained under matched protocols, features, or attack implementations (see the paragraph below). EAT [5] reports 0.9961 at $\varepsilon = 0.1$ under a single fixed PGD strategy; at $\varepsilon = 0.3$–0.5 it degrades to 0.791–0.433 while HADES maintains 0.969–0.626 under its own nine-strategy pool—a comparison of numbers computed under different attack pools, and therefore suggestive rather than demonstrative of a robustness advantage.

GADoT [4] achieves F1 > 0.98 on a single attack family with no strategy-distribution gap measurement. Table 32 is explicitly *indicative*, not a controlled head-to-head comparison: entries differ in protocols evaluated, preprocessing pipeline, feature space, and attack budget, and the numbers should not be interpreted as a performance ranking. A controlled comparison would require reimplementing representative baselines (unified Transformer, fixed-PGD adversarial training, random multi-attack training, EAT) under the same splits, feature representation, perturbation budgets, attack iterations, and evaluation sample sizes; this reimplementation is left as future work, and until it is done, no superiority claim relative to prior systems is made.

6 Conclusion and Future work

This paper presents HADES, the first network intrusion detection framework to jointly address three challenges that prior systems handle separately: robustness against a strategy-adaptive adversary, cost-efficient multi-tier routing, and protocol-specialized detection. Each pillar targets a distinct term of the minimax surrogate (Eq. (10)) and delivers measurable, independently verifiable gains.

Pillar 1—Protocol stratification yields a confounded mean advantage $\Delta\Phi = +0.043$ over the unified baseline (an upper bound on the pure effect; the matched-condition effect is +0.0003 and not statistically significant), with gains concentrated on the most heterogeneous protocols (TCP_OTHER: +0.21; SSH: +0.03).

Pillar 2—Co-evolutionary hardening achieves a clean and adversarial F1 of 0.9998 at $\varepsilon = 0.1$ over nine white-box attack strategies, with worst-case portfolio robustness $\mathcal{R}_{\text{portfolio}} = 99.92\%$, and reduces the strategy-distribution gap by up to 74× on DNS and 39× on SSH. At higher budgets ($\varepsilon = 0.5$), HADES maintains a +0.193 F1 advantage over EAT trained at $\varepsilon = 0.30$; budget-matched baselines trained directly at $\varepsilon = 0.50$ remain stronger at that fixed budget (Section 5.2).

Pillar 3—Confidence-gated cascade holds traffic-weighted mean inference cost at 5.4 ms on clean traffic, routing five of six protocols entirely through Tier 1, and delivers a 33.4× speedup over full Tier-2 routing under adversarial inputs.

All results carry 95% bootstrap confidence intervals (max width 0.0007) on CIC-IDS-2018. Two limitations constrain the scope of these claims: the threat model is *white-box only*, and all evaluations are on a single controlled-environment dataset. Critically, zero-shot transfer to CIC-IDS-2017 collapses performance (mean $\Delta_{\text{total}} = -0.96$; Section 5.7), with 98% of the gap attributable to traffic-distribution shift rather than normalizer mismatch; HADES does not generalize across datasets without domain adaptation and should presently be regarded as a within-distribution defense. The reported F1-score values are worst-case bounds within the ℓ_∞ ball and should not be interpreted as guarantees against physically craftable adversarial traffic or under real-world distribution shift. Three further limitations, surfaced by the extended round-2 evaluation, qualify the scope of these claims. First, feasibility-projected robustness (Section 5.5) holds at $\varepsilon \leq 0.1$ across all protocols but breaks down materially on HTTP/TCP_OTHER at $\varepsilon = 0.3$ (up to −0.215 F1 under PGD-20), so the ℓ_∞-ball robustness claim should not be extrapolated to high budgets on heterogeneous strata without a constraint-aware attack. Second, the Exp3 rate amplifier $c = 3$ shows a non-zero training-collapse frequency (1/6 short-horizon runs) that is more stable at $c = 1$ (Section 5.2); the deployed pipeline's checkpoint-selection safeguard bounds, but does not eliminate, this risk. Third, LoRA-only target-domain fine-tuning (Section 5.7) fails to remediate cross-dataset collapse even at 10% labeled data, and the FTP stratum's near-perfect F1 is partly attributable to low behavioral diversity rather than pure generalization (Section 5.6).

The following eleven research directions are identified, extending HADES toward production deployment and stronger formal foundations, ordered by operational consequence.

1. **Black-box adaptive adversaries**: The Red-Blue loop uses only white-box attacks. Extending the Red Agent to query-based strategies (e.g., Square Attack [32]) is necessary for the robustness guarantee to hold against realistically deployed adversaries without gradient access.
2. **Cross-dataset generalization**: Zero-shot transfer to CIC-IDS-2017 degrades F1 by 0.96 on average, with 98% attributable to traffic-distribution shift. Closing this gap requires domain-invariant representation learning via IRM [33] or DANN [34], with validation on UNSW-NB15 and CICIOT-2023.
3. **Intra-protocol heterogeneity**: The catch-all strata (`TCP_OTHER`, `UDP_OTHER`) aggregate multiple application-layer protocols, violating the distributional coherence of $\mathcal{D}_p$. The gate-mode collapse (F1 = 0.0008) confirms underlying multi-modality. Finer sub-protocol clustering or a mixture-of-experts detector with a learned gating function is the natural extension.
4. **Joint optimization of ρ and f**: The current decomposition hardens **f** first and calibrates ρ post-hoc. A fully coupled solution via Gumbel-softmax [39] or the straight-through estimator [40] would close the gap between Eq. (9) and the operational surrogate (10).
5. **Calibrated epistemic uncertainty for routing**: The confidence proxy $u_p(x)$ can be miscalibrated on adversarially crafted inputs where $f_p(x)$ is confidently incorrect. Replacing it with deep ensemble variance [41] or conformal scores [42] would reduce under-escalation at a modest latency cost.
6. **Gradient obstacle on the F1 objective**: The hard classifier $h_{f_p,\tau}$ makes Eq. (9) non-differentiable. Randomized smoothing [13] would provide differentiable lower bounds, though adapting ℓ_2 certificates to the ℓ_∞ perturbation model of network flows is non-trivial.
7. **Two-player regret bound for the Red-Blue loop**: The Exp3 bound assumes a passive reward environment, but the Blue Agent actively modifies $f_p^{(t)}$ in response to Red Agent selections, making rewards endogenous. A formal treatment via no-regret dynamics in zero-sum games [43,44] is required to give Pillar 2 a fully game-theoretic guarantee.
8. **Constraint-aware feasibility attack**: The feasibility projection of Section 5.5 enforces only per-feature empirical range, not integer packet counts, duration–rate consistency, forward/backward-flow relationships, or TCP-flag logic. A raw-feature-space attack respecting these relational constraints is required to close the gap exposed at $\varepsilon = 0.3$ on `HTTP/TCP_OTHER` (Section 5.5).
9. **Controlled baseline reimplementation**: Table 32 is an indicative, cross-paper comparison. Reimplementing unified-Transformer, fixed-PGD, random multi-attack, and EAT baselines under matched splits, features, attack budgets, and sample sizes is required before any comparative robustness claim relative to prior systems can be made.
10. **Validation on contemporary traffic corpora**: The evaluation is confined to CIC-IDS-2018 for the reasons given in Section 5, but the protocol mix of that capture predates the current prevalence of encrypted, IoT, and 5G/6G traffic. Re-instantiating the six-stratum partition on recent corpora (CIC-IoT-2023, UNSW-NB15, and successor CIC releases), and re-running the full three-pillar pipeline there, is required before any claim of contemporary operational relevance. Recent CMES work on split-attention Transformer detection [17] and Wasserstein-GAN/autoencoder IoT detection [12] provides natural comparison points for that re-evaluation.
11. **Leakage-controlled retraining**: The mutual-information screening of Section 5.6 identifies near-deterministic features on every stratum, most starkly `Dst Port` on `TCP_OTHER`. Retraining each detector with its top-MI features ablated, and reporting the resulting F1 drop, would separate the share of measured performance carried by genuine multivariate flow structure from the share carried by a small number of label-proxy columns.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Abdelouahid Derhab: conceptualization and methodology, writing—original draft; Adlen Kerboua: software, investigation, data curation, formal analysis, writing—original draft; Noureddine Seddari: formal analysis, writing—review and editing; Anis Haniche: conceptualization, methodology, formal analysis; Mohammad Mehedi Hassan: writing—review and editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: CIC-IDS-2018 is publicly available from the Canadian Institute for Cybersecurity at https://www.unb.ca/cic/datasets/ids-2018.html. Processed protocol-stratified splits and pre-trained model checkpoints will be deposited in a public repository (Zenodo or equivalent) and assigned a persistent DOI upon acceptance. The full implementation is provided as Data at https://github.com/aderhab/HADES. The architecture is fully reproducible from a single Jupyter notebook; minor run-to-run variance due to GPU non-determinism is reported via the bootstrap confidence intervals.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ennaji S, De Gaspari F, Hitaj D, Kbidi A, Mancini LV. Adversarial challenges in network intrusion detection systems: research insights and future prospects. IEEE Access. 2025;13:148613–45. doi:10.1109/ACCESS.2025.3600984.
2. Tafreshian B, Zhang S. A defensive framework against adversarial attacks on machine learning-based network intrusion detection systems. In: Proceedings of the IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2024 Dec 17; Sanya, China. p. 2436–41. doi:10.1109/TrustCom63139.2024.00337.
3. Yassine HAK, Saleh ME, Nakhal BE, Chakik AE. Centralized two-tiered tree-based intrusion-detection system (C2T-IDS). IoT. 2025;6(4):67. doi:10.3390/iot6040067.
4. Abdelaty M, Scott-Hayward S, Doriguzzi-Corin R, Siracusa SD. GADoT: GAN-based adversarial training for robust DDoS attack detection. In: Proceedings of the IEEE Conference on Communications and Network Security (CNS); 2021 Oct 4–6; Tempe, AZ, USA. p. 119–27. doi:10.1109/CNS53000.2021.9705040.
5. Affan AM. Evolving adversarial training (EAT) for AI-powered intrusion detection systems. Am J Comput Sci Technol. 2024;7(3):115–21. doi:10.11648/j.ajcst.20240703.16.
6. Awad Z, Zakaria M, Hassan R. An enhanced ensemble defense framework for boosting adversarial robustness of intrusion detection systems. Sci Rep. 2025;15(1):14177. doi:10.1038/s41598-025-94023-z.
7. Niknami N, Mahzoon V, Wu J. Meta-IDS: a multi-stage deep intrusion detection system with optimal CPU usage. In: Proceedings of the IEEE International Conference on Meta Computing (ICMC); 2024 Jun 20–23; Qingdao, China. p. 74–83. doi:10.1109/ICMC60390.2024.00015.
8. Aghaei E, Niu X, Shadid W, Al-Shaer E. SecureBERT: a domain-specific language model for cybersecurity. In: Security and Privacy in Communication Networks: 18th EAI International Conference. Cham, Switzerland: Springer; 2023. p. 39–56. doi:10.1007/978-3-031-25538-0_3.
9. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. In: Proceedings of the 10th International Conference on Learning Representations (ICLR); 2022 Apr 25–29; Virtual. doi:10.48550/arXiv.2106.09685.
10. Auer P, Cesa-Bianchi N, Freund Y, Schapire RE. The nonstochastic multiarmed bandit problem. SIAM J Comput. 2002;32(1):48–77. doi:10.1137/S0097539701398375.

11. Zhang H, Yu Y, Jiao J, Xing EP, El Ghaoui L, Jordan MI. Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the 36th International Conference on Machine Learning (ICML); 2019 Jun 9–15; Long Beach, CA, USA. p. 7472–82. doi:10.48550/arXiv.1901.08573.
12. Alshehri MS, Saidani O, Malwi WA, Asiri F, Latif S, Khattak AA, et al. A hybrid Wasserstein GAN and autoencoder model for robust intrusion detection in IoT. Comput Model Eng Sci. 2025;143(3):3899–920. doi:10.32604/cmes.2025.064874.
13. Cohen J, Rosenfeld E, Kolter Z. Certified adversarial robustness via randomized smoothing. In: Proceedings of the 36th International Conference on Machine Learning (ICML); 2019 Jun 9–15; Long Beach, CA, USA. p. 1310–20. doi:10.48550/arXiv.1902.02918.
14. Elkan C. The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI); 2001 Aug 4–10; Seattle, WA, USA. p. 973–8. doi:10.5555/1642194.1642224.
15. Adjewa F, Esseghir M, Merghem-Boulahia L. LLM-based continuous intrusion detection framework for next-gen networks. In: Proceedings of the 2025 International Wireless Communications and Mobile Computing (IWCMC); 2025 May 12–16; Abu Dhabi, United Arab Emirates. p. 1198–203. doi:10.1109/IWCMC65282.2025.11059643.
16. Sana L, Nazir MM, Yang J, Hussain L, Chen YL, Ku CS, et al. Securing the IoT cyber environment: enhancing intrusion anomaly detection with vision transformers. IEEE Access. 2024;12:82443–68. doi:10.1109/ACCESS.2024.3404778.
17. Zhu G, Yu Y, Deng X, Dai Y, Li Z. A hybrid split-attention and transformer architecture for high-performance network intrusion detection. Comput Model Eng Sci. 2025;145(3):4317–48. doi:10.32604/cmes.2025.074349.
18. Yang J, Govindarajan V, Por LY, Shaikh ZA, Xin Q, Bhattacharya P, et al. DDoS attack detection in consumer IoT-based healthcare systems using improved off-policy proximal policy optimization and generative adversarial network. IEEE Trans Consum Electron. 2025;71(4):11662–75. doi:10.1109/TCE.2025.3605098.
19. Khan AY, Latif R, Latif S, Tahir S, Batool G, Saba T. Malicious insider attack detection in IoTs using data analytics. IEEE Access. 2020;8:11743–53. doi:10.1109/ACCESS.2019.2959047.
20. Mallikarjun A, Patro P. Hybrid data balancing with MLP probabilities-based categorical boosting model for robust intrusion detection system in IoT environment. Syst Soft Comput. 2026;8(16):200443. doi:10.1016/j.sasc.2026.200443.
21. Ragunthar T, Christila SA, Jyoshna B, Meenachi L, Das S, Patro P. Machine learning-driven intrusion detection systems for anomaly detection in vehicular ad-hoc networks. Trans Emerg Telecommun Technol. 2026;37(2):e70363. doi:10.1002/ett.70363.
22. Cantone M, Marrocco C, Bria A. Machine learning in network intrusion detection: a cross-dataset generalization study. IEEE Access. 2024;12:144489–508. doi:10.1109/ACCESS.2024.3472907.
23. Guerziz I, El Houda ZA, Le LB. Enhanced adversarial domain adaptation for intrusion detection systems. In: Proceedings of the 21st International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob); 2025 Oct 20–22; Marrakesh, Morocco. p. 1–6. doi:10.1109/WiMob66857.2025.11257501.
24. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS); 2017 Dec 4–9; Long Beach, CA, USA. doi:10.48550/arXiv.1706.03762.
25. Bubeck S, Cesa-Bianchi N. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Found Trends Mach Learn. 2012;5(1):1–122. doi:10.1561/2200000024.
26. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP); 2018 Jan 22–24; Funchal, Portugal. p. 108–16. doi:10.5220/0006639801080116.
27. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57. doi:10.1613/jair.953.
28. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR); 2015 May 7–9; San Diego, CA, USA. doi:10.48550/arXiv.1412.6572.

29. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6th International Conference on Learning Representations (ICLR); 2018 Apr 30–May 3; Vancouver, BC, Canada. doi:10.48550/arXiv.1706.06083.
30. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, et al. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun 18–22; Salt Lake City, UT, USA. p. 9185–93. doi:10.1109/CVPR.2018.00957.
31. Croce F, Hein M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of the 37th International Conference on Machine Learning (ICML); 2020 Jul 13–18; Virtual. doi:10.48550/arXiv.2003.01690.
32. Andriushchenko M, Croce F, Flammarion N, Hein M. Square attack: a query-efficient black-box adversarial attack via random search. In: Proceedings of the 16th European Conference on Computer Vision (ECCV); 2020 Aug 23–28; Glasgow, UK. p. 484–501. doi:10.1007/978-3-030-58592-1_29.
33. Lin Y, Dong H, Wang H, Zhang T. Bayesian invariant risk minimization. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; New Orleans, LA, USA. p. 16021–30. doi:10.1109/CVPR52688.2022.01555.
34. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, et al. Domain-adversarial training of neural networks. J Mach Learn Res. 2016;17(1):59:1–59:35. doi:10.48550/arXiv.1505.07818.
35. Hnamte V, Hussain J. Dependable intrusion detection system using deep convolutional neural network: a novel framework and performance evaluation approach. Telemat Inform Rep. 2023;11(9):100077. doi:10.1016/j.teler.2023.100077.
36. Bouidaine AB, Moussaoui D, Hadjila M, Ferhi W, Hachemi MH. Deep learning-based anomaly and intrusion detection using the CSE-CIC-IDS2018 dataset. Eng Technol Appl Sci Res. 2025;15(4):24782–7. doi:10.48084/etasr.11173.
37. Long Z, Yan H, Shen G, Zhang X, He H, Cheng L. A transformer-based network intrusion detection approach for cloud security. J Cloud Comput. 2024;13(1):5. doi:10.1186/s13677-023-00574-9.
38. Omotayo OS, Abiodun O, Mebawondu JO, Akinduyite OC. An ensemble models to detect intrusion using CIC-IDS-2017 and CIC-IDS-2018 as benchmarks. ISA J Multidiscip. 2025;2(5):14–23. doi:10.5281/zenodo.17073568.
39. Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-softmax. In: Proceedings of the 5th International Conference on Learning Representations (ICLR); 2017 Apr 24–26; Toulon, France. doi:10.48550/arXiv.1611.01144.
40. Lee DH, Zhang S, Fischer A, Bengio Y. Difference target propagation. In: Machine learning and knowledge discovery in databases. Cham, Switzerland: Springer International Publishing; 2015. p. 498–515. doi:10.1007/978-3-319-23528-8_31.
41. Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS); 2017 Dec 4–9; Long Beach, CA, USA. doi:10.48550/arXiv.1612.01474.
42. Angelopoulos AN, Bates S. Conformal prediction: a gentle introduction. Found Trends Mach Learn. 2023;16(4):494–591. doi:10.1561/2200000122.
43. Freund Y, Schapire RE. Adaptive game playing using multiplicative weights. Games Econ Behav. 1999;29(1–2):79–103. doi:10.1006/game.1999.0738.
44. Lockhart E, Lanctot M, Pérolat J, Lespiau JB, Morrill D, Timbers F, et al. Computing approximate equilibria in sequential adversarial games by exploitability descent. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI); 2019 Aug 10–16; Macao, China. p. 464–70. doi:10.24963/ijcai.2019/66.