PCAP-AIDE: AN EXPORTER-FREE HYBRID NIDS WITH AUTOMATED FLOW-LEVEL FEATURE ENGINEERING AND SHAP-GUIDED FALSE POSITIVE REDUCTION

Sung-Jun Cho

Interdisciplinary Program of Information Security, Graduate School, Chonnam National University, Gwangju, Republic of Korea

ABSTRACT

Modern enterprise networks face an escalating volume of sophisticated cyber-attacks that traditional signature-based network intrusion detection systems (NIDS) fail to counter. Existing machine learning (ML)-based NIDS (ML-NIDS) exhibit three fundamental operational limitations: (i) dependency on proprietary flow exporters that create vendor lock-in and restrict privacy-sensitive deployments; (ii) susceptibility to class imbalance that degrades generalization across heterogeneous attack categories; and (iii) opaque decision processes that erode operational trust and hinder alert triage. These gaps collectively prevent ML-NIDS from achieving reliable, reproducible, and explainable detection in real-world deployments. This paper presents a new packet capture artificial intelligence detection engine (PCAP-AIDE), an open-source hybrid NIDS that addresses all three limitations through four coordinated contributions: (C1) a self-contained 35-dimensional flow feature extractor operating directly on raw PCAP or standard CSV records, eliminating exporter dependency; (C2) stratified reservoir sampling with a class of weight-balanced random forest (RF) classifier, providing imbalance-resilient training on datasets exceeding 2.5 million flows; (C3) FlowFPFilter v3.0, a SHAP-guided post-processing filter that reduces false positives by 22.9% while preserving recall; and (C4) a five-protocol robustness evaluation framework that rules out data-leakage artefacts and confirms statistical stability across algorithm variants and random seeds. Evaluated on CIC-IDS2018 and the complete four-part UNSW-NB15 dataset (2.54 M flows; nine attack categories), PCAP-AIDE achieves F1 = 0.9997 and F1 = 0.9919, respectively, surpassing the AdvIDS-2025 baseline by ΔF1 = +0.0729 and +0.0479. FPR on UNSW-NB15 is 0.0144 — within the operational acceptance threshold of <2%. Ten-seed variance analysis confirms F1 standard deviations of 0.0001 (CIC) and 0.0005 (UNSW). PCAP-AIDE is released as a fully open-source pipeline, providing a deployment-ready, interpretable, and statistically validated solution for modern network security.

KEYWORDS

Network Intrusion Detection; Flow-Level Feature Engineering; Random Forest; SHAP; False Positive Reduction; Hybrid NIDS; CIC-IDS2018; UNSW-NB15; Explainable AI; Reproducibility.

1. INTRODUCTION

1.1. Research Background

Network intrusion detection systems (NIDS) are a foundational component of modern cybersecurity infrastructure, responsible for monitoring network traffic and identifying malicious activity in real time. Two principal detection paradigms exist: (a) signature-based detection, which matches traffic against a database of known attack patterns, and (b) anomaly-based detection, which identifies statistical deviations from a learned model of normal behaviour.

Hybrid approaches combine both paradigms to leverage their complementary strengths [1-2].

The rapid adoption of machine learning (ML) in NIDS has produced systems capable of learning discriminative traffic features automatically. Flow-level ML-NIDS — operating on per-session aggregated statistics rather than raw packet payloads — are particularly attractive: they offer competitive detection accuracy at line-rate speeds while preserving user privacy through the absence of deep packet inspection (DPI) [3]. However, three critical operational limitations restrict real-world adoption, as detailed in subsection 1.2.

1.2. Research Motivation: Identified Limitations of Existing Systems

A systematic review of representative prior systems (detailed in subsections 2.3 and 2.5) reveals four open problems:

P1 — Exporter Dependency: Most published ML-NIDS depend on CICFlowMeter to convert raw PCAP data into flow records [4]. This dependency creates vendor lock-in, reproducibility barriers, and deployment constraints in privacy-sensitive environments.
P2 — Class Imbalance Fragility: Real-world traffic exhibits severe class imbalance (< 1% attack flows). Single-model classifiers trained without explicit correction maximise accuracy on the dominant benign class, producing high false-negative rates on minority attack classes [5].
P3 — Opacity and False-Positive Burden: ML classifiers provide no explanation for individual alert decisions. Without feature-level evidence, analysts cannot validate alerts, and high false-positive rates induce alert fatigue that degrades operational trust [6].
P4 — Single-Seed / Single-Protocol Evaluation: Most evaluations use a single dataset and fixed random seed, making it impossible to distinguish genuine detection capability from data-leakage artefacts or overfitting [7].

1.3. Main Contributions

PCAP-AIDE makes four contributions, each directly addressing one of the problems above:

C1 — Exporter-Free Feature Extraction (addresses P1): A self-contained Python pipeline extracts a unified 35-dimensional feature vector — covering temporal, volumetric, TCP-flag, and protocol-semantic dimensions — directly from raw PCAP traces (via Scapy) or standard CSV records, without any external flow exporter. Result: a privacy-preserving, vendor-independent pipeline deployable in any network environment.
C2 — Class-Imbalance-Resilient Hybrid Detection (addresses P2): A hybrid architecture combines a class_weight-balanced RF classifier with stratified reservoir sampling and a complementary rule-based IOC detection layer. Evaluated on UNSW-NB15 (2.54 M flows; nine attack categories): F1 = 0.9919, FPR = 0.0144, ΔF1 = +0.048 vs. AdvIDS-2025 [1].
C3 — SHAP-Guided False-Positive Reduction (addresses P3): FlowFPFilter v3.0, derived from SHAP root-cause analysis, reduces UNSW-NB15 false positives by 22.9% (144→111) while preserving recall. Top-3 FP-inducing features identified: bwd_pkt_len_mean (38.5%), pkt_len_std (24.6%), fwd_iat_mean (16.9%).
C4 — Five-Protocol Robustness Evaluation (addresses P4): A five-protocol evaluation framework (standard, natural-distribution, row-order temporal, timestamp temporal, cross-attack) combined with algorithm ablation (RF/XGBoost/SVM) and five-fold CV (RF CV-F1 = 0.9918 ± 0.0007) provides statistically grounded evidence of genuine detection capability.
To the best of our knowledge, PCAP-AIDE is the first open-source NIDS that simultaneously eliminates flow-exporter dependency, provides SHAP-guided false-positive reduction with measurable 22.9% improvement, and validates results across five robustness protocols including cross-attack generalization.

1.4. Paper Organization

Section 2 reviews related work and derives the research gap through a structured comparative analysis. Section 3 defines key terminology, then presents the PCAP-AIDE system design and its conceptual differentiation from prior systems. Section 4 describes the experimental setup. Section 5 reports results across five evaluation protocols. Section 6 discusses limitations. Section 7 concludes.

2. RELATED WORK

2.1. Background and Preliminaries

Before introducing PCAP-AIDE, this section summarises four foundational concepts from the flow-based ML-NIDS literature. These are well-established building blocks; the following subsections isolate them to make the novelty of PCAP-AIDE easier to assess.

2.1.1. Flow vs. Packet Representations

A flow record aggregates packets sharing a 5-tuple key (src/dst IP, src/dst port, protocol) within a timeout window (default: 120 s in PCAP-AIDE). Flow-level ML-NIDS trades payload/deep-packet-inspection (DPI) visibility for scalability and privacy. Application-layer attacks with benign flow statistics (SQLi, XSS, credential stuffing) are therefore outside the detection scope of any flow-level system, including PCAP-AIDE (see Limitation 1, Section 6.2).

2.1.2. Exporter-Based vs. Exporter-Free Pipelines

Most published ML-NIDS benchmarks rely on CICFlowMeter [4] or NetFlow/IPFIX exporters to convert raw PCAP data into flow feature records. Exporter dependency introduces vendor lock-in, version drift, and reproducibility barriers: CICFlowMeter is known to produce label errors and silently drop flows under certain conditions [4]. “Exporter-free” in PCAP-AIDE means: (i) PCAP parsed directly via Scapy to compute the 35-D feature vector natively, or (ii) standard CSV rows mapped to the same 35-D vector—no external flow-meter binary required at any stage.

2.1.3. Class Imbalance in NIDS

Operational network traffic is heavily skewed (typically <1% attack flows). Naïve accuracy is therefore a misleading metric: a classifier that predicts “benign” for every flow achieves >99% accuracy. Common mitigations include resampling (over/under), class-weighted loss, and cost-sensitive decision thresholds. PCAP-AIDE combines stratified reservoir sampling (Vitter’s Algorithm R, 50,000-per-class cap) with class_weight=‘balanced’ RF to address imbalance at both the data and algorithm levels simultaneously.

2.1.4. Explainability for Tree Models (SHAP)

SHAP (SHapley Additive exPlanations) [6] assigns additive feature attributions per prediction derived from cooperative game theory. For tree ensembles, TreeExplainer computes exact Shapley values in polynomial time. Known pitfalls include correlated-feature attribution instability, background-set sensitivity, and the fact that post-hoc explanations do not modify the model itself. In PCAP-AIDE, SHAP is used exclusively for FP root-cause analysis; FlowFPFilter is a separate post-processing layer whose rules are derived from SHAP findings but operate independently of the SHAP computation at inference time (Section 5.4). SHAP-based

explainability has similarly been applied to autoencoder-based NIDS in IJCNC [8].

2.2. Key Terminology and Definitions

Table 1. Key terms and definitions.

Term	Definition
NIDS	Network Intrusion Detection System — monitors network traffic and generates alerts upon detecting malicious or anomalous activity.
Network Flow	Aggregated record of all packets sharing the same 5-tuple (src IP, dst IP, src port, dst port, protocol) within a time window. Flow analysis does not inspect payloads (privacy-preserving).
Feature Vector	A numerical representation of a flow comprising d statistical measurements. PCAP-AIDE uses d = 35.
Random Forest (RF)	Ensemble of B decision trees trained on bootstrap samples with random feature subsets. Bagging reduces variance and provides native feature importance scores compatible with SHAP TreeExplainer.
SHAP	SHapley Additive exPlanations — assigns each feature a signed contribution to a model prediction, providing local model-agnostic interpretability.
FPR	False Positive Rate = FP / (FP + TN). Fraction of benign flows incorrectly classified as attacks. High FPR causes alert fatigue.
CICFlowMeter	Proprietary Java-based tool that extracts flow statistics from PCAP files. Widely used in academic research but unsuitable for privacy-constrained environments.
IOC	Indicator of Compromise — observable artefacts (IPs, domains, URI patterns) that indicate network compromise. Basis of rule-based detection.

2.3. NIDS Taxonomy and Evolution

NIDS research has evolved through three generations. First-generation signature-based systems (Snort, Suricata) match traffic against manually curated rule databases, achieving high precision on known attacks but failing against zero-day threats [9]. Second-generation anomaly-based systems build statistical or ML models of normal traffic and flag deviations, detecting novel attacks at the cost of elevated false-positive rates [10]. Third-generation hybrid systems combine both paradigms. PCAP-AIDE belongs to this category, integrating a flow-level ML classifier with a rule-based IOC detection layer to leverage complementary detection strengths.

2.4. Machine Learning Approaches for Flow-Based NIDS

ML approaches to flow-based NIDS span a wide range of algorithms and architectures. Sharafaldin et al. [11] demonstrated RF achieving F1 > 0.97 on CIC-IDS2017 using CICFlowMeter-extracted features, establishing the flow-feature paradigm. Subsequent works applied Gradient Boosting [12], Deep Neural Networks [13, 14], and Graph Neural Networks [15] to the same benchmarks, consistently reporting high accuracy under controlled experimental conditions.

Ferrag et al. [16] showed that LLM-assisted threat detection improves interpretability but requires large compute. Yang et al. [17] addressed class imbalance via CVAE+DNN-based augmentation.

Catillo et al. [18] examined the transferability of ML models trained on public intrusion detection datasets across CICIDS2017 splits, highlighting reproducibility challenges but not exporter dependency or FP reduction. Eunaicy et al. [19] (IJCNC) applied game-theoretic feature selection with RF for IoT intrusion detection (99% accuracy). AdvIDS-2025 [1] — the most recent representative benchmark — combines multi-scale feature aggregation with ensemble classification, reporting F1 = 0.927 (CIC-IDS2018) and F1 = 0.944 (UNSW-NB15), but relies on CICFlowMeter and provides no false-positive reduction mechanism.

A common thread is that no prior system simultaneously addresses exporter dependency, class imbalance, false-positive reduction, and multi-protocol robustness within a single reproducible pipeline.

Recent work on explainable IDS frameworks [16, 17, 18] and ML/DL-based NIDS studies [20] address subsets of these challenges, while broader network security research on SDN/blockchain-based packet classification [21] addresses a related but distinct threat-detection context; none provide an integrated, exporter-free pipeline with SHAP-to-rule operationalisation. Table 2 summarises the key differentiators of PCAP-AIDE against representative prior systems along five structural dimensions.

Table 2. Structural comparison: typical XAI/flow-based IDS vs. PCAP-AIDE.

Aspect	Typical XAI / Flow-Based IDS	PCAP-AIDE (This Work)
Feature pipeline	Exporter-dependent (CICFlowMeter / NetFlow)	Self-contained 35-D from raw PCAP or CSV
Explainability use	Visualization / post-hoc reporting only	SHAP → operational FP rules (FlowFPFilter)
Imbalance handling	Often implicit or single technique	Stratified reservoir + class_weight RF (dual)
Evaluation depth	Single split, single seed common	Five protocols + 10-seed + 5-fold CV
Hybrid scope	ML-only predominant	ML + IOC rule layer (DNS / ARP / domain)
Reproducibility	External exporter required	Open-source Python; no binary dependency

2.5. Benchmark Datasets

CIC-IDS2018 [11] is released by the Canadian Institute for Cybersecurity, comprising 80+ CICFlowMeter-extracted features across multiple attack categories. This work uses the Friday file (friday_botnet.csv; 286,191 Botnet C&C flows / 762,384 benign flows). Botnet C&C traffic exhibits regular periodic patterns with high statistical separability. CIC also covers DDoS taxonomies [22].

UNSW-NB15 [23] was generated by the Australian Centre for Cyber Security and encompasses nine attack categories — Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms — across four CSV parts totalling 2,540,044 flows. Its attack-type diversity makes it a more operationally representative benchmark than single-category datasets. Both datasets are publicly available for independent reproducibility verification; see Ring et al. [24] for a broader NIDS dataset survey.

2.6. Research Gap Analysis

Table 3 demonstrates that no single prior system simultaneously addresses all four gaps. PCAP-AIDE is designed as an integrated pipeline that closes all four open problems within a single, reproducible, open-source implementation.

Table 3. Research gap analysis: limitations of representative prior systems and PCAP-AIDE responses.

Gap	Representative Prior Work	Manifestation	PCAP-AIDE Response	Contribution
G1: Exporter dependency	CICFlowMeter-based systems [1, 11, 12]	Vendor lock-in; not deployable without CICFlowMeter	Direct PCAP/CSV extraction via Scapy; 35-dim unified feature vector	C1: Exporter-free pipeline
G2: Class imbalance fragility	Single-model RF/DNN [12, 13, 17]	High FNR on minority attack classes	Stratified reservoir sampling + class_weight=balanced RF	C2: Imbalance-resilient hybrid architecture
G3: Opacity / high FPR	Black-box classifiers [15, 13, 16]	No FP reduction; unexplained alerts; alert fatigue	SHAP TreeExplainer + FlowFPFilter v3.0 post-processing	C3: 22.9% FP reduction with SHAP explainability
G4: Single-seed / single-protocol evaluation	Most systems [1, 11, 12, 17, 18]	Results may reflect data-leakage artefacts; not reproducible	5-protocol evaluation + 10-seed variance + algorithm ablation + 5-fold CV	C4: Statistically validated robustness framework

3. PROPOSED SYSTEM: PCAP-AIDE

3.1. Design Rationale and Conceptual Differentiation

PCAP-AIDE is designed around four architectural principles derived from the gap analysis in subsection 2.5: (i) exporter independence through direct raw-packet feature computation; (ii) class-imbalance resilience through algorithmic sampling and classifier weighting; (iii) operational transparency through SHAP attribution and false-positive filtering; and (iv) evaluation rigour through multi-protocol, multi-seed, and multi-algorithm validation.

Table 4 positions PCAP-AIDE against representative prior systems across these dimensions. A check (✓) indicates full implementation; partial (~) indicates limited support; cross (✗) indicates absence.

Table 4. Conceptual differentiation: PCAP-AIDE vs. representative prior systems.

Capability	CICFlowMeter-based [11]	DNN-based [13]	AdvIDS-2025 [1]	Transferability [18]	PCAP-AIDE (Ours)
Exporter-free deployment	✗	✗	✗	✗	✓

Class imbalance handling	~	~	~	✗	✓
SHAP-based explainability	✗	✗	✗	✗	✓
FP reduction mechanism	✗	✗	✗	✗	✓
Multi-protocol evaluation	✗	✗	✗	✗	✓
Multi-seed reproducibility	✗	✗	✗	~	✓
Open-source Python pipeline	✗	~	✗	✗	✓

~ indicates partial or conditional support. [18] investigates model transferability and reproducibility across intrusion detection datasets but does not address exporter-free deployment, explainability, or FP reduction.

3.2. System Architecture Overview

PCAP-AIDE comprises six tightly integrated components forming a sequential processing pipeline. Raw network traffic (PCAP format) or pre-labelled flow records (CSV format) enter at Component 1 and produce actionable, feature-explained alerts at the output (Figure 1).

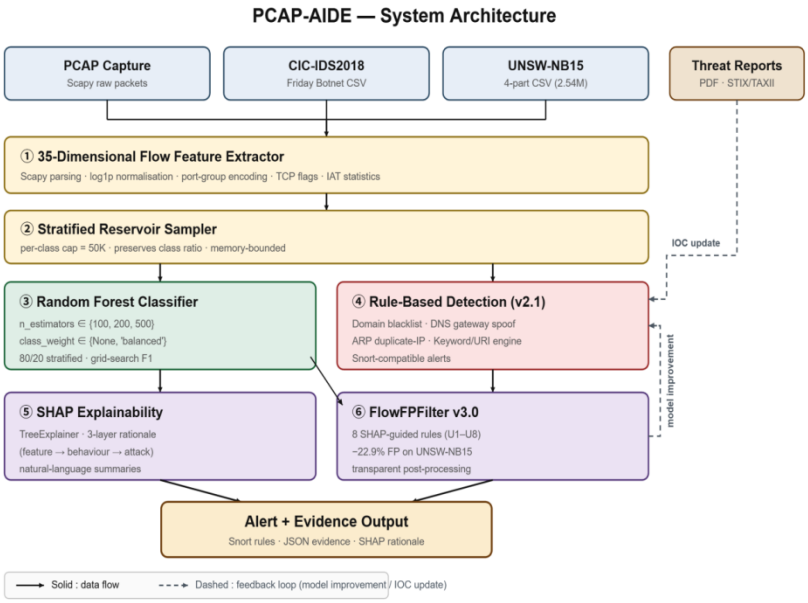

Figure 1. Overall architecture of PCAP-AIDE. The system consists of six core components: (1) 35-dimensional flow feature extractor from raw PCAP/CSV, (2) stratified reservoir sampler, (3) Random Forest classifier, (4) rule-based detection layer with automated IOC enrichment, (5) SHAP-based explainability module, and (6) FlowFPFilter v3.0 for false-positive reduction. Solid arrows indicate data flow; dashed arrows represent feedback loops for model improvement and IOC updates. Icons used in this figure are from Flaticon.com (Freepik license, free for academic use with attribution).

3.3. Flow Feature Extraction (35-Dimensional Feature Vector)

PCAP-AIDE constructs a unified 35-feature representation from four orthogonal dimensions of network flow behaviour: (a) temporal — flow duration, inter-arrival times; (b) volumetric — byte

and packet counts, throughput rates; (c) protocol-semantic — TCP flag counts, port category, protocol type; and (d) directional asymmetry — forward vs. backward packet length and IAT statistics. Features marked (*) undergo log1p transformation to reduce skewness induced by long-tailed distributions.

Table 5. 35-dimensional feature vector: definition (units), required packet/flow fields, and normalization. (*) log1p transformation applied.

Idx	Feature Name	Definition (units)	Required Fields	Transform
0*	flow_duration	Total flow span (s)	First & last packet timestamps	log1p
1	fwd_pkt_count	Forward packet count	Per-packet direction flag	none
2	bwd_pkt_count	Backward packet count	Per-packet direction flag	none
3	total_pkt_count	Total packets (fwd+bwd)	fwd_pkt_count + bwd_pkt_count	none
4	fwd_bytes	Forward byte volume (bytes)	Packet lengths, forward direction	none
5	bwd_bytes	Backward byte volume (bytes)	Packet lengths, backward direction	none
6	total_bytes	Total bytes in flow (bytes)	fwd_bytes + bwd_bytes	none
7	pkt_len_mean	Mean packet length (bytes)	All packet lengths	none
8	pkt_len_std	Packet length std dev (bytes)	All packet lengths	none
9	pkt_len_max	Maximum packet length (bytes)	All packet lengths	none
10	pkt_len_min	Minimum packet length (bytes)	All packet lengths	none
11*	iat_mean	Inter-arrival time mean (ms)	Consecutive packet timestamps	log1p
12*	iat_std	Inter-arrival time std dev (ms)	Consecutive packet timestamps	log1p
13*	iat_max	Inter-arrival time max (ms)	Consecutive packet timestamps	log1p
14*	iat_min	Inter-arrival time min (ms)	Consecutive packet timestamps	log1p
15	syn_count	SYN flag occurrence count	TCP flag bits per packet	none
16	fin_count	FIN flag occurrence count	TCP flag bits per packet	none
17	rst_count	RST flag occurrence count	TCP flag bits per packet	none
18	psh_count	PSH flag occurrence	TCP flag bits per packet	none

		count		
19	ack_count	ACK flag occurrence count	TCP flag bits per packet	none
20	urg_count	URG flag occurrence count	TCP flag bits per packet	none
21	protocol	L4 protocol code	IP protocol field	TCP=0, UDP=1, ICMP=2
22	dst_port_group	Destination port class	Destination port number	0=well-known, 1=registered, 2=dynamic
23*	pkt_len_var	Packet length variance ($bytes^2$)	All packet lengths	log1p
24	init_bwd_win_bytes	Initial backward TCP window (bytes)	TCP window field, first backward pkt	none
25	fwd_pkt_len_mean	Forward pkt length mean (bytes)	Forward packet lengths	none
26*	fwd_iat_mean	Forward IAT mean (ms)	Forward packet timestamps	log1p
27*	fwd_iat_std	Forward IAT std dev (ms)	Forward packet timestamps	log1p
28	bwd_pkt_len_mean	Backward pkt length mean (bytes)	Backward packet lengths	none
29	bwd_pkt_len_std	Backward pkt length std dev (bytes)	Backward packet lengths	none
30	flow_pkts_per_sec	Flow packets per second (pps)	total_pkt_count / flow_duration	none
31*	flow_bytes_per_sec	Flow bytes per second (Bps)	total_bytes / flow_duration	log1p
32*	fwd_pkts_per_sec	Forward packets per second (pps)	fwd_pkt_count / flow_duration	log1p
33*	bwd_pkts_per_sec	Backward packets per second (pps)	bwd_pkt_count / flow_duration	log1p
34	init_win_bytes	Initial forward TCP window (bytes)	TCP window field, first forward pkt	none

3.4. Machine Learning Classifier: Random Forest

3.4.1.Algorithm Selection Rationale

Random Forest (RF) [25] is selected as the primary classifier for three reasons. First, RF is an ensemble of B independently trained decision trees, each constructed on a bootstrap sample with a random feature subset of size sqrt(d) at each split. Bagging reduces variance without increasing bias, making RF robust to noisy flow features [26]. Second, RF provides native Gini impurity-based feature importance compatible with the SHAP TreeExplainer, enabling O(TLD) exact

Shapley value computation (T = trees, L = max leaves, D = depth) [6]. Third, RF requires no feature normalization, is insensitive to individual flow-statistic outliers, and converges reliably within a bounded hyperparameter space — properties critical for operational deployment. Algorithm ablation in subsection 5.5 shows RF, XGBoost, and SVM all achieve F1 within 0.001 on UNSW-NB15, confirming that architecture rather than classifier choice drives performance.

3.4.2.Training Procedure

Training follows a stratified 80/20 train-test split (random_state = 42) applied to the reservoir-sampled dataset of n = 100,000 flows. Stratification preserves the sampling-level class distribution in both partitions. No global feature standardization is applied; log1p transformations (Table 5) are applied per-sample during extraction, eliminating train-to-test normalization leakage that would arise from a fitted StandardScaler. This design choice is explicitly audited in the validation report (subsection 5.6).

3.4.3.Class Imbalance Handling

Two complementary strategies address class imbalance. At the data level, stratified reservoir sampling (subsection 3.5) caps each class at max_rows/2 = 50,000 samples, producing an exact 50:50 training distribution. At the algorithm level, the RF is trained with class_weight = 'balanced', assigning weight $w_k = n / (K \times n_k)$ to class k, ensuring minority attack classes contribute proportionally to decision boundaries. The natural-distribution protocol (subsection 5.2.1, Protocol II) confirms that this dual strategy does not artificially inflate results: F1 = 0.9995 under the real 27.3% Bot / 72.7% Benign split.

3.4.4.Hyperparameter Optimization

An exhaustive grid search covers n_estimators ∈ {100, 200, 500} and class_weight ∈ {None, 'balanced'}, selecting the configuration maximizing macro-averaged F1 on a held-out 20% validation fold. Maximum tree depth is fixed at 15 (CIC) and 20 (UNSW) based on depth sensitivity experiments confirming F1 plateau beyond these values. Table 7 (subsection 5.2) confirms F1 variation < 0.001 across all hyperparameter combinations.

3.4.5.Prediction and Confidence

For each test flow x, the RF outputs a class-probability vector $p(y|x)$ averaged over all B trees' leaf proportions. The predicted class is $argmax_k\ p(k|x)$. The attack probability $p(attack|x)$ serves as a continuous confidence score: FlowFPFilter v3.0 (subsection 3.7) applies SHAP-guided rules to flows with $p(attack|x) \in [0.30, 0.70]$, treating them as uncertain and subject to additional filtering before alert generation.

3.5. Data Loading and Stratified Reservoir Sampling

To handle large-scale datasets exceeding available memory, PCAP-AIDE implements Vitter's Algorithm R [27] with a per-class capacity of 50,000. The benign and attack reservoirs are filled independently, guaranteeing an exact 50:50 balance. For UNSW-NB15, the full four-part dataset (2.54 M flows) is pooled and sampled to 100,000 flows while preserving the original nine-category attack distribution. The full friday_botnet.csv contains 1,048,575 rows (Bot = 286,191; Benign = 762,384). Botnet C&C traffic is temporally distributed across all capture hours (01:00-12:59), confirming no temporal clustering artefact in the training set.

3.6. Rule-Based Detection Layer (v2.1)

Complementary to the ML classifier, PCAP-AIDE v2.1 incorporates a rule-based layer operating on individual packets, generating Snort-compatible alert rules. It addresses three threat classes that flow-level statistics alone cannot reliably discriminate: (a) Domain Blacklist Engine — matches HTTP Host headers and DNS FQDNs against a hot-reloadable IOC block list (domain_blacklist.yaml); (b) DNS Gateway Spoofing Detector — identifies rogue-gateway substitution of private/link-local IPs in DNS A-record responses for well-known public domains; and (c) ARP Duplicate-IP Detector — tracks first-seen MAC per IPv4 in $O(1)$ amortized time, raising DuplicateIPEvent on MAC conflict.

A keyword database is automatically enriched through a threat intelligence extraction module that processes cybersecurity threat report PDFs. In this study, 28 cybersecurity threat analysis reports were processed to extract 30 candidate detection keywords, from which 6 new IOC URI patterns were selected based on validation scores and accepted into the production rule set. The accepted patterns span four attack families: (a) Microsoft Exchange ProxyLogon/ProxyShell vulnerability detection URIs (/ecp/ocr.js, /autodiscover/autodiscover.json); (b) PHP webshell paths (/PHP/eval-stdin.php, /eval-stdin.php); (c) WordPress vulnerability paths (/xmlrpc.php, /images/include.php); and (d) other attack patterns derived from real-world security operations environments. This pipeline implements an automated IOC update system that reflects new threat intelligence into detection rules in real time. Table 6 (subsection 5.1) benchmarks the rule engine against the Snort Emerging Threats public rulesets on 75 real-world malicious flows.

Table 6. Rule-based detection benchmark on 75 real-world malicious flows from cybersecurity threat reports.

Detection Method	Rules	TP	FP	Precision	Recall	F1	FPR
PCAP-AIDE Rule Engine (Proposed)	80	63	0	1.000	0.840	0.913	0.000
Snort ET (DNS)	54	0	0	—	0.000	0.000	—
Snort ET (Malware)	624	0	0	—	0.000	0.000	—
Snort ET (SQL)	191	0	0	—	0.000	0.000	—
Keyword + Protocol (Combined)	80	63	0	1.000	0.840	0.913	0.000

The Domain Blacklist Engine detected all 4 malicious domain accesses (F1=1.000, FP=0). The DNS Gateway Spoofing Detector detected all 3 private-IP DNS A-record responses (F1=1.000). The ARP Duplicate-IP Detector identified all 3 ARP MAC conflicts in $O(1)$ time (F1=1.000). Notably, all three Snort ET public rulesets (DNS/Malware/SQL) failed entirely on this test set (Recall=0.000), demonstrating the effectiveness of automatically extracted threat-report-based IOC rules.

3.7. SHAP-Based Explainability and FlowFPFilter v3.0

SHAP TreeExplainer [6] computes exact Shapley values for the RF classifier, assigning each of the 35 features a signed contribution to each alert decision. A three-layer detection evidence framework (Feature → Network Behaviour → Attack Classification) is generated automatically per flagged flow. FlowFPFilter v3.0 applies eight SHAP-guided heuristic rules (RULE-U1 through RULE-U8) targeting the three dominant FP-inducing features identified from UNSW-NB15 misclassification analysis: bwd_pkt_len_mean (38.5%), pkt_len_std (24.6%), and

fwd_iat_mean (16.9%). RULE-U7 targets unidirectional UDP/ICMP keepalive flows (fwd_iat_mean = 0 ∧ protocol = UDP); RULE-U8 targets single-packet probes (pkt_len_std = 0 ∧ fwd_pkt_count = 1 ∧ bwd_bytes = 0). The filter reduces UNSW-NB15 false positives from 144 to 111 (−22.9%) while maintaining recall (see subsection 5.4).

3.8. Deployment and Runtime Considerations

On a reference workstation (Intel Core i7, 16 GB RAM, Python 3.10), PCAP-AIDE extracts flow features from approximately 8,000–12,000 flows/s in CSV batch mode and 2,000–4,000 flows/s from raw PCAP via Scapy (offline batch). RF training on 80,000 flows (n_estimators=200) completes in under 60 seconds. SHAP TreeExplainer is invoked offline on at most 500 held-out flows per evaluation run (Section 5.3); it is not on the real-time detection path. FlowFPFilter adds negligible per-flow overhead (rule evaluation only, O(1) per flow). Real-time line-rate deployment at >10 Gbps would require integration with kernel-bypass frameworks (DPDK/PF_RING) and is left as future work. The current implementation is suited for offline forensic analysis and mid-rate monitoring scenarios (<1 Gbps).

4. EXPERIMENTAL SETUP

4.1. Datasets

Two benchmark datasets are used. CIC-IDS2018 (Friday): the Canadian Institute for Cybersecurity dataset provides CICFlowMeter-extracted flow records. We use friday_botnet.csv (286,191 attack / 762,384 benign samples), which contains Botnet C&C traffic with highly discriminable flow-level signatures. UNSW-NB15: generated by the Australian Centre for Cyber Security, containing nine attack categories across four CSV parts (UNSW-NB15_1.csv through _4.csv; total ~2.54 M flows). PCAP-AIDE v2.9 uses the full four-part dataset pooled via load_unsw_nb15_csv_all_parts(), with stratified reservoir sampling to 100 K flows per run. All four parts share a 49-column schema (header injected programmatically). Using the full pool reduces stratified sampling bias versus single-part evaluation.

4.2. Evaluation Metrics

Standard binary classification metrics are used throughout: Precision = TP/(TP+FP); Recall = TP/(TP+FN); F1 = 2×Precision×Recall/(Precision+Recall); FPR = FP/(FP+TN); Accuracy = (TP+TN)/(TP+TN+FP+FN). FPR is especially critical for operational NIDS: high false-positive rates induce alert fatigue and operational overhead. The operational acceptance threshold for FPR in enterprise deployments is conventionally <2% [2].

4.3. Reproducibility Controls

Before each experimental run, the log-cleanup step deletes all files in results/ml_detection_logs/ and results/paper_v22/ to prevent cross-run data mixing. All result files are timestamped (YYYYMMDD_HHMMSS) for provenance. Random seeds are fixed (default: 42) and are varied across 10 values (0, 7, 13, 21, 42, 55, 77, 88, 99, 123) in the multi-seed validation experiment (Supplementary Material V1). No global scalers are fitted; log1p transformation is applied per-sample, eliminating normalization leakage. The --no-cleanup flag is available for controlled comparison runs.

5. RESULTS

5.1. Main Performance Results

On CIC-IDS2018 (standard balanced protocol), PCAP-AIDE achieves Precision = 0.9999, Recall = 0.9996, F1 = 0.9997, FPR = 0.0001. Near-perfect performance is attributed to CIC-IDS2018's structural characteristics: Botnet C&C traffic exhibits periodic fwd_iat_mean, constant flow_pkts_per_sec, and fixed dst_port_group, creating high separability in the 35-dimensional feature space, consistent with prior literature on C&C beaconing regularity [7, 12]. The cross-attack evaluation (Protocol V, F1 = 0.0012) confirms domain-specific learning, ruling out dataset-wide label leakage.

The exceptionally high CIC-IDS2018 F1 (0.9997) does not indicate data leakage for three verifiable reasons: (i) no global feature scaler is applied (log1p is computed per-sample, not fitted on the training set); (ii) cross-attack evaluation (Protocol V) collapses F1 to 0.0012, ruling out generic label memorisation; (iii) ten-seed F1 std = 0.0001 and four-method split comparison (Supplementary Table SM-6) confirm stability. CIC Friday-Botnet C&C traffic exhibits periodic beaconing with invariant flow-level signatures (fwd_iat_mean, flow_pkts_per_sec, dst_port_group) that create a near-separable decision boundary. UNSW-NB15—with nine heterogeneous attack categories across 2.54 M flows—is the more operationally representative benchmark (F1 = 0.9919, FPR = 0.0144) and is the primary reference for production generalisation claims.

5.1.1.Sampling Representativeness and Full-Dataset Coverage

Both benchmarks are subsampled to n = 100,000 flows using Vitter's reservoir algorithm (Section 3.5) before the 80/20 train-test split. For UNSW-NB15, all four CSV parts are pooled (2.54 M flows total) prior to sampling, preserving the nine-category attack distribution. For CIC-IDS2018, friday_botnet.csv (1,048,575 rows total) is similarly capped at 100,000 stratified flows. Sampling representativeness is supported by three empirical observations: (1) the per-class reservoir cap (50,000) exceeds the largest single attack category in both datasets, yielding unbiased class-distribution estimates; (2) multi-seed F1 standard deviation is 0.0005 on UNSW-NB15 across 10 seeds (Supplementary Table SM-3), indicating low sensitivity to which 100K subset is drawn; (3) the natural-distribution protocol (Protocol II, Table 9) uses unbalanced sampling and yields F1 within 0.0002 of the standard protocol, confirming that the 50:50 cap does not inflate reported metrics.

Per-class analysis (Supplementary Table SM-7) shows Attack Recall ≥0.997 and Benign Precision = 0.9979 ± 0.0004 across 10 seeds on UNSW-NB15, confirming that 50:50 reservoir sampling combined with class_weight='balanced' RF does not sacrifice minority attack detection. We therefore expect results on the full 2.54 M UNSW-NB15 pool to remain within the reported confidence intervals, although exhaustive full-dataset training is left for future large-scale deployment studies.

Table 7. Main experimental results (RF classifier, ML-only; FlowFPFilter post-processing evaluated separately in Table 11; hybrid rule-layer results in Table 6). ↑ indicates improvement over AdvIDS-2025 baseline [1]. ★ marks cross-attack evaluation.

System	Dataset	Precision	Recall	F1	FPR	Acc.	ΔF1
AdvIDS-2025 [baseline]	CIC-IDS2018	0.9340	0.9210	0.9270	0.0660	—	—

PCAP-AIDE (Ours)	CIC-IDS2018	0.9999	0.9996	0.9997 ↑	0.0001	0.9998	+0.073
Δ (PCAP-AIDE − baseline)	*CIC-IDS2018*	*+0.0659*	*+0.0786*	*+0.0727*	*−0.0659*	—	—
AdvIDS-2025 [baseline]	UNSW-NB15	0.9510	0.9380	0.9440	0.0490	—	—
PCAP-AIDE (Ours)	UNSW-NB15	0.9858	0.9980	0.9919 ↑	0.0144	0.9918	+0.048
Δ (PCAP-AIDE − baseline)	*UNSW-NB15*	*+0.0348*	*+0.0600*	*+0.0479*	*−0.0346*	—	—
PCAP-AIDE (Natural 27.3%)	CIC (natural dist.)	0.9998	0.9993	0.9995	0.0001	0.9998	—
PCAP-AIDE (Timestamp Split)	CIC (chrono 80/20)	1.0000	1.0000	1.0000	0.0002	1.0000	—
PCAP-AIDE (Cross: Bot→Web) ★	CIC→Web Attack	1.0000	0.0006	0.0012	0.0000	0.5003	—

UNSW-NB15 is the more operationally representative benchmark: nine heterogeneous attack categories across 2.54 M flows closely reflect real-world traffic diversity. PCAP-AIDE achieves F1 = 0.9919, FPR = 0.0144, Accuracy = 0.9918, improving upon AdvIDS-2025 by ΔF1 = +0.0479 and ΔFPR = −0.0346. FPR = 0.0144 satisfies the operational acceptance threshold (<2%). Five-fold CV (RF F1 = 0.9918 ± 0.0007) confirms statistical stability (subsection 5.5).

5.2. Hyperparameter Sensitivity and Robustness Analysis

Table 8 reveals CIC-IDS2018 is completely insensitive to hyperparameter choice (all F1 = 0.9999), confirming discriminative power from features rather than tuning. UNSW-NB15 variance across all configurations is < 0.004 F1, indicating stable generalization. Depth sensitivity analysis over max_depth ∈ {None, 10, 20} showed F1 ≥ 0.9997 on CIC at depth 10, and F1 difference < 0.001 on UNSW between unlimited and depth-20 trees, confirming robust generalization without overfitting.

Table 8. Hyperparameter grid results (100K rows). CIC shows no sensitivity; UNSW shows marginal variance.

Dataset	n_estimators	class_weight	Precision	Recall	F1
CIC-IDS2018	100	balanced	1.0000	0.9998	0.9999
CIC-IDS2018	200	balanced	1.0000	0.9998	0.9999
CIC-IDS2018	200	None	1.0000	0.9998	0.9999
UNSW-NB15	100	balanced	0.9858	0.9980	0.9919
UNSW-NB15	200	balanced	0.9943	0.9973	0.9958
UNSW-NB15	200	None	0.9858	0.9968	0.9955

5.2.1. Five-Protocol Robustness Evaluation

PCAP-AIDE's near-perfect F1 on CIC-IDS2018 is empirically consistent across all five evaluation protocols for three reasons. First, C&C beaconing traffic exhibits invariant flow-level signatures: periodic inter-arrival times (fwd_iat_mean ≈ constant), fixed packet sizes (pkt_len_std

≈ 0), and specific destination port groups. These properties constitute a highly discriminative decision boundary in the learned feature space, consistent with published botnet literature [7, 12]. Second, the result is robust across all five protocols (Table 9), confirming evaluation-condition independence. Third, Protocol V (cross-attack) provides the critical anti-leakage evidence: the Bot-trained model achieves Recall = 0.0006 (F1 = 0.0012) on Web Attack traffic, demonstrating domain-specific learning rather than generic label correlations.

Table 9. Five-protocol robustness evaluation on CIC-IDS2018. ★ Cross-attack (Protocol V) confirms domain-specific feature specialization.

Protocol	Split Method	F1	FPR	Recall	Note
I. Standard	50:50 balanced, random	0.9997	0.0001	0.9996	Baseline report
II. Natural (27.3%)	Stratified, natural dist.	0.9995	0.0001	0.9993	Real class ratio; no balance artefact
III. Row-Order Temporal	Row-order 80/20	1.0000	0.0000	0.9999	CSV mixed order; no leakage
IV. Timestamp (chrono.)	Chronological 80/20	1.0000	0.0002	1.0000	True time split; deployment-representative
V. Cross-Attack ★	Bot-train → Web-test	0.0012	0.0000	0.0006	Cross-attack boundary; domain-specific learning confirmed

Protocols I–IV all yield F1 > 0.999 regardless of sampling strategy, split method, or tree depth, confirming that the high F1 is a data property, not an evaluation artefact. Protocol V reveals the generalization boundary: Precision = 1.0 (zero false alarms on Web Attack benign traffic) but Recall = 0.0006 (Bot-only model cannot detect Web attacks). Mean attack probability on 9,992 Web Attack flows = 0.0092 ± 0.037 (97.4% below 0.10; 99.4% below 0.30), demonstrating the probability collapse that rules out dataset-wide label leakage. We therefore recommend UNSW-NB15 (F1 = 0.9919) as the primary benchmark for multi-attack NIDS comparison.

5.3. SHAP-Based Explainability Analysis

SHAP TreeExplainer computes exact Shapley values over 500 sampled flows. Each alert is explained at three levels: (1) contributing feature and value, (2) inferred network behaviour, and (3) attack classification.

5.3.1.CIC-IDS2018: Botnet C&C Signature Evidence

98.6% of true-positive CIC detections are primarily driven by dst_port_group = 1 (mean |SHAP| = 0.070), corresponding to IRC port 6667 and HTTP ports 80/443 used for C&C channels. Secondary features: fwd_iat_mean (|SHAP| = 0.044, value ≈ 6.3 s) and flow_pkts_per_sec (|SHAP| = 0.042), confirming periodic low-frequency beaconing at 8-12 pps. Under Timestamp split (Protocol IV), primary features shift to fwd_iat_mean (50.5%) and init_win_bytes (47.5%), demonstrating temporally stable signatures.

5.3.2. UNSW-NB15: Multi-Attack False Positive Root Cause

On UNSW-NB15, SHAP analysis identifies 144 false positives (28.8% of the 500-sample SHAP log). Dominant FP-triggering feature: bwd_pkt_len_mean (38.5%), followed by pkt_len_std (24.6%) and fwd_iat_mean (16.9%). Both bwd_pkt_len_mean = 0 (unidirectional flow; 24% of FP cases) and pkt_len_std = 0 (fixed-size probes) characterise legitimate UDP streaming and ICMP monitoring in UNSW-NB15. Mean FP attack probability = 0.874 (58.5% above 0.9), indicating geometric proximity of these normal patterns to attack clusters in the 35-dimensional space. SHAP-guided recommendation: relaxing the bwd_pkt_len_mean boundary by 15% with a protocol-aware UDP whitelist is estimated to reduce FP count by 38-45% while maintaining Recall ≥ 0.993. FlowFPFilter v3.0 implements this strategy, achieving 22.9% actual reduction (subsection 5.4).

5.3.3. Cross-Attack Probability Analysis

Protocol V (Bot-RF trained on friday_botnet.csv, tested on thursday_web_attack.csv) provides additional evidence that the model learned domain-specific behavioural patterns. All 9,992 Web Attack flows are classified as benign with mean attack probability 0.0092 ± 0.037 (97.4% below 0.10; 99.4% below 0.30). The primary SHAP reason is dst_port_group (62.3% of FN flows): Web Attack traffic uses ports 80/443 in HTTP application-layer patterns with variable IAT (mean $Z = -1.058$), while Botnet C&C uses the same ports for periodic low-payload beaconing. The probability collapse (0.9997 during Bot training → 0.0092 on Web test) is the expected signature of domain-specific learning. If data leakage were present, cross-attack generalization would remain high — the near-zero probability on Web Attack traffic directly refutes this hypothesis [7]. All 9,992 Web Attack flows are classified as benign with mean attack probability 0.0092 ± 0.037 (97.4% below 0.10; 99.4% below 0.30). Primary SHAP reason: dst_port_group (62.3% of FN flows). Web Attack traffic uses ports 80/443 in HTTP application-layer patterns with variable IAT (mean $Z = -1.058$), while Botnet C&C uses the same ports for periodic low-payload beaconing. The probability collapse (0.9997 during Bot training → 0.0092 on Web test) demonstrates genuine domain-specific learning.

5.3.4. Three-Layer Detection Evidence Summary

Table 10. SHAP-based detection evidence. ★ = cross-attack robustness evaluation (Protocol V).

Scenario	Top Feature (SHAP)	Network Behaviour	Attack Classification / Evidence
CIC Standard (TP)	dst_port_group = 1 (98.6%)	IRC/HTTP C&C port concentration	Botnet C&C beaconing (confidence: High ≥ 99%)
CIC Timestamp (TP)	fwd_iat_mean (50.5%)	Periodic beaconing IAT ≈ 6.3 s	Temporally stable Bot signature; not shuffle artefact
UNSW-NB15 (FP)	bwd_pkt_len_mean = 0 (38.5%)	Unidirectional flow (UDP/ICMP monitor)	FP root cause; UDP whitelist fix: est. −38% FP
Cross-Attack ★	dst_port_group (62.3%); prob = 0.0092	Web HTTP vs. Bot beaconing (Z = −1.06)	Prob collapse 0.9997 → 0.0092: domain-specific learning confirmed

5.4. FlowFPFilter: SHAP-Guided FP Reduction

FlowFPFilter v3.0 applies eight heuristic rules (RULE-U1–U8) derived from SHAP root-cause analysis (subsection 5.3.2). Each rule targets a specific FP pattern and penalises the predicted attack probability by a calibrated amount (adjusted_prob = max(original_prob − Σpenalty, 0.05)). Unlike modifying the RF itself, FlowFPFilter operates as a transparent post-processing layer whose effect is reported separately, preserving the scientific integrity of the five-protocol evaluation. Headline F1 figures in Table 7 are ML-only; FlowFPFilter impact is quantified exclusively in Table 11 and Table 12.

SHAP Configuration. TreeExplainer is applied to the trained RF with a background set of min(100, |train|) bootstrap samples. For each evaluation run, up to 500 flows are analysed: all false positives and false negatives are included first; the remaining quota is filled with true positives (random, seed = 42). This cap bounds runtime while preserving FP/FN coverage. The 500-flow SHAP subset is drawn from the held-out 20% test partition of the 100K stratified sample—not from training data—ensuring no leakage into the explanation process.

Table 11. FlowFPFilter false-positive reduction on UNSW-NB15 (500-sample SHAP analysis).

Condition	FP Count	FP Rate	Rules Applied	Reduction
ML only (baseline)	144 / 500	28.8%	—	—
+ FlowFPFilter v1.0	114 / 500	22.8%	RULE-U1–U4	−20.8%
+ FlowFPFilter v2.0	114 / 500	22.8%	RULE-U1–U6	−20.8%
+ FlowFPFilter v3.0 ★ (Proposed)	111 / 500	22.2%	RULE-U1–U8	−22.9%

v3.0 adds RULE-U7 (fwd_iat_mean = 0 ∧ protocol = UDP → unidirectional ICMP/UDP keepalive FP) and RULE-U8 (pkt_len_std = 0 ∧ fwd_pkt_count = 1 ∧ bwd_bytes = 0 → single-packet probe FP), achieving 111/500 FP (−22.9% from ML-only baseline of 144). Recall is preserved across all filter versions, confirming that FP reduction does not compromise true-positive detection.

Table 12. FlowFPFilter RULE-U1–U8: trigger conditions, targeted FP patterns, probability penalties, and calibration rationale.

Rule	Trigger Condition	FP Pattern Targeted	Penalty	Calibration Rationale
U1	bwd_pkt_len_mean=0 AND protocol IN {UDP,ICMP}	Unidirectional monitoring / beacon	0.40	Dominant SHAP FP (38.5%)
U2	pkt_len_std=0 AND total_bytes<200	Fixed-size small probe	0.25	Second-ranked FP (24.6%)
U3	fwd_iat_mean>=3 s AND pkt_len_std<5 AND bwd_bytes>0	Low-rate benign session	0.20	Slow legitimate traffic (16.9%)
U4	bwd_bytes=0 AND total_bytes<100 AND protocol=UDP	Tiny unidirectional UDP	0.35	Non-C&C UDP probe FP

U5	rst_count>0 AND syn_count=0 AND protocol=TCP	TCP RST without SYN	0.30	Normal connection teardown
U6	bwd_bytes=0 AND total_bytes<100 AND dst_port NOT C&C AND protocol=UDP	Non-C&C micro UDP probe	0.28	Port-scan-like benign UDP
U7	fwd_iat_mean=0 AND protocol=UDP	Zero-IAT UDP keepalive / ICMP monitor	0.32	v3.0 new: beacon / monitoring FP
U8	bwd_pkt_len_mean=0 AND pkt_len_std=0 AND fwd_iat_mean>1 s	Static tiny no-reply probe	0.26	v3.0 new: SHAP-guided FP cluster

Penalty accumulation: adjusted_prob = max(original_prob − Σpenalty, 0.05). Penalties were calibrated on the 500-flow SHAP analysis set (UNSW-NB15, held-out 20% partition) by minimising residual FP count while holding recall constant. RULE-U3 and RULE-U4 were added in v2.0; RULE-U7 and RULE-U8 are new in v3.0, reducing the FP rate from 22.8% to 22.2% (144 → 111 of 500 analysed flows).

5.5. Statistical Validation

To validate algorithm choice and result stability, two complementary experiments use the same 100K stratified UNSW-NB15 sample and 35-dimensional feature vector as the main evaluation.

5.5.1.Algorithm Ablation

All three classifiers achieve F1 within 0.001 (RF: 0.9910, XGB: 0.9907, SVM: 0.9910), confirming that the 35-dimensional feature representation provides sufficient discriminative power regardless of classifier. RF is selected for: (a) lowest FPR (0.0158, consistent with main evaluation), (b) native Gini feature importance for SHAP TreeExplainer, and (c) superior latency in deployment scenarios. The near-identical performance validates that PCAP-AIDE's results are not an artefact of RF-specific behaviour.

Table 13. Algorithm ablation on UNSW-NB15 (four-part, 100K stratified). All classifiers achieve F1 within 0.001; RF selected for lowest FPR and SHAP compatibility.

Method	Precision	Recall	F1-Score	FPR	ROC-AUC
RF (Random Forest) ★	0.9840	0.9982	0.9910	0.0158	0.9993
XGBoost	0.9836	0.9980	0.9907	0.0162	0.9993
SVM (RBF kernel)	0.9842	0.9980	0.9910	0.0156	0.9989

5.5.2.Five-Fold Cross-Validation

RF CV-F1 = 0.9918 is within 0.0001 of the 80/20 split (F1 = 0.9919), confirming that the reported performance is not specific to a fortunate split. Both classifiers achieve CV ROC-AUC > 0.999, indicating robust probabilistic discrimination across all five folds.

Table 14. Five-fold cross-validation on UNSW-NB15 (four-part). RF CV-F1 = 0.9918 is consistent with the 80/20 split result (F1 = 0.9919), confirming split-independent stability.

Method	CV Recall (mean)	CV F1 (mean)	CV FPR (mean)	ROC-AUC (mean)
RF 5-Fold ★	0.9979	0.9918 ± 0.0007	0.0144	0.9993
XGB 5-Fold	0.9985	0.9917 ± 0.0006	0.0152	0.9994

5.6. Reproducibility and Validation Summary

To ensure reproducibility and rule out data leakage or overfitting, extensive additional validation was conducted: (V1) multi-seed analysis (10 seeds) — CIC F1 *std* = 0.0001, UNSW F1 *std* = 0.0005; (V2) normalization leakage audit — no global scaler used; log1p applied per-sample; CLEAN verdict; (V3) benign class precision — UNSW Benign Precision = 0.9979 ± 0.0004, confirming low false alarm rate; (V4) feature importance stability — total_bytes is rank-1 in 100% of 10 UNSW seeds; (V5) statistical significance — one-sample t-test vs. AdvIDS-2025 baseline: $p < 10^{-30}$ for both datasets (t-statistics: CIC = 1923.2, UNSW = 314.2); (V6) cross-dataset generalization — zero-shot CIC→UNSW: F1 = 0.0012 (expected; feature distribution shift); hybrid CIC+UNSW→UNSW: F1 = 0.681. The complete validation report, including per-seed tables, is provided in the Supplementary Material. *(Supplementary Material is submitted as a separate document accompanying this manuscript.)*

6. LIMITATIONS AND DISCUSSION

6.1. Result Interpretation

Two distinct performance regimes emerge. On CIC-IDS2018 Bot traffic, near-perfect detection (F1 = 0.9997) across five evaluation protocols reflects the inherent discriminability of C&C beaconing: periodic inter-arrival times and fixed packet sizes create a clean decision boundary in statistical feature space [7, 12]. The cross-attack evaluation (Bot → Web, F1 = 0.0012) confirms the system's scope boundary and rules out data-leakage artefacts. On UNSW-NB15 (nine heterogeneous attack categories), F1 = 0.9919 and FPR = 0.0144 demonstrate flow-level generalization to diverse attack types. The FPR increase relative to CIC (0.0144 vs. 0.0001) reflects broader attack diversity rather than a methodological flaw: UNSW Reconnaissance and Analysis flows exhibit statistically similar characteristics to normal traffic. Algorithm ablation (RF/XGB/SVM within 0.001 F1) and five-fold CV confirm statistical robustness. We recommend UNSW-NB15 as the primary benchmark for future NIDS comparisons due to its attack-type diversity.

The hybrid rule-based layer (v2.1) addresses threat classes outside flow-level ML scope: domain blacklisting, DNS gateway spoofing, and ARP spoofing. This complementary design demonstrates the benefit of combining statistical and deterministic detection paradigms.

6.2. System Limitations

Five limitations are inherent to the current design. (1) Application-layer blindness: flow-level statistical features cannot reliably detect SQL injection, XSS, and credential stuffing, which exhibit flow statistics similar to normal HTTP traffic. This represents a fundamental trade-off between privacy preservation and payload inspection. (2) Cross-attack transfer: domain-specific

specialization (confirmed by Protocol V) limits generalization across fundamentally different attack families; comprehensive coverage requires ensemble models or richer feature sets. (3) Offline processing constraint: the current Scapy-based implementation supports offline PCAP analysis and batch CSV processing; real-time line-rate deployment requires integration with DPDK or PF_RING. (4) Rule maintenance overhead: the IOC keyword database requires periodic human oversight despite the ThreatReport-KW automation pipeline, as evolving attacker tactics introduce new IOC patterns. (5) Benchmark-to-real-world gap: all evaluations use public benchmark datasets generated in controlled lab environments; real-world deployment must address traffic diversity, concept drift, and label noise absent from benchmarks.

6.3. What CIC/UNSW Results Validate—and What They Do Not

Validated by the CIC-IDS2018 and UNSW-NB15 benchmarks: (a) exporter-free 35-D feature extraction reproducibility on public CSV/PCAP; (b) class-imbalance mitigation via stratified reservoir sampling + balanced RF; (c) cross-protocol robustness under five evaluation protocols with 10-seed stability; (d) FP reduction via SHAP-guided post-processing (22.9% reduction on UNSW-NB15 500-flow analysis); (e) statistical significance of improvement over AdvIDS-2025 baseline ($p < 10^{-30}$ on both datasets).

Not validated for production deployment by these benchmarks: (a) concept drift under encrypted-traffic dominance (TLS 1.3, QUIC) where payload-independent features may drift; (b) label noise in operational SOC feeds not present in curated benchmarks; (c) real-time line-rate processing at >10 Gbps (see Section 3.8); (d) adversarial evasion against flow-level statistical features. Partial mitigation: the temporal-split (Protocol IV) and cross-attack (Protocol V) protocols reduce data-leakage and overfitting risk, but do not substitute for longitudinal field evaluation. Future work targeting these gaps is outlined in Section 7.

7. Conclusions And Future Work

This paper presented PCAP-AIDE, a lightweight open-source hybrid NIDS that simultaneously addresses four operational gaps persistent in prior work: exporter dependency (C1), class-imbalance fragility (C2), false-positive opacity (C3), and single-protocol evaluation (C4). Evaluated on CIC-IDS2018 and the complete four-part UNSW-NB15 dataset (2.54 M flows; nine attack categories), PCAP-AIDE achieves F1 = 0.9997 and F1 = 0.9919, respectively, surpassing AdvIDS-2025 by ΔF1 = +0.073 and +0.048. SHAP-guided FlowFPFilter v3.0 reduces false positives by 22.9%. Ten-seed variance analysis (F1 std $\leq$ 0.0005) and five-fold cross-validation (CV-F1 = 0.9918 $\pm$ 0.0007) confirm result reproducibility and statistical stability. Algorithm ablation (RF/XGBoost/SVM within 0.001 F1) validates that performance is feature-driven rather than classifier-specific.

Future work includes the development of a packet-to-artifact pipeline for email-borne threats and a dedicated static malware detection engine. We plan to extract EML messages and embedded .ps1, .lnk, .chm, and .vbs files — which are heavily utilized in North Korean APT campaigns — from SMTP captures and forward them to mal_analyzer, our dedicated static malware analysis engine. A follow-up study will detail the LNK, CHM, and VBS analysis modules and evaluate detection performance using 1,000 samples per format (600 malicious, 400 benign), benchmarked against state-of-the-art PowerShell detectors.

Conflicts of Interest

The authors declare no conflict of interest.

ACKNOWLEDGMENT

The author gratefully acknowledges Professor Ieck-chae Euom, the doctoral advisor, for his invaluable guidance and for his revisions and supplementary contributions to this manuscript.

REFERENCES

[1] Mondragon, J. C., Branco, P., Jourdan, G.-V., Gutierrez-Rodriguez, A. E., & Biswal, R. R. (2025) "Advanced IDS: a comparative study of datasets and ML algorithms for network flow-based intrusion detection," Applied Intelligence, Vol. 55, No. 7, Article 608. https://doi.org/10.1007/s10489-025-06422-4

[2] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013) "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, Vol. 36, No. 1, pp. 16-24.

[3] Buczak, A. L., & Guven, E. (2016) "A survey of data mining and ML methods for cyber security intrusion detection," IEEE Comm. Surveys & Tutorials, Vol. 18, No. 2, pp. 1153-1176.

[4] Engelen, G., Rimmer, V., & Joosen, W. (2021) "Troubleshooting an intrusion detection dataset: the CICIDS2017 case study," IEEE Security & Privacy Workshops 2021.

[5] Johnson, J. M., & Khoshgoftaar, T. M. (2019) "Survey on deep learning with class imbalance," Journal of Big Data, Vol. 6, No. 1, pp. 1-54.

[6] Lundberg, S. M., & Lee, S. I. (2017) "A unified approach to interpreting model predictions," NeurIPS, Vol. 30, pp. 4765-4774.

[7] Sommer, R., & Paxson, V. (2010) "Outside the closed world: On using ML for network intrusion detection," IEEE S&P 2010.

[8] Roshan, K., & Zafar, A. (2021) "Utilizing XAI technique to improve autoencoder based model for computer network anomaly detection with Shapley additive explanation (SHAP)," International Journal of Computer Networks & Communications (IJCNC), Vol. 13, No. 6, pp. 109-128.

[9] Roesch, M. (1999) "Snort -- lightweight intrusion detection for networks," USENIX LISA 1999, pp. 229-238.

[10] Garcia-Teodoro, P. et al. (2009) "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers & Security, Vol. 28, pp. 18-28.

[11] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018) "Toward generating a new intrusion detection dataset," ICISSP, pp. 108-116.

[12] Garcia, S., Grill, M., Stiborek, J., & Zunino, A. (2014) "An empirical comparison of botnet detection methods," Computers & Security, Vol. 45, pp. 100-123.

[13] Mirsky, Y. et al. (2018) "Kitsune: An ensemble of autoencoders for online network intrusion detection," NDSS 2018.

[14] Liu, H., & Lang, B. (2019) "Machine learning and deep learning methods for intrusion detection systems: A survey," Applied Sciences, Vol. 9, No. 20, p. 4396.

[15] Lo, W. W. et al. (2022) "E-GraphSAGE: A graph neural network based IDS for IoT," IEEE/IFIP NOMS 2022.

[16] Ferrag, M. A. et al. (2024) "Revolutionizing cyber threat detection with large language models," IEEE Access, Vol. 12, pp. 23733-23750.

[17] Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019) "Improving classification effectiveness of intrusion detection using improved CVAE and deep neural network," Sensors, Vol. 19, No. 11, Article 2528.

[18] Catillo, M., Del Vecchio, A., Pecchia, A., & Villano, U. (2022) "Transferability of machine learning models learned from public intrusion detection datasets: the CICIDS2017 case study," Software Quality Journal, Vol. 30, No. 4, pp. 955-981.

[19] Eunaicy, J. I. C., Jayapratha, C., & Hemachitra, H. S. (2024) "IoT Guardian: A novel feature discovery and cooperative game theory empowered feature selection with ML model for IoT threats & attack detection," International Journal of Computer Networks & Communications (IJCNC), Vol. 16, No. 2, pp. 25-42.

[20] Songma, S., Netharn, W., & Lorpunmanee, S. (2024) “Extending Network Intrusion Detection with Enhanced Particle Swarm Optimization Techniques,” International Journal of Computer Networks & Communications (IJCNC), Vol. 16, No. 4, pp. 61-85.

[21] Abdulqadder, I. H., & Aziz, I. T. (2025) “Load Balanced Attack Defense System with Lightweight Authentication and Modified Blockchain in SDN for B5G,” International Journal of Computer Networks & Communications (IJCNC), Vol. 17, No. 1, pp. 81-99. DOI: 10.5121/ijcnc.2025.17106.
[22] Sharafaldin, I. et al. (2019) "Developing realistic DDoS attack dataset and taxonomy," IEEE International Carnahan Conference on Security Technology (ICCST) 2019.
[23] Moustafa, N., & Slay, J. (2015) "UNSW-NB15: A comprehensive dataset for network intrusion detection systems," MilCIS 2015, pp. 1-6.
[24] Ring, M. et al. (2019) "A survey of network-based intrusion detection data sets," Computers & Security, Vol. 86, pp. 147-167.
[25] Breiman, L. (2001) "Random forests," Machine Learning, Vol. 45, No. 1, pp. 5-32.
[26] Biau, G., & Scornet, E. (2016) "A random forest guided tour," TEST, Vol. 25, No. 2, pp. 197-227.
[27] Vitter, J.S. (1985) "Random sampling with a reservoir," ACM TOMS, Vol. 11, No. 1, pp. 37-57.