An Auditable Hybrid Cascade for Network Threat Detection

Frowin Rabanus Kifaru[1]*, Daniel Mwakiposa[2], Shadrack Stephen Madila[3]
Department of Information and Communication Technology, Moshi Cooperative University, P.O. Box 474, Moshi, Tanzania
frowin2005@gmail.com

Corresponding author: frowin2005@gmail.com

Riwayat Artikel:
Naskah diterima: 25/08/2026 | Naskah direvisi: 29/09/2026 | Naskah diterbitkan: 31/10/2026

Creative Commons License 4.0 (CC BY –NC)

Abstract*—Machine-learning intrusion detectors can provide limited decision transparency in shared-server environments, where analysts must distinguish routine service activity from suspicious flows. This study proposes DIRAF (Direction-Aware Rule-Augmented Forest), a two-stage cascade comprising a conservative, interpretable rule gate and a Random Forest (RF) classifier for unresolved flows. The framework is supported by Quadruple-View Flow Engineering (QVFE), which organizes behavioral predictors into temporal, flow-morphology, port-semantics, and TCP-flag views, and by the Triangulated Attribution Protocol (TAP), which reconciles inferential, permutation-based, and SHAP feature rankings. Evaluation used the CIDDS-001 External Server Week 1 benchmark (172.838 flows); the primary binary task included 156.950 flows labeled normal or suspicious, while 15.888 unknown flows were excluded. On the held-out binary test set, DIRAF achieved 100% accuracy, precision, recall, F1-score, and AUC under the evaluated configuration. Its rule gate resolved 42,3% of test traffic, covering 61,9% of suspiciously labeled flows with 100% precision; the remaining flows were classified by the RF. The cascade reduced mean batch decision latency from 8,01 to 4,77 µs per record. These results are specific to this benchmark: direction laundering reduced Model B accuracy to 68,8%, and the direction-label association limits external validity. DIRAF therefore demonstrates an auditable cascade design on CIDDS-001 Week 1, not universal or production-level detection performance.

***Keywords*—intrusion detection, network flows, hybrid cascade, Random Forest, feature attribution, auditability**

Intisari*—Detektor intrusi berbasis pembelajaran mesin dapat memberikan transparansi keputusan yang terbatas di lingkungan server bersama, di mana analis perlu membedakan aktivitas layanan rutin dari aliran yang mencurigakan. Penelitian ini mengusulkan DIRAF (Direction-Aware Rule-Augmented Forest), sebuah cascade dua tahap yang menggabungkan gerbang aturan konservatif yang dapat diinterpretasikan dengan pengklasifikasi Random Forest (RF) untuk aliran yang belum ditentukan. Kerangka ini didukung oleh Quadruple-View Flow Engineering (QVFE), yang mengelompokkan prediktor perilaku ke dalam pandangan temporal, morfologi aliran, semantik port, dan flag TCP, serta Triangulated Attribution Protocol (TAP), yang merekonsiliasi pemeringkatan atribusi inferensial, berbasis permutasi, dan SHAP. Evaluasi menggunakan benchmark CIDDS-001 External Server Week 1 yang berisi 172,838 aliran. Tugas biner utama mencakup 156,950 aliran berlabel normal atau mencurigakan, sedangkan 15,888 aliran berlabel tidak diketahui dikecualikan. Pada data uji tertahan, DIRAF mencapai akurasi, presisi, recall, F1-score, dan AUC sebesar 100% dalam konfigurasi yang dievaluasi. Gerbang aturan memutuskan 42,3% lalu lintas uji dan mencakup 61,9% aliran berlabel mencurigakan dengan presisi 100%; aliran sisanya diklasifikasikan oleh RF. Cascade menurunkan latensi keputusan batch rata-rata dari 8,01 menjadi 4,77 µs per record. Hasil ini terbatas pada benchmark tersebut: penyamaran arah menurunkan akurasi Model B menjadi 68,8%, dan keterkaitan label dengan arah membatasi validitas eksternal. Dengan demikian, DIRAF menunjukkan rancangan cascade yang dapat diaudit pada CIDDS-001 Week 1, bukan kinerja deteksi universal atau kinerja produksi.

Kata Kunci— deteksi intrusi, aliran jaringan, cascade hibrida, Random Forest, atribusi fitur, auditabilitas

I. INTRODUCTION

Shared-server environments host multiple tenants and services on a common attack surface, complicating the distinction between authorized activity and reconnaissance or intrusion attempts [1], [2]. Flow-based machine-learning (ML) detectors can support this task, and engineered traditional ML pipelines remain competitive with more complex models on structured security data [3], [4]. However, high predictive scores alone do not show which decisions can be audited, whether benchmark labels encode environmental shortcuts, or how a detector behaves when contextual features are unavailable.

Prior work has explored feature selection, ensemble classifiers, hybrid learning, and deep architectures for intrusion detection [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Yet many evaluations emphasize aggregate accuracy, while operational use also requires a decision path that analysts can inspect, measured compute cost, and explicit treatment of dataset-specific cues. In particular, a flow benchmark may contain structural associations between endpoint roles and operator-assigned labels. A detector can

exploit those associations without learning traffic behaviors that transfer to a different network.

This study addresses that gap by evaluating a two-stage cascade on the CIDDS-001 External Server Week 1 capture. DIRAF (Direction-Aware Rule-Augmented Forest) applies a conservative, threat-side rule gate and delegates unmatched records to a Random Forest (RF). QVFE (Quadruple-View Flow Engineering) groups predictors into four behavior-oriented views, while TAP (Triangulated Attribution Protocol) compares inferential, permutation-based, and additive attribution rankings. The study contrasts a behavior-only feature set (Model A) with a feature set that adds protocol and endpoint-direction context (Model B), and examines ablations, feature corruption, attribution stability, and batch latency.

The contributions are: (1) a reproducible description and audit of the DIRAF cascade, including its gate coverage and residual classification; (2) a four-view feature organization and a triangulated attribution procedure; (3) a controlled comparison of behavioral and contextual representations, including robustness and feature-laundering analyses; and (4) benchmark-scoped operational measurements and a reference deployment design. The study does not claim that the benchmark results generalize to production traffic. Section II reviews related work; Section III presents DIRAF, QVFE, and TAP; Section IV describes the data and experimental design; Section V reports the results; Section VI discusses interpretation, deployment implications, and limitations; and Section VII concludes.

II. RELATED WORK

A. Machine-Learning Intrusion Detection

Network intrusion-detection research includes genetic-algorithm feature selection, ensemble classification, Random Forest-based systems, and studies comparing ML and deep-learning models [5], [6], [7], [8], [9]. Hybrid approaches combine machine and deep learning or use ensemble methods to improve detection under class imbalance [10], [11]. Attention-based and software-defined-network methods offer alternative architectures, although they may increase model complexity [12], [13]. Broader reviews emphasize the continuing role of artificial intelligence in cybersecurity [14]. More recent work considers sequence-aware anomaly detection and federated learning for log and event analysis [15], [16], [17]. These studies motivate strong tabular baselines but do not by themselves address an auditable rule-to-model cascade for flow classification.

B. Explainability, Attribution, and Operations

Feature-level explainability is commonly studied using SHAP and LIME, including applications to intrusion and botnet detection [18]–[21]. Such methods characterize different aspects of model behavior: additive attribution, local surrogate explanations, or performance changes under feature perturbation. Security operations research also examines alert prioritization and analyst workflows [22], [23]. Work on concept drift and adversarial perturbations highlights the need to monitor changing data and model reliability [24], [25]. DIRAF and TAP build on these lines by making part of the classification path rule-auditable and by comparing distinct attribution channels rather than treating one ranking as definitive.

C. Open-Set Detection, Robustness, and Statistical Inference

Open-set recognition and unknown-attack detection address cases not represented in the training labels [26], [27]. Adversarial-security surveys similarly caution that learned NIDS can be affected by evasion and adaptive behavior [28], [29]. For inferential analysis under complete or quasi-complete separation, Firth-penalized logistic regression provides finite estimates where ordinary maximum-likelihood estimates may diverge [30], [31], [32]. These methods inform the present study's statistical analysis and motivate future evaluations on independent datasets, including CICIoT2023 [33]. The CIDDS-001 flow dataset and its construction are described by Ring et al. [34].

D. Research Gap

The reviewed literature motivates a combined assessment of prediction, auditability, attribution, and operational behavior in one flow-based study. The present work evaluates these aspects under a shared split and explicitly distinguishes behavior-derived predictors from endpoint/protocol context. The evaluation is benchmark-specific and does not constitute an external validation across network datasets.

Table I summarizes representative approaches and the combined audit, attribution, and deployment focus of this study.

TABLE I
POSITIONING OF THIS STUDY AGAINST REPRESENTATIVE WORK

Prior work	Core approach	Explainability	Hybrid/audit	Deployment pathway
Ahmed et al., 2024 [5]	Genetic-algorithm-optimized decision forest	Feature selection	No	No
Goyal et al., 2025 [18]	Deep-learning IDS	SHAP and LIME	No	No
Jalalvand et al., 2024 [22]	SOC alert-prioritization survey	SHAP noted	No	Analyst workflow
Shyaa et al., 2024 [24]	Drift-aware IDS survey	No	No	Drift adaptation
This study	QVFE + DIRAF rule-gate/RF cascade	TAP: inferential, sensitivity-based, additive consensus	42,3% of test decisions directly rule-auditable	Thresholding, latency, drift monitoring, retraining, rollback

III. PROPOSED METHOD

A. DIRAF Architecture and Decision Flow

DIRAF is a two-stage classifier. Given a flow record x, the QVFE transformation produces a feature vector. Stage 1 evaluates an ordered list of four threat-side rules. The first matching rule assigns the flow to the operator-labelled suspicious class and records the rule responsible. A record that matches no gate rule is passed to a 200-tree RF trained on the Model B feature representation. The residual RF applies a probability threshold $\tau = 0{,}05$ for alerting. The rule gate is asymmetric: it does not assign normal labels; all gate

JUTEI Edisi Volume. 10 No.2 Oktober 2026
ISSN 2579-3675, e-ISSN-5538
DOI 10.21460/jutei.2026.102.533

misses, including normal flows, are classified by the RF. Rules are evaluated in a fixed order, so overlaps are counted once.

Gate candidates were searched exhaustively as single-candidate rules on the training split. A candidate was retained only when its training precision was 100%, reducing the risk of gate-issued false positives on the training data. This induction constraint does not guarantee zero false positives under distribution shift; test-set precision is reported separately. The cascade decision can be written as:

$$\hat{y}(x) = \text{suspicious, if the first applicable gate rule } g_i(x) = 1; \text{ otherwise } \hat{y}(x) = 1[pRF(x) \geq \tau], \text{ with } \tau = 0{,}05. \quad (1)$$

Here, pRF(x) is the RF-estimated probability of the suspicious class, and the binary label 1 denotes suspicious. If no gate rule applies, the RF produces the final classification. The gate therefore provides a directly inspectable decision and reduces the share of records requiring learned inference, while the RF handles the residual cases.

B. Quadruple-View Flow Engineering

QVFE organizes the 17 core behavioral predictors into four views defined by detection hypotheses before model fitting. This grouping supports feature auditing and view ablation; it is not a claim that the grouped variables are independent.

V1—Temporal view: hour of day and a working-hours indicator for 07:00–18:00. V2—Flow-morphology view: $\log_{10}$ duration, $\log_{10}$ packets, $\log_{10}$ bytes, bytes per packet, packet rate, and byte rate. V3—Port-semantics view: an indicator for destination ports below 1024, source port (SrcPt), and destination port (DstPt). V4—TCP-flag-semantics view: flag count, RST, FIN, PSH, and ACK indicators, and a SYN-only indicator. These features describe timing, flow size and rates, service targeting, and connection/flag behavior.

Model B adds four context features: Proto_UDP, Proto_ICMP, isServerSrc, and isServerDst. The endpoint variables indicate whether a flow originates from or terminates at the monitored EXT_SERVER and OPENSTACK_NET endpoints. They depend on deployment-specific role definitions and therefore are not fully environment-independent. Model A uses the 17 behavioral features; Model B uses all 21 features. A reduced 14-predictor specification is used only for Firth regression and TAP rankings: $\log_{10}$PktRate and $\log_{10}$ByteRate are omitted because of high collinearity (r = 0,95), and DstPt is omitted because the well-known destination-port indicator captures the relevant service distinction. The descriptive variable isTCP is not included in either classifier configuration.

C. Triangulated Attribution Protocol

TAP reconciles three attribution channels: (1) absolute Firth-penalized regression coefficients (|β|) from the reduced feature specification, (2) permutation importance measured by F1 decrease, and (3) TreeExplainer SHAP rankings. Features are tiered by cross-channel agreement. Tier 1 represents the operational predictive core; Tier 2 contains signature-like variables with stronger inferential than predictive ranks; and Tier 3 contains weaker supporting features. The three channels answer different questions and should not be treated as interchangeable measures of causal effect.

Permutation importance is calculated on a 10.000-record subset for TAP; SHAP is computed on a 4.000-record subset. A separate attribution-stability comparison uses 10 bootstrap repetitions and reports mean pairwise Spearman rank correlation and runtime. LIME is included only as an external stability baseline, not as a TAP channel.

D. Model Variants and Component Interaction

The principal comparison evaluates Models A and B with five classifiers: Logistic Regression (LR) with balanced class weights, RF with 200 trees, a decision tree with maximum depth 6, an RBF-kernel support vector machine (SVM), and Gradient Boosting with 200 estimators. DIRAF uses the Model B RF as the residual classifier and the four-rule threat-side gate. QVFE ablations remove one view at a time or remove context features. This design separates the contribution of feature representation from that of the cascade decision path.

IV. RESEARCH METHODOLOGY AND EXPERIMENTAL SETUP

A. Dataset and Label Construction

The study uses CIDDS-001 (Coburg Intrusion Detection Data Sets), External Server Week 1. The dataset was generated in a virtualized environment containing an external public server and an internal OpenStack network with simulated employee hosts [34]. The capture contains 172.838 flow records. Available fields include flow start time, duration, protocol (TCP/UDP/ICMP/GRE), anonymized endpoints and ports, packet and byte counts, and TCP flag strings. Constant Flows fields and the all-zero TOS field were omitted. Week 1 attackType, attackID, and attackDescription fields were not used because they do not provide attack annotations for this capture.

The primary binary task distinguishes operator-labeled suspicious flows (n = 107.344) from normal flows (n = 49.606), for N = 156.950. The suspicious-to-normal ratio is approximately 2,16:1. The remaining 15.888 unknown flows are excluded from the binary task because their direction could not be uniquely determined in the simulation; they are retained for descriptive analysis and for the separate three-class evaluation. Throughout the paper, “threat” is shorthand for the operator-assigned suspicious label and does not imply independently verified malicious activity.

B. Preprocessing, Splits, and Model Fitting

The binary task uses a stratified 70:30 train–test split, yielding 109.865 training records and 47.085 test records. Five-fold stratified cross-validation is used for model development and for the comparison in Table V. An independent ten-fold stratified cross-validation procedure is used for the fold-level statistical comparison in Section V. For the computationally expensive RBF SVM, training uses a stratified subsample of 20.000 records. A separate stratified 70:30 split of the full three-class dataset is used for the robustness evaluation, with 51.852 test records.

Continuous predictors in logistic regression are standardized. Inferential analyses contrast maximum-likelihood estimation (MLE) with Firth-penalized logistic regression, implemented via iteratively reweighted least squares with a Jeffreys-prior score adjustment [30]–[32]. Distributional analyses use chi-square goodness-of-fit and independence tests; a continuity correction is applied to 2 × 2 tables with sparse or empty cells. Effect sizes are reported as Cramér's V and odds ratios. Continuous predictor associations are assessed using point-biserial Pearson correlations with Fisher-z confidence intervals; multiple comparisons are controlled using the Benjamini–Hochberg false-discovery-rate procedure.

C. Evaluation Metrics and Statistical Tests

Classification performance is reported using accuracy, precision, recall, F1-score, area under the receiver operating characteristic curve (AUC), and confusion-matrix error counts. Accuracy confidence intervals use the Wilson method. Pairwise classifier comparisons use McNemar's test with continuity correction. The ROC analysis includes Youden's J threshold and high-sensitivity operating points. Feature attribution uses permutation importance and SHAP as described in Section III.C. Robustness experiments add Gaussian noise, introduce missing cells followed by median imputation, withhold behavior families during training, and launder endpoint direction. No independent external dataset was used.

D. Experimental Configuration and Available Environment Details

The two classifier configurations and their predictor counts are summarized in Table II.

TABLE II
PREDICTOR VARIABLES USED IN THE TWO CONFIGURATIONS

Configuration	Predictors	Purpose
Model A (17)	$\log_{10}$Duration, Hour, WorkingHours, SrcPt, DstPt, isWellKnownDstPort, $\log_{10}$Packets, $\log_{10}$Bytes, BytesPerPacket, $\log_{10}$PktRate, $\log_{10}$ByteRate, FlagCount, SYN_only, F_RST, F_FIN, F_PSH, F_ACK	Behavioral baseline without endpoint context
Model B (21)	Model A + Proto_UDP, Proto_ICMP, isServerSrc, isServerDst	Incremental protocol and endpoint-direction context

The deployment-capacity measurements used scikit-learn 1.6 and Python 3.13 on a single two-core virtual machine (Xeon 2,60 GHz, 2,1 GB RAM). The source reports model size and CPU use during inference, but does not specify every software package version, random seed, or all preprocessing implementation details. [AUTHOR INPUT REQUIRED: provide code/repository or an implementation appendix, random seeds, and complete software/package versions for full computational reproducibility.]

V. RESULTS

A. Descriptive and Association Analysis

The capture is highly structured: TCP dominates, while UDP, ICMP, and GRE account for smaller shares ($\chi^2(3)$ = 486,775, $p < 0{,}001$). The labels are imbalanced, with suspicious flows the largest class, followed by normal and unknown flows ($\chi^2(2)$ = 74,259, $p < 0{,}001$). Approximately two-fifths of flows occur during working hours ($\chi^2(1)$ = 2,789, $p < 0{,}001$), but suspicious traffic is distributed across the day. AP.SF is the dominant TCP flag pattern, followed by SYN-only probes and other aborted or reset exchanges. SSH- and Telnet-related destination-port activity is strongly associated with the suspicious label, consistent with the dataset's operator labeling policy.

Cross-tabulations show that non-TCP traffic is almost entirely labelled suspicious or unknown, whereas TCP includes all three classes ($\chi^2(6)$ = 2.516,1, $p < 0{,}001$, Cramér's V = 0,085). Normal flows are confined to monitored infrastructure; external-host flows are labelled suspicious or unknown ($\chi^2(2)$ = 38.384,6, $p < 0{,}001$, V = 0,471; binary-task odds ratio = ∞). Within the binary task, no SYN-only flow is normal ($\chi^2(1)$ = 5.290,0, continuity-corrected, $p < 0{,}001$, V = 0,184). Working-hours status is not meaningfully associated with the binary threat label ($\chi^2(1)$ = 0,8, p = 0,363, V = 0,002, OR = 1,01).

Table III reports the predictor–label correlations and adjusted significance values.

TABLE III
CORRELATIONS WITH THE THREAT LABEL, RANKED BY |R| (N = 156.950)

Variable	r	95% CI	p (FDR)	Interpretation
isServerDst	−0,489	[−0,493; −0,485]	<0,001	Threat-labelled flows terminate at external hosts; normal flows at servers
isWellKnownDstPort	+0,463	[+0,459; +0,466]	<0,001	Threat-labelled flows favor well-known service ports
logDuration	+0,373	[+0,369; +0,378]	<0,001	Threat-labelled flows persist longer than normal replies
F_RST	+0,338	[+0,333; +0,342]	<0,001	Reset-heavy exchanges are associated with probing or aborted sessions
SYN_only	+0,184	[+0,179; +0,188]	<0,001	No SYN-only flow is labelled normal
isTCP	−0,110	[−0,115; −0,106]	<0,001	Non-TCP flows are predominantly anomalous/unknown
logBytes	+0,066	[+0,061; +0,071]	<0,001	Threat-labelled exchanges are modestly larger
logPackets	+0,053	[+0,048; +0,058]	<0,001	Threat-labelled exchanges involve modestly more packets
Hour	−0,013	[−0,018; −0,008]	<0,001	Small time-of-day association in a large sample
WorkingHours	−0,002	[−0,007; +0,003]	0,360	Not associated with threat occurrence

Note. isTCP is descriptive only and is not used in either model. $\log_{10}$Packets and $\log_{10}$Bytes are collinear (r = 0,95); isServerDst and isWellKnownDstPort covary (r = 0,43).

Normal flows were short (median 0,062 s, 7 packets, 586 bytes) and server-bound; suspicious flows persisted longer (9,5 s, 14 packets, 2.185 bytes), while unknown flows were long-lived probes, 46% of which were SYN-only. Empty cross-tabulation cells produce extreme odds ratios, which are interpreted descriptively rather than as stable estimates.

JUTEI Edisi Volume. 10 No.2 Oktober 2026
ISSN 2579-3675, e-ISSN-5538
DOI 10.21460/jutei.2026.102.533

B. Firth Logistic Regression

Ordinary MLE did not converge under either feature configuration. After 50 Newton iterations, the SYN-only coefficient in Model A and the isServerDst coefficient in Model B diverged, with sharply increasing standard errors, consistent with complete separation. MLE estimates are therefore documented as a failure mode and are not interpreted. Firth penalization yielded finite estimates. For reduced Model A, Nagelkerke R^2 was 0,963, AUC was 0,9972, and accuracy was 99,29% (likelihood-ratio $\chi^2(14) = 182.011$). Model B produced near-perfect in-sample separation (Nagelkerke $R^2 = 1,000$, AUC = 1,0000, accuracy = 100,00%). These in-sample values reflect the strong association between endpoint context and operator-assigned labels.

The reduced-specification Firth estimates for Models A and B are reported in Table IV.

TABLE IV
FIRTH-PENALIZED LOGISTIC REGRESSION RESULTS FOR MODEL A (REDUCED 14-PREDICTOR SPECIFICATION) AND MODEL B

Model	Predictor	Firth B	SE	p	OR (95% CI)
A	logDuration	+3,98	0,08	<0,001	53,50 (46,00–62,21)
A	Hour	−0,21	0,03	<0,001	0,81 (0,78–0,86)
A	SrcPt	−6,52	0,17	<0,001	0,001 (<0,001–0,01)
A	logPackets	−3,34	0,11	<0,001	0,04 (0,03–0,04)
A	logBytes	+2,88	0,11	<0,001	17,73 (14,20–22,14)
A	BytesPerPacket	−0,12	0,02	<0,001	0,89 (0,86–0,92)
A	FlagCount	−4,47	0,51	<0,001	0,01 (<0,001–0,03)
A	WorkingHours	−0,16	0,05	0,002	0,85 (0,77–0,94)
A	isWellKnownDstPort	+19,53	0,43	<0,001	>10^6 (capped; e^B = $3{,}0\times10^8$)
A	SYN_only	+11,21	1,50	<0,001	73.935,18 (3.929,94–$1{,}39\times10^6$)
A	F_RST	+12,16	0,49	<0,001	$1{,}91\times10^5$ (73.759,76–$4{,}97\times10^5$)
A	F_FIN	−5,51	0,57	<0,001	0,004 (<0,001–0,01)
A	F_PSH	+6,50	0,54	<0,001	665,13 (232,61–1.901,94)
A	F_ACK	−7,40	0,79	<0,001	<0,001
B	logDuration	−1,32	0,46	0,004	0,27 (0,11–0,66)
B	Hour	+0,07	0,33	0,831	1,07 (0,56–2,04)
B	SrcPt	−0,30	0,49	0,540	0,74 (0,28–1,94)
B	logPackets	+1,71	0,81	0,035	5,54 (1,13–27,24)
B	logBytes	+0,28	0,87	0,744	1,33 (0,24–7,28)
B	BytesPerPacket	−0,27	0,06	<0,001	0,77 (0,68–0,86)
B	FlagCount	+4,02	1,96	0,041	55,57 (1,19–2.603,67)
B	WorkingHours	+0,24	0,69	0,725	1,27 (0,33–4,90)
B	isWellKnownDstPort	+0,51	1,01	0,618	1,66 (0,23–12,09)
B	SYN_only	+2,50	1,91	0,190	12,20 (0,29–511,61)
B	F_RST	−2,30	2,04	0,260	0,10 (<0,001–5,48)
B	F_FIN	−2,38	2,21	0,282	0,09 (<0,001–7,06)
B	F_PSH	−6,83	3,15	0,030	0,001 (<0,001–0,52)
B	F_ACK	+0,83	2,10	0,695	2,28 (0,04–141,26)
B	Proto_UDP	+3,89	1,81	0,031	48,90 (1,42–1.687,73)
B	Proto_ICMP	+2,71	1,72	0,115	14,98 (0,52–432,86)
B	isServerSrc	−22,75	1,83	<0,001	<0,001
B	isServerDst	−23,23	1,78	<0,001	<0,001

Note. N = 156.950. MLE diverged (B ≈ 32, SE ≈ 9.636 at iteration 50); Firth penalization yields finite estimates (penalized log-likelihood −6.877,8). Extreme odds ratios reflect empty cells; selected displays are capped at 10^6. Predictor scope is described in Section III.B.

Adding direction context largely suppresses several Model A behavioral coefficients, consistent with strong overlap between endpoint roles and labels in this capture. This is an association in the dataset, not evidence that endpoint direction causes suspicious activity.

C. Comparative Classifier Performance

Table V reports held-out performance for the binary task (47.085 records: 14.882 normal and 32.203 suspicious), Wilson 95% accuracy intervals, five-fold cross-validation accuracy, and training time. Contextual Model B is highly separable: LR, RF, Gradient Boosting, and DIRAF achieved 100,00% across the reported held-out classification metrics. SVM and the decision tree showed small deviations. On behavior-only Model A, RF achieved 99,99% accuracy (5 errors), Gradient Boosting 99,98% (9 errors), and LR 99,32% (152 false positives and 166 false negatives). All reported model training times were below 36 s; RF training took 7,52 s for Model A and 5,58 s for Model B on the reported two-core environment.

TABLE V
COMPARATIVE PERFORMANCE OF MODELS A AND B

Config.	Classifier	Acc.	95% CI	Prec.	Rec.	F1	AUC	CV accuracy (μ±σ)	Train (s)
A	Logistic Regression	0,9932	0,9925–0,9939	0,9953	0,9948	0,9951	0,9979	0,9936±0,0003	0,72
A	Random Forest	0,9999	0,9998–1,0000	0,9999	0,9999	0,9999	1,0000	0,9998±0,0001	7,52
A	Decision Tree	0,9972	0,9966–0,9976	0,9963	0,9996	0,9979	0,9998	0,9968±0,0002	0,30
A	SVM (RBF)	0,9960	0,9954–0,9966	0,9962	0,9980	0,9971	0,9994	0,9955±0,0015	0,26
A	Gradient Boosting	0,9998	0,9996–0,9999	0,9998	0,9999	0,9999	1,0000	0,9998±0,0001	35,14
B	Logistic Regression	1,0000	0,9999–1,0000	1,0000	1,0000	1,0000	1,0000	1,0000±0,0000	0,24
B	Random Forest	1,0000	0,9999–1,0000	1,0000	1,0000	1,0000	1,0000	1,0000±0,0000	5,58
B	Decision Tree	0,9971	0,9966–0,9976	0,9963	0,9995	0,9979	0,9998	0,9968±0,0002	0,30
B	SVM (RBF)	0,9999	0,9998–1,0000	0,9999	1,0000	1,0000	1,0000	0,9997±0,0002	0,19
B	Gradient Boosting	1,0000	0,9999–1,0000	1,0000	1,0000	1,0000	1,0000	1,0000±0,0000	34,08
B	DIRAF (proposed)	1,0000	0,9999–1,0000	1,0000	1,0000	1,0000	1,0000	0,99997±0,00002	5,58†

Note. † DIRAF induction time includes RF; gate extraction took <10 ms. DIRAF AUC combines gate decisions and RF residual probabilities. Model B CV AUC = 1,0000±0,0000 for all; Model A CV AUC: LR 0,9977±0,0002, DT 0,9996±0,0001, SVM 0,9998±0,0004, RF/GB 1,0000.

D. Pairwise Tests and Three-Class Evaluation

On the Model B test set, LR, RF, and Gradient Boosting classified all 47.085 records correctly. The decision tree made 136 errors and the SVM made 3. McNemar tests recomputed from the test predictions show statistically significant differences between the perfect classifiers and the decision tree, while comparisons with the SVM are not significant after continuity correction for three discordant pairs.

Table VI presents the corresponding pairwise McNemar comparisons.

TABLE VI
MCNEMAR TESTS ON HELD-OUT MODEL B PREDICTIONS

Comparison	b	c	p-value
LR vs. Decision Tree	136	0	<0,001
LR vs. SVM (RBF)	3	0	0,248†
RF vs. Decision Tree	136	0	<0,001
RF vs. SVM (RBF)	3	0	0,248†

Comparison	b	c	p-value
Decision Tree vs. SVM (RBF)	3−o	136−o	<0,001 for all o≤3
Decision Tree vs. Gradient Boosting	0	136	<0,001
SVM (RBF) vs. Gradient Boosting	0	3	0,248†

Note. b = first classifier correct/second incorrect; c = converse. Perfect pairs (LR–RF, LR–GB, RF–GB) have b = c = 0 (p = 1,0) and are omitted. † Continuity-corrected $\chi^2 = (3-1)^2/3 = 1,33$; exact binomial p = 0,250. For Decision Tree vs. SVM, o is the number of shared errors (o ≤ 3).

In a separate three-class evaluation using the full 172.838-record dataset and a stratified 70:30 split (51.852 test records), RF achieved 99,988% accuracy and macro F1 = 0,9998. Six test records were misclassified, all between Suspicious and Unknown; the Normal class was classified without error.

E. DIRAF Gate Audit and Cascade Performance

The gate consists of four ordered, threat-side rules: SYN-only probe, non-TCP protocol, Telnet destination (port 23), and destination outside monitored infrastructure. Rules were selected for 100% training precision. Table VII reports training and held-out test fires and precision. On the held-out test set, the gate resolved 19.922 records (42,3% of 47.085), including 61,9% of suspiciously labelled flows, with zero false positives. The residual RF classified the remaining stream without error in this test, so the cascade matched the monolithic RF metrics on this benchmark.

TABLE VII
DIRAF STAGE 1 GATE AUDIT (ORDERED RULES; FIRST MATCH WINS)

Gate rule	Train fires	Train precision	Test fires	Test precision
SYN-only probe	7.360	100,0%	3.312	100,0%
Non-TCP protocol	2.803	100,0%	1.233	100,0%
Telnet destination (port 23)	31	100,0%	10	100,0%
Destination outside monitored infrastructure	36.250	100,0%	15.367	100,0%
All gate rules combined	46.444	100,0%	19.922	100,0%

Note. Normal-side induction found no perfectly pure candidate; the best server-bound port signature had 99,2% purity. The gate is therefore intentionally asymmetric, with benign triage delegated to the RF.

F. QVFE and DIRAF Ablations

The QVFE view ablations use RF with identical held-out splits (N = 47.085). Removing flow morphology (V2) caused the largest Model A decline: accuracy decreased from 99,99% to 99,81%. Removing other views caused smaller changes. Raw exporter fields and automated feature selection also performed near perfectly, indicating that QVFE's principal value in this benchmark is the organization and auditability of features rather than a unique increase in predictive accuracy.

Table VIII compares the QVFE views with raw exporter fields and automated selection.

TABLE VIII
QVFE VIEW ABLATION, RAW FIELDS, AND AUTOMATED FEATURE SELECTION

Configuration	k	Acc.	F1	AUC	FP	FN	Time (s)
QVFE full Model B	21	1,0000	1,0000	1,0000	0	0	5,7
Behavioral only Model A	17	0,9999	0,9999	1,0000	2	3	7,8
Model B − direction view	19	0,9999	0,9999	1,0000	2	3	7,0
Model B − direction and SYN only	18	0,9999	0,9999	1,0000	2	3	7,2
Model A − temporal view V1	15	0,9999	0,9999	1,0000	3	4	6,0
Model A − morphology view V2	11	0,9981	0,9986	0,9991	43	46	3,7
Model A − port view V3	14	0,9997	0,9998	1,0000	9	6	6,5
Model A − flag view V4	11	0,9998	0,9999	1,0000	7	2	8,7
Model A − SYN_only indicator	16	0,9999	0,9999	1,0000	2	3	6,2
Raw NetFlow fields only	10	0,9999	0,9999	1,0000	3	2	5,2
Mutual-information k-best	10	1,0000	1,0000	1,0000	0	0	8,6 + 7,2
RFE (RF, 30k subsample)	10	0,9997	0,9998	1,0000	—	—	4,7 + 7,2

Note. Time is training time; for the last two rows it includes selection plus training. RFE had approximately 15 test errors (split omitted in source). Model A has 17 predictors (V1=2, V2=6, V3=3, V4=6); Model B has 21.

Table IX isolates the cascade components. The gate-only variant is fast but misses suspicious flows not captured by the rules; the RF-only variant classifies accurately but exposes no rule-level decisions. The cascade retains RF-level benchmark performance and makes 42,3% of decisions directly auditable. Batch mean latency decreased from 8,01 to 4,77 µs per record. The source also reports 9,2 ms single-flow online latency (p95 9,5 ms), including Python overhead; this is distinct from batch latency.

The detector-level component ablation is summarized in Table IX.

TABLE IX
ABLATION OF DETECTOR VARIANTS ON THE IDENTICAL TEST SET

Detector variant	Acc.	F1	FP	FN	Auditable	Latency/record	Throughput
Random Forest only	1,0000	1,0000	0	0	0%	8,01 µs	124.768 rec/s
Rule gate only	0,7392	0,7644	0	12.281	100%	0,15 µs	6.819.129 rec/s
DIRAF (gate + RF cascade)	1,0000	1,0000	0	0	42,3%	4,77 µs	209.627 rec/s

Note. Latency and throughput measured on the 47.085-record held-out batch. The rule-only residual comprises 27.163 records.

G. Attribution Stability and TAP Rankings

Table X presents a comparative analysis of the four attribution channels used in the Triangulated Attribution

JUTEI Edisi Volume. 10 No.2 Oktober 2026
ISSN 2579-3675, e-ISSN-5538
DOI 10.21460/jutei.2026.102.533

Protocol (TAP), focusing on attribution stability and runtime performance. Attribution stability reflects the consistency of the results produced by each channel across repeated evaluations, while runtime indicates the computational efficiency of the respective approach. The comparison provides a basis for examining how reliably and efficiently each attribution channel contributes to the overall TAP process. The results also help identify differences among the channels in terms of consistency and computational cost, supporting a more systematic interpretation of the final attribution outcomes.

TABLE X
STABILITY AND COST OF ATTRIBUTION CHANNELS

Attribution channel	Stability (ρ)	Runtime/run	Top 3 (Model A)
Permutation importance	0,400	3,2 s	SrcPt, isWellKnownDstPort, logBytes
SHAP (TreeExplainer)	0,994	6,6 s	$\log_{10}$ByteRate, SrcPt, isWellKnownDstPort
Firth |β| (inferential)	0,327	7,7 s	isWellKnownDstPort, SrcPt, FlagCount
LIME (local surrogates)	0,852	19,4 s	isWellKnownDstPort, SrcPt, logBytes

Note. Stability is mean pairwise Spearman ρ across 10 bootstrap repetitions. LIME is an external baseline and is not part of TAP.

For Model B, the attribution channels consistently ranked endpoint direction as one of the most influential features across the five classifiers. Permutation-based F1 decreases ranged from 0.073 to 0.267 for isServerDst and from 0.072 to 0.266 for isServerSrc. Similarly, the mean SHAP values for Random Forest were 0.081 and 0.079, respectively, while Gradient Boosting produced 10.702 and 8.924. By comparison, the next most influential behavioral feature, $\log_{10}$ByteRate, had a SHAP value of 1.458. Model A provides a more informative assessment of behavior-only predictors, while the predictive channels within TAP exhibited strong rank-order agreement, with Spearman's ρ = 0.87 and Kendall's W = 0.50 (tie-corrected W = 0.52). Table XI reports the feature-level ranks and TAP tiers for the behavior-only model.

TABLE XI
TAP RECONCILIATION ON BEHAVIOR-ONLY FEATURES (REDUCED MODEL A, 14 PREDICTORS)

Feature	Firth |β| rank	Permutation rank	SHAP rank	TAP tier
SrcPt	5	1	3	T1
isWellKnownDstPort	1	2	2	T1
logBytes	11	3	4	T1
logPackets	10	4	5	T1
logDuration	9	5	1	T1
BytesPerPacket	14	6	6	T1
FlagCount	8	7	8	T3
F_FIN	7	11	7	T3
Hour	12	11	13	T3
WorkingHours	13	11	12	T3
SYN_only	3	11	14	T2
F_RST	2	11	10	T2
F_PSH	6	11	9	T2
F_ACK	4	11	11	T2

Note. T1 = predictive operational core (top six by both predictive channels); T2 = signature-style variables with high inferential but weaker predictive rank; T3 = weaker support. Ranks use the reduced 14-predictor scope.

H. Cross-Validation, Robustness, and Threshold Analysis

Ten-fold cross-validation for the behavioral configuration found high performance across classifiers. RF and Gradient Boosting had the highest mean accuracy, followed by decision tree, SVM, and LR. The Friedman test indicated an overall classifier difference ($\chi^2(4) = 37,3$, $p < 0,001$). Nemenyi post hoc comparisons found RF and Gradient Boosting significantly better than SVM and LR, but not significantly different from the decision tree. Holm-corrected one-sided exact Wilcoxon tests found RF significantly better than LR, SVM, and decision tree, while RF and Gradient Boosting did not differ ($p = 1,0$).

Under feature corruption, performance declined as Gaussian noise increased; missing values were median-imputed. With 20% noise, accuracy decreased to 0,9325; with 20% missing cells, it was 0,9821. When SYN-only probes were withheld from training and the corresponding indicator removed, recall on the withheld probes was 1,0000. Withheld non-TCP flows were detected at 0,9992 recall when protocol information was retained. Direction laundering, which made flows appear infrastructure-bound, reduced Model B accuracy to 0,688 and increased the false-alarm rate to 0,988. Removing only the SYN-only indicator did not measurably change performance.

Table XII summarizes the corruption and withheld-behavior evaluations.

TABLE XII
ROBUSTNESS AND UNSEEN-BEHAVIOR EVALUATION FOR BEHAVIORAL RF (MODEL A)

Condition	Setting / n (test)	Accuracy or recall	F1	AUC	FP
Baseline (clean test set)	—	0,9999	0,9999	1,0000	2
Gaussian noise	5% feature SD	0,9972	0,9980	1,0000	127
Gaussian noise	10% feature SD	0,9863	0,9901	0,9999	638
Gaussian noise	20% feature SD	0,9325	0,9529	0,9974	3.139
Missing (median imputed)	5% cells	0,9991	0,9994	1,0000	10
Missing (median imputed)	10% cells	0,9963	0,9973	0,9999	10
Missing (median imputed)	20% cells	0,9821	0,9867	0,9994	28
Held out: SYN-only probes; feature removed	3.312	1,0000 recall	0,9999*	—	—
Held out: non-TCP flows; protocol retained	1.233	0,9992 recall	0,9999*	—	—

*Note. For stress rows, reported scores are for the corrupted test set. For held-out behavior rows, the third column is recall on withheld flows; *F1 is for remaining records. Dashes indicate values not reported in the source.*

Model B ROC curves lie near the upper-left corner, with AUC values of at least 0,9998. Model A RF and Gradient Boosting had AUC = 1,0000, with decision tree and SVM close behind and LR lowest. The preferred threshold depends

on the relative costs of false negatives and false positives. The source recommends sensitivity-first operating points (τ = 0,01–0,05) when missed suspicious flows are more costly. DIRAF applies τ = 0,05 to the RF residual; the rule gate is threshold-free. For SVM, thresholds apply to the decision-function scale.

Batch latency, throughput, model size, and deployment-capacity measurements are reported with their operational interpretation in Section VI.C.

VI. DISCUSSION

A. Interpretation of Predictive Performance and 100% Precision

On the evaluated CIDDS-001 External Server Week 1 split, DIRAF achieved 100% precision and matched the monolithic Model B RF across reported held-out metrics. This finding is bounded to the dataset, feature construction, split, and evaluation procedure. It does not establish perfect intrusion detection in general. The gate's 100% test precision means that no gate-issued suspicious prediction was false in this test set; it does not mean that the gate detected every suspiciously labelled record. The gate covered 61,9% of such flows and resolved 42,3% of total test traffic, with remaining records delegated to the RF.

The unusually high scores are consistent with structural associations in this benchmark. Endpoint direction is strongly related to the operator labels, and direction laundering reduces Model B accuracy to 68,8%. Thus, performance with endpoint context can reflect benchmark-specific cues as well as behavior. Model A provides a useful behavior-only comparison and remains highly predictive in the same split, but this result also requires external replication before it can be interpreted as transferable detection capability.

B. Contribution of the Cascade, QVFE, and TAP

The gate/RF combination illustrates a division between decisions supported by simple, inspectable rules and residual cases requiring a learned classifier. Rule coverage is not equivalent to total detection coverage: it describes the share of all test records decided directly by the gate. Suspicious-flow coverage describes the share of operator-labelled suspicious records handled by those gate rules. The gate-only ablation shows why the residual classifier remains necessary; many suspicious flows do not satisfy any retained rule. The batch-latency reduction is measured under the reported batch setup and should not be equated with end-to-end or single-record production latency.

QVFE organizes variables into four auditable groups, but ablations show that near-perfect benchmark performance is not unique to that grouping: raw fields and automated feature selection performed similarly. Flow morphology had the largest impact among the behavioral views. Accordingly, the evidence supports QVFE primarily as a feature-organization and analysis framework, not as a demonstrated source of across-dataset accuracy gains. TAP offers complementary rankings rather than a single definitive explanation. Variation in attribution stability reinforces the importance of reporting method and sampling procedure alongside feature ranks.

C. Operational Implications

DIRAF can be considered as a design pattern when analysts need a traceable rule path for a subset of traffic while retaining an ML model for unresolved flows. Before any deployment, endpoint-role definitions and rule precision must be re-evaluated on representative local traffic. The 9,2 ms single-flow online latency, despite high batch throughput, illustrates that processing mode matters. Parsing, enrichment, storage, network transport, and SOC workflow overheads were not included in the benchmark batch figures. Proposed controls—population stability index monitoring, shadow retraining, a versioned model registry, analyst feedback, and rollback—are governance recommendations, not experimentally validated production controls. Figure 1 presents the reference architecture: flow producers feed the QVFE feature service; the DIRAF gateway applies the four rules and sends unmatched records to the 200-tree Model B RF; a threshold policy (τ = 0,05) supports sensitivity-first alerting; analyst-labelled outcomes feed governed retraining with drift monitoring, shadow validation, model registry, and rollback. The monitoring thresholds shown are design recommendations and were not validated in a live environment.

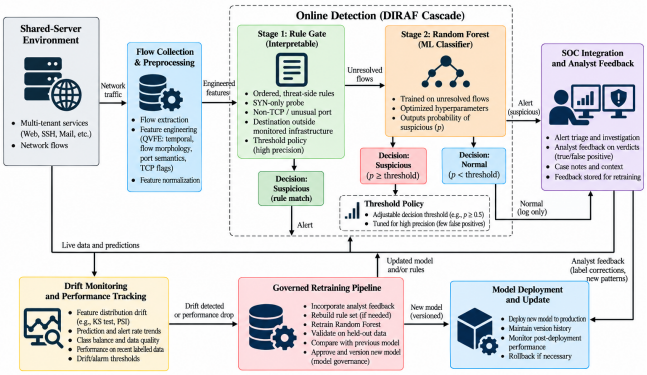


Fig. 1. Reference deployment architecture for DIRAF in a shared-server SOC pipeline, with threshold policy, analyst feedback, drift monitoring, and governed retraining.

This section presents the benchmark results for the evaluated models. It compares throughput, latency, model size, and training performance. These measures provide a concise view of computational efficiency and practical deployment requirements. Table XIII consolidates benchmark-based throughput, latency, model-size, and training measurements.

TABLE XIII
DEPLOYMENT-CAPACITY ANALYSIS (SCIKIT-LEARN 1.6, PYTHON 3.13, SINGLE TWO-CORE VM)

Dimension	Measured value	Basis
Mean / peak flow arrival rate	0,28 / 42 flows/s	CIDDS-001 timestamps; busiest second
DIRAF throughput (42,3% gated)	209.627 records/s	47.085-record test batch (Table IX)
DIRAF throughput (large batch)	Up to 281.369 records/s	500.000-record batch; larger batches amortize overhead
Throughput headroom vs. peak	≈4.991× (conservative); up to ≈6.699×	209.627/42; 281.369/42
Single-flow online latency	9,2 ms (p95 9,5 ms)	2.000 sequential single-record calls

JUTEI Edisi Volume. 10 No.2 Oktober 2026
ISSN 2579-3675, e-ISSN-5538
DOI 10.21460/jutei.2026.102.533

Dimension	Measured value	Basis
Model size / inference CPU	2,0 MB / 98%	Joblib-serialized 200-tree RF; psutil
RF training cadence	5,8 s (Model B), weekly feasible	Two-core VM (Xeon 2,60 GHz, 2,1 GB RAM)

Note. All values are benchmark measurements, not production trials. Batch-throughput ratios are not guarantees of end-to-end production capacity.

D. Limitations and Validity Threats

This evaluation uses one simulated CIDDS-001 capture from Week 1. Labels are operator-assigned; the capture has no verified attack-type annotations, and "suspicious" may encompass heterogeneous anomalies rather than confirmed malicious events. The data's endpoint-direction and flag-label coupling can make classification easier than in operational networks. The dataset dates from 2017 and may not represent current protocol and traffic distributions, including widespread QUIC or DNS-over-HTTPS. No independent dataset or live network was evaluated, so external validity remains unknown.

The threat-side gate is environment-specific: endpoint encoding, tunneling, NAT, proxies, CDNs, load balancers, or attacker adaptation may change rule coverage or purity. Perfect-precision gate induction can reduce coverage when signatures no longer match, but this conservative behavior under real temporal drift was not validated. Robustness tests are controlled perturbations on the same benchmark, not substitutes for longitudinal or cross-domain testing. SVM results rely on a stratified 20.000-record subsample. TAP depends on the selected feature space and attribution methods and should be rerun after model or feature changes. Future evaluation should include other CIDDS-001 weeks, independent datasets (including CICIDS 2017, CSE-CIC-IDS2018, UNSW-NB15, and CICIoT2023 [33]), temporal splits, open-set attacks [26], [27], and adversarial testing [28], [29].

VII. CONCLUSION

This study proposed and evaluated DIRAF, a rule-augmented RF cascade supported by QVFE and TAP, using the CIDDS-001 External Server Week 1 benchmark. On the held-out binary test set, the cascade achieved 100% accuracy, precision, recall, F1-score, and AUC. Its gate directly resolved 42,3% of test traffic and covered 61,9% of suspiciously labelled flows with 100% precision; mean batch latency decreased from 8,01 to 4,77 µs per record. The results are specific to this benchmark and do not demonstrate general-purpose intrusion-detection performance. Direction laundering reduced Model B accuracy to 68,8%; morphology was the most consequential QVFE view ablation, while the overall feature grouping primarily supports audit and analysis. Independent, temporal, and operational validation is required before transferability or deployment suitability can be established.

ACKNOWLEDGMENT

The author acknowledges Moshi Cooperative University and its Department of Information and Communication Technology for the research environment and computational resources. CIDDS-001 is publicly available through Hochschule Coburg.

ETHICS STATEMENT

This study used a publicly available benchmark dataset and involved no human participants, personally identifiable information, or privately collected human-subject data. Ethical approval was therefore not required for the reported analysis.

DATA AVAILABILITY

The CIDDS-001 External Server Week 1 data used in this study are publicly available through Hochschule Coburg (https://www.hs-coburg.de/cidds). No private or personally identifiable datasets were used.

CONFLICT OF INTEREST

The author declares no conflict of interest associated with this study.

AUTHOR CONTRIBUTIONS

The author conceived the study, developed the DIRAF, QVFE, and TAP framework, conducted the data analysis and experiments, interpreted the results, and prepared and revised the manuscript.

AI TOOL USE DECLARATION

An AI assistant was used to assist with manuscript language editing, restructuring, translation of the abstract into Bahasa Indonesia, and document formatting. The author is responsible for reviewing and verifying the final manuscript and all scientific content.

REFERENCES

[1] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Mach. Learn. Appl., vol. 12, Art. no. 100470, 2023, doi: 10.1016/j.mlwa.2023.100470.

[2] R. Usharani, E. Nirmala, N. Arockia Rosy, and P. Jagadeesan, "Optimization based machine learning for authenticated cloud security using anomaly detection and classification," Int. J. Mach. Learn. Cybern., vol. 17, no. 3, Art. no. 114, 2026, doi: 10.1007/s13042-025- 02956-8.

[3] S. Ali, C. Boufaied, D. Bianculli, P. Branco, and L. Briand, "A comprehensive study of machine learning techniques for log-based anomaly detection," Empir. Softw. Eng., vol. 30, Art. no. 129, 2025, doi: 10.1007/s10664-025-10669-3.

[4] M. T. Abdelaziz et al., "Enhancing network threat detection with random forest -based NIDS and permutation feature importance," J. Netw. Syst. Manage., vol. 33, Art. no. 2, 2024, doi: 10.1007/s10922-024-09874-0.

[5] A. Ahmed, M. Asim, I. Ullah, Zainulabidin, and A. A. Ateya, "An optimized ensemble model with advanced feature selection for network intrusion detection," PeerJ Comput. Sci., vol. 10, Art. no. e2472, 2024, doi: 10.7717/peerj-cs.2472.

[6] C. K. K. Reddy, P. A. Reddy, P. S. Reddy, M. Shuaib, S. Alam, S. Ahmad, and A. Rajaram, "Twined ensemble framework for network security: integrating Random Forest, AdaBoost, and Gradient Boosting for enhanced intrusion detection," Discover Internet Things, vol. 5, 2025, doi: 10.1007/s43926-025-00199-1.

[7] P. Azizidoost, S. Sarhsan Moghadam, E. Khezri, A. Basem, and M. Trik, "A new intrusion detection method using ensemble classification and feature selection," Sci. Rep., vol. 15, Art. no. 13642, 2025, doi: 10.1038/s41598-025-98604-w.

[8] S. A. Almahaqeri, M. H. Almourish, A. A. Nasser, A. S. A. Alghawli, A. A. K. Elsayed, and A. N. Alhejoj, "An optimized gradient boosting framework for IoT intrusion detection: a comprehensive evaluation on the CICIoT2023 dataset," Sci. Rep., vol. 16, 2026, doi: 10.1038/s41598-026-47399-5.

[9] A. Alharthi, M. Alaryani, and S. Kaddoura, "A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems," Array, vol. 26, Art. no. 100406, 2025, doi: 10.1016/j.array.2025.100406.

[10] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, and A. U. Rehman, "Enhancing intrusion detection: a hybrid machine and deep learning approach," J. Cloud Comput., vol. 13, Art. no. 123, 2024, doi: 10.1186/s13677-024-00685-x.

[11] A. Qaddos, M. U. Yaseen, A. S. Al Shamayleh, M. Imran, A. Akhunzada, and Z. S. Alharthi, "A novel intrusion detection framework for optimizing IoT security," Sci. Rep., vol. 14, Art. no. 21789, 2024, doi: 10.1038/s41598-024-72049-z.

[12] M. Umer, M. Tahir, M. Sardaraz, M. Sharif, H. Elmannai, and A. D. Algarni, "Network intrusion detection model using wrapper -based feature selection and multi head attention transformers," Sci. Rep., vol. 15, Art. no. 28718, 2025, doi: 10.1038/s41598-025-11348-5.

[13] M. S. Ataa, E. E. Sanad, and R. A. El khoribi, "Intrusion detection in software defined network using deep learning approaches," Sci. Rep., vol. 14, Art. no. 29159, 2024, doi: 10.1038/s41598-024-79001-1.

[14] L. Ofusori, T. Bokaba, and S. Mhlongo, "Artificial intelligence in cybersecurity: A comprehensive review and future direction," Appl. Artif. Intell., vol. 38, no. 1, Art. no. 2439609, 2024, doi: 10.1080/08839514.2024.2439609.

[15] P. Himler, M. Landauer, F. Skopik, and M. Wurzenberger, "Anomaly detection in log event sequences: A federated deep learning approach and open challenges," Mach. Learn. Appl., vol. 16, Art. no. 100554, 2024, doi: 10.1016/j.mlwa.2024.100554.

[16] P. Han, H. Li, G. Xue, and C. Zhang, "Distributed system anomaly detection using deep learning based log analysis," Comput. Intell., vol. 39, no. 3, pp. 433–455, 2023, doi: 10.1111/coin.12573.

[17] X. Ye et al., "Research on insider threat detection based on personalized federated learning and behavior log analysis," Sci. Rep., vol. 15, Art. no. 19214, 2025, doi: 10.1038/s41598-025-04029-w.

[18] D. Goyal, S. K. Singh, S. Kumar, V. Chilkoti, K. T. Chui, C. H. Hsu, and B. B. Gupta, "An explainable AI framework for intrusion detection using SHAP and LIME," in Proc. 14th IEEE Global Conf. Consum. Electron. (GCCE), Osaka, Japan, 2025, pp. 1315–1319, doi: 10.1109/GCCE65946.2025.11275089.

[19] M. Pawlicki, A. Pawlicka, S. Szelest, R. Kozik, and M. Choraś, "Class based SHAP analysis for improved explainability insights in NIDS," in Applied Intelligence (ICAI 2024), Commun. Comput. Inf. Sci., vol. 2387. Singapore: Springer, 2025, pp. 303–313, doi: 10.1007/978-981- 96-1907-8_29.

[20] S. Gupta and B. Singh, "An intelligent multi -layer framework with SHAP integration for botnet detection and classification," Comput. Secur., vol. 140, Art. no. 103783, 2024, doi: 10.1016/j.cose.2024.103783.

[21] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi -layer perceptron," IEEE Access, vol. 12, pp. 30164 –30175, 2024, doi: 10.1109/ACCESS.2024.3368377.

[22] F. Jalalvand, M. Baruwal Chhetri, S. Nepal, and C. Paris, "Alert prioritisation in security operations centres: A systematic survey on criteria and methods," ACM Comput. Surv., vol. 57, no. 2, pp. 1 –36, 2024, doi: 10.1145/3695462.

[23] J. Ghadermazi, A. Shah, and J. Jajodia, "A machine learning and optimization framework for efficient alert management in a cybersecurity operations center," ACM Digit. Threats: Res. Pract., vol. 5, no. 2, pp. 1–23, 2024, doi: 10.1145/3644393.

[24] M. A. Shyaa, N. F. Ibrahim, Z. Zainol, R. Abdullah, M. Anbar, and L. Alzubaidi, "Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems," Eng. Appl. Artif. Intell., vol. 137, Art. no. 109143, 2024, doi: 10.1016/j.engappai.2024.109143.

[25] G. Apruzzese, A. Fass, and F. Pierazzi, "When adversarial perturbations meet concept drift: An exploratory analysis on ML NIDS," in Proc. 2024 Workshop Artif. Intell. Security (AISec), 2024, pp. 149–160, doi: 10.1145/3689932.3694757.

[26] J. Fang and C. Xie, "Unknown intrusion traffic detection method based on unsupervised learning and open set recognition," Sci. Rep., vol. 15, Art. no. 17001, 2025, doi: 10.1038/s41598-025-01084-1.

[27] G. Y. Shin, D. W. Kim, and M. M. Han, "Open set recognition with dissimilarity weight for unknown attack detection," IEEE Access, vol. 11, pp. 102381–102390, 2023, doi: 10.1109/ACCESS.2023.3277871.

[28] J. Vitorino, I. Praça, and E. Maia, "SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection," Comput. Secur., vol. 134, Art. no. 103433, 2023, doi: 10.1016/j.cose.2023.103433.

[29] A. Alotaibi and M. A. Rassam, "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense," Future Internet, vol. 15, no. 2, Art. no. 62, 2023, doi: 10.3390/fi15020062.

[30] S. Uno, H. Noma, and M. Gosho, "Firth type penalized methods of the modified Poisson and least squares regression analyses for binary outcomes," Biometrical J., vol. 66, no. 4, Art. no. e202400004, 2024, doi: 10.1002/bimj.202400004.

[31] S. Suhas et al., "Firth's penalized logistic regression: a superior approach for analysis of data from India's National Mental Health Survey, 2016," Indian J. Psychiatry, vol. 65, no. 11, pp. 1208 –1213, 2023, doi: 10.4103/indianjpsychiatry.indianjpsychiatry_827_23.

[32] G. Heinze, R. Puhr, and D. Dunkler, logistf: Firth's bias reduced logistic regression, R package version 1.26.1, 2025. [Online]. Available: https://cran.r -project.org/package=logistf, accessed Aug. 20, 2026.

[33] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real time dataset and benchmark for large scale attacks in IoT environment," Sensors, vol. 23, no. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.

[34] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Creation of flow based data sets for intrusion detection," J. Univers. Comput. Sci., vol. 23, no. 11, pp. 1087–1113, 2017, doi: 10.3217/jucs-023-11- 1087.