

Original Article

SpeciNet-IDS: A False-Alarm-Aware Intrusion Detection System for Reliable Network Security

Sailaja Rani Setty¹, Valli Kumari Vatsavayi²

^{1,2}Computer science and Systems Engineering, Andhra University, Andhra Pradesh, India.

¹Corresponding Author : sailajaradhika@gmail.com

Received: 29 January 2026

Revised: 25 July 2026

Accepted: 13 August 2026

Published: 30 September 2026

Abstract - In practice, false alarms still plague the ability of deployed intrusion detection systems to be effective. If models are primarily tuned for accuracy, with unbalanced traffic, they're overly tolerant of benign misclassifications; analysts become overwhelmed by a rising number of false alarms, and real threats go unnoticed. We suggest a new hybrid network (CNN-Dense) called SpeciNet, which is trained with specificity, rather than accuracy. There are 4 mechanisms that function together. A parallel Conv1D branch learns local statistical patterns, and a Dense Residual branch learns multi-feature global interactions that are important for rare attack classes. The training is based on a focal loss ($\gamma=2$, $\alpha_r=0.75$) which pulls towards flows that are borderline benign behind the majority of false alarms. The operating threshold is selected based on the far target calibration process, which is a threshold that is as high as possible and does not exceed the 2% validation far threshold. The gains are confirmed by a SHAP-based attribution gate (not used after configuration is selected), but during configuration selection. We evaluate on CICIDS2017 as the primary benchmark and NSL-KDD for corroboration. SpeciNet-IDS reaches binary false alarm rates of 0.01% and 2.71%, with specificity of 99.99% and 97.29%. On CICIDS2017, only 16 false alarms occur across 318,547 benign test instances - a 13-fold reduction over the strongest baseline, significant against all baselines (McNemar $p < 0.001$). On NSL-KDD, McNemar's test shows significant FAR reductions over three of four baselines and equivalence with the best classical model. Component analysis identifies three dominant sources of false alarm suppression - class weighting, the Dense Residual branch, and threshold calibration - each contributing measurably and independently.

Keywords - Intrusion detection systems, False alarm rate, Hybrid CNN-Dense architecture, Focal loss, Threshold calibration, IDS using Deep Learning-based models, Network security, Explainable Artificial Intelligence.

1. Introduction

Securing a modern network is fundamentally a classification problem. Every packet, flow, or session must be sorted into legitimate or threatening, and that decision must be made continuously, across very large volumes of traffic, with limited human review. For years, the research community gauged progress primarily through detection accuracy, defined as the proportion of flows labelled correctly. Accuracy improved, architectures became more elaborate, and benchmark scores advanced steadily. Despite these gains, the detectors deployed in production environments remain unreliable in one specific and costly respect: they raise an excessive number of false alarms [1, 2].

Consider the workload inside a security operations centre during a typical shift. Analysts confront hundreds or thousands of alerts, the majority of which correspond to benign traffic that the detector has flagged in error. Each alert investigated consumes time and attention that cannot be recovered. As the shift progresses, fatigue accumulates, and genuine attacks concealed among the false positives go

undetected - not because the system failed to flag them, but because the analyst was unable to reach those alerts before the response window had closed [8].

Axelsson [4] formally characterized this difficulty more than two decades ago as the base-rate fallacy. Because malicious traffic constitutes only a small fraction of total network volume, even a model achieving 99% accuracy may still produce more false positives than true positives in absolute terms. The metric that ultimately determines operational utility is therefore not accuracy but specificity: the probability that a benign flow is correctly identified as benign.

The transition to deep learning has not resolved this tension. LSTM-based detectors [20], hybrid convolutional-recurrent designs [21], and multi-tiered ensembles [22] all report strong benchmark performance, yet the underlying training objective remains unchanged: minimise cross-entropy, adopt a fixed probability threshold, and report accuracy. Transformer-based models [24] and graph-neural network approaches [26, 27] have extended the architectural



frontier further, yet the specificity problem is absent from their evaluation criteria. Explainability tools, including SHAP [9] and LIME [10], have been integrated into several frameworks, but as diagnostic accessories applied after the model is finalised rather than as instruments that guide it toward lower false alarm rates [28, 29].

Binary detection - which reduces the intrusion problem to a single benign-versus-attack boundary - offers a more controllable setting for specificity optimisation than multi-class formulations [7, 8]. When a model must simultaneously maintain K attack-specific boundaries, each defined by a small minority of training samples, several of those boundaries inevitably encroach on the benign feature region and generate false alarms on legitimate traffic that happens to resemble minority attack classes [5, 8].

This paper presents SpecINet-IDS framework that treats false alarm rate as a primary training objective rather than an incidental consequence of accuracy optimization. The framework realizes this principle through four mechanisms:

- Hybrid CNN-Dense architecture - A Conv1D branch extracts local sequential patterns within the feature vector, while a parallel Dense Residual branch models the global multi-feature interactions that convolution alone is unable to capture.
- Focal loss with asymmetric class weighting - Update gradients are focused on the harder to classify borderline benign flows, directly in the loss function surface instead of after the training.
- FAR-target threshold calibration - Instead of using a fixed threshold, the operating point for the FAR-target threshold calibration is determined to be the highest point capable of meeting a specified false alarm rate.
- Attribution-based quality gate - SHAP is applied during configuration selection to confirm that specificity improvements stem from genuine traffic semantics rather than from statistical artefacts.

The rest of this paper is structured as follows: Section 2 surveys Related work. Section 3 describes the proposed Methodology. Sections 4 and 5 present the Results and analysis of experiments. Section 6 Concludes the paper with Future research directions.

2. Related Work

2.1. Intrusion Detection: Traditional and Statistical

Early deployments of IDS were based on signature libraries, which contained lists of well-known sequences of bytes in traffic to be monitored. This technique is safe when an attack has already been identified, and no encryption is used, but will not protect against variants of attacks that differ in just a few bits and will not protect against new attacks. The anomaly detection relaxed this requirement, instead of

signatures, it builds a profile of normal behaviour to detect its deviation [11]. Axelsson's analysis [4] quantified the difficulty: when attack rate is low, the absolute error rate of any faulty classifier is more likely to be the number of false positive than the number of true positive, so the operational criterion for viability is specificity.

2.2. Intrusion Detection using Machine Learning

The supervised learning treated the boundary of benign/malicious more principled. Support vector machines, random forests and gradient-boosted decision trees all offered ways to learn complex discriminative boundaries from labelled data, and feature engineering techniques [13, 25] eliminated redundant and misleading attributes that would otherwise be detrimental to the quality of the boundaries [2, 12]. This enhanced the accuracy of the benchmark, but the goal of the training remained unchanged: to minimize classification error. Class imbalance was addressed by resampling and cost-sensitive weighting, but it was not explicitly formulated, rather it was a by-product of specificity enhancement.

2.3. Binary vs. Multi-Class Intrusion Detection

Labeling traffic as benign versus attack, and differentiating between various attack categories, have direct implications for false alarm behavior [7, 8]. Each binary formulation maintains one decision boundary, which can be set and tuned comparatively precisely. Multi-class formulations require to set K boundaries simultaneously (one per attack class), and each boundary for each minority class has very few training samples. Under-determined boundaries will tend to spill over into the benign feature region, resulting in false alarms on real traffic that is physically near a minority attack cluster.

2.4. Deep Learning-Driven Intrusion Detection

Deep learning significantly improved the accuracy limit for intrusion detection [3]. One-dimensional convolutional networks were found by V.V. Mandhare et al. [16] to be capable of learning statistical dependencies present in flow feature vectors that are hardcoded by shallow classifiers completely. Yin et al. [20] demonstrated that LSTM units are able to capture the temporal structure in sequential packet arrivals. Ferrag et al. [23] presented a detailed survey that shows that deep architectures outperform classical machine learning in 10 different benchmarks. More recent papers have further extended these results with multi-layered hybrid approaches [22], and transformer-based attention models [24] showing further accuracy gains, and graph-neural-network approaches [26, 27] suggesting a relational representation of traffic. Even with these developments, specificity remains absent from the primary evaluation criteria throughout this body of work. MTH-IDS [22] is a partial exception in that it incorporates class-weighted loss, but it retains a fixed threshold, leaving available specificity gains unexploited.

2.5. Explainability in Intrusion Detection Systems

The opacity of deep neural networks created a parallel research thread focused on making IDS decisions auditable. SHAP [9] and LIME [10] became dominant attribution tools [17]. Marino et al. [14] investigated adversarial approaches to explainability; Mane and Rao [7] found that SHAP-annotated alerts increase analyst confidence; Barnard et al. [15] showed that integrating XAI can improve decision stability; Gaspar et al. [18] benchmarked LIME and SHAP across IDS datasets; A. Albashayreh et al. [19] extended XAI-based detection to identify the most significant features; Zebin et al. [28] and Amarasinghe et al. [29] each demonstrated that SHAP can retrospectively explain false alarm behaviour.

In every case, however, attribution is applied after training concludes. To the best of our knowledge, SpecINet-IDS is the first framework to use SHAP during configuration selection as a forward-looking quality gate rather than a post-hoc diagnostic tool.

2.6. New Developments and Research Gaps

Three issues remain unaddressed in prior work. To the best of our knowledge, no published IDS framework simultaneously combines a hybrid CNN-Dense architecture with focal loss and calibrated threshold selection to explicitly

target FAR as a primary training objective. The specificity difference between binary and multi-class formulations has not been measured under controlled conditions. Attribution-based explainability has not been deployed as a prospective validation mechanism during model development. SpecINet-IDS addresses all three within a single coherent framework.

3. Methodology

The proposed SpecINet-IDS is a specificity-driven IDS framework developed to improve the operational reliability of intrusion detection systems by emphasising the reduction of false alarms over accuracy-driven optimisation. The methodology utilises a unified data preprocessing stage, a hybrid CNN-Dense detection model, specificity-oriented training strategies, and an explainability-based validation method to evaluate the detection performance of SpecINet-IDS under both binary and multi-class settings.

The entire workflow of the proposed framework is presented in Figure 1 below and illustrates the Proposed SpecINet-IDS architecture for specificity-driven intrusion detection. SpecINet-IDS treats detection reliability - specifically false alarm suppression - as a design constraint that shapes every component choice, from the loss function through to the final operating threshold.

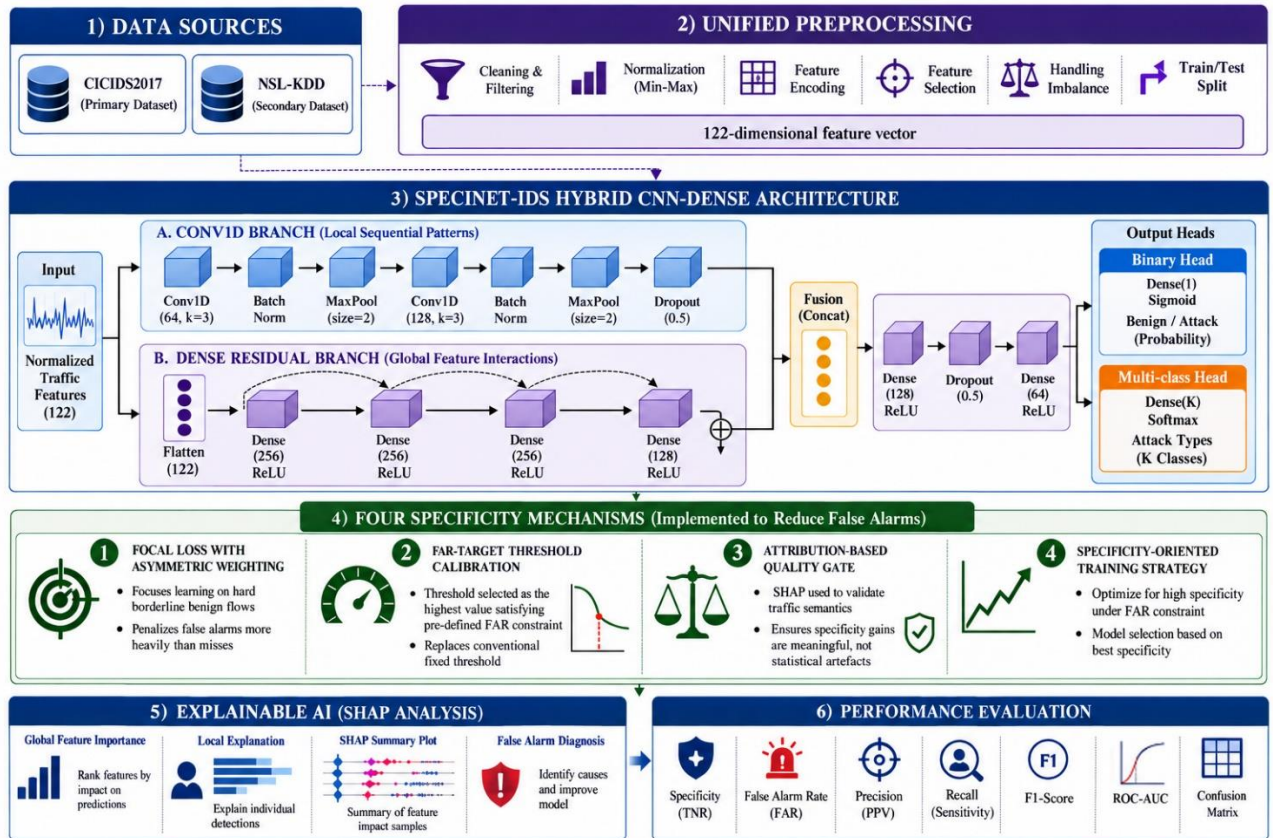


Fig. 1 Proposed SpecINet-IDS architecture for specificity-driven intrusion detection

3.1. Dataset and Preprocessing

The proposed hybrid CNN-Dense IDS framework is evaluated on two benchmark datasets: CICIDS2017 as the primary benchmark and NSL-KDD as the corroborating benchmark.

CICIDS2017 uses the full seven-day partition, with Monday excluded (benign only) [6, 30]. Six included days yield 2,018,473 clean records across 15 attack classes. MinMaxScaler normalization is applied. Split: Train = 1,291,822, Val = 322,956, Test = 403,695. Inverse-frequency class weighting is applied for binary training. SMOTE oversamples all multi-class minority categories (< 10k samples) to 10,000; multi-class weights are additionally capped at a maximum multiplier of 10× to prevent training instability on severely underrepresented classes. NSL-KDD contains 125,973 labelled training flows and 22,544 test flows across five traffic classes [5]; the binary and multi-class label mappings are given in Table 1. Three categorical attributes are one-hot encoded; all features are min-max scaled to [0, 1]. Split: Train = 100,778, Val = 25,195, Test = 22,544 (80:20 stratified). Inverse-frequency class weighting is applied for binary training. SMOTE oversamples R2L to 5,000 and U2R to 2,000 ($k = \min(5, \text{class_size} - 1)$) for multi-class training.

Table 1. Dataset encoded label mapping

Dataset	Binary	Multi-Class Mapping-Class Mapping
NSL-KDD	0=Normal, 1=Attack	Normal-0; DoS-1; Probe-2; R2L-3; U2R-4
CICIDS2017	0=BENIGN, 1=Attack	BENIGN-0; Bot-1; DDoS-2; DoS Golden Eye-3; DoS Hulk-4; DoS Slowhttptest-5; DoS slowloris-6; FTP-Patator-7; Heartbleed-8; Infiltration-9; PortScan-10; SSH-Patator-11; Web-BF-12; Web-SQLi-13; Web-XSS-14

3.2. Mathematical Framework

This section describes the mathematical framework upon which SpecNet-IDS operates. The description covers the representation of network traffic data, the feature extraction and learning process for both parallel branches, output modelling for binary and multi-class detection, and the optimisation objectives that embed specificity into the training procedure.

3.2.1. Representation and Partitioning of the Dataset

The data representing the network traffic is represented as follows (referred in Equation (1)):

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \quad (1)$$

where $x_i \in \mathbb{R}^{122}$ is the normalised feature vector of the i -th network traffic sample, and y_i in $\{0, \dots, K-1\}$ is its corresponding class label. Stratified sampling is applied to ensure that class proportions are preserved under class imbalance conditions. An 80:20 split partitions the dataset into training and test sets.

3.2.2. Input Representation and Feature Extraction

Conv1D Branch treats each traffic sample as a 122-dimensional feature vector reshaped to (122, 1), enabling Conv1D layers to capture local statistical structure within the feature sequence and detect local co-activation patterns. The output of each convolutional layer $h^{(l)}$ over traffic sample x is defined by (Equation (2)):

$$h^{(l)} = \sigma \left(\text{BN}(W^{(l)} * h^{(l-1)}) \right) \quad (2)$$

Where $W^{(l)}$ are the learned filter weights for layer l with no bias term (bias is redundant when Batch Normalization follows); $*$ denotes 1D convolution with kernel size $k=3$; $\text{BN}(\cdot)$ is Batch Normalization applied before activation; $\sigma(\cdot)$ is ReLU. MaxPool1D halves the spatial dimension after the first two blocks; Dropout ($p=0.2$) follows each block for regularization. GlobalMaxPool1D aggregates the final sequence by retaining the strongest activation at each filter position, producing $z_{\text{cnn}} \in \mathbb{R}^{128}$.

Each residual block within the Dense Residual branch computes its output as (Equation (3)):

$$z_{\text{out}} = \sigma \left(\text{BN}(\text{Dense}(z_{\text{in}})) \right) + \text{projection}(z_{\text{in}}) \quad (3)$$

Where z_{in} is the block input vector; $\text{Dense}(\cdot)$ is a fully connected layer with no bias term; $\text{BN}(\cdot)$ is Batch Normalization; $\sigma(\cdot)$ is ReLU activation; $\text{projection}(z_{\text{in}})$ is a linear mapping applied only when input and output dimensions differ (e.g., 256→128 in Block 2), otherwise $\text{projection}(z_{\text{in}})=z_{\text{in}}$. The skip connection drives gradients cleanly through deep residual stacks, keeping the signal intact where plain stacks would let it decay. Its final output $\in \mathbb{R}^{128}$ captures all 122 features at once, free of any spatial ordering assumption.

3.2.3. Output Modeling for Binary and Multi-Class Detection

For binary detection, the output head applies sigmoid: $\hat{y}$ in [0,1], the probability that a flow is malicious. For multi-class detection, the model classifies across $K=15$ classes for CICIDS2017 and $K=5$ classes for NSL-KDD (Normal, DoS, Probe, R2L, U2R). The SoftMax function computes the predicted probability for each class k (Equation (4)):

$$\hat{y}_k = \frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)}, \quad k = 1, \dots, K \quad (4)$$

ensuring the output probabilities sum to one. This formulation enables a consistent comparison between binary (coarse-grained) and multi-class (fine-grained) intrusion detection under identical architectural conditions.

3.2.4. Optimization Objectives and Training Strategy

The binary detection model is optimized using Focal Loss (Equation (5)):

$$FL(p_t) = -\alpha_t \cdot (1 - p_t)^\gamma \cdot \log(p_t) \quad (5)$$

Where $p_t = \hat{y}$ when $y=1$ (attack sample), else $p_t = 1-\hat{y}$ (benign sample); $\alpha_t = 0.75$ when $y=1$ (higher penalty on missed attacks), $\alpha_t = 0.25$ when $y=0$ (lower penalty on correctly classified benign), concentrating learning on attack-class errors; $\gamma = 2.0$ is the focusing parameter. The modulating factor $(1 - p_t)^\gamma$ down-weights easy, well-classified examples and concentrates gradient updates on hard borderline benign flows - the primary source of false alarms. When a benign flow is confidently misclassified as an attack ($p_t \rightarrow 0$), $(1 - p_t)^\gamma \rightarrow 1$ and the full gradient penalty is applied; when correctly classified with high confidence, the factor $\rightarrow 0$ and the loss contribution is suppressed. This embeds the specificity objective directly into the loss surface rather than imposing it as a post-training correction.

FAR-target threshold calibration selects the operating point enforcing the 2% false alarm constraint at inference (Equation (6)):

$$\tau^* = \max\{\tau \in T : FAR_{val}(\tau) \leq 0.02\}, \quad T = \{0.30, 0.35, \dots, 0.80\} \quad (6)$$

where $FAR_{val}(\tau) = FP(\tau) / (FP(\tau) + TN(\tau))$ is the false alarm rate on the validation partition at threshold τ ; T is the 11-point calibration grid; τ^* is the selected operating threshold applied at test time.

For multi-class detection, the model is trained with Sparse Categorical Cross-entropy (Equation (7)). Because this formulation takes integer class indices directly, it removes the need for one-hot label encoding during training:

$$L_{multi} = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}_{i,y_i}) \quad (7)$$

Here, $\hat{y}_{i,y_i}$ is the SoftMax probability the model assigns to the true class label y_i for the i -th training sample, and N denotes the number of training samples in the batch.

3.3. Model Architecture of SpecNet-IDS

The SpecNet-IDS hybrid architecture (Figure 1) follows from a single structural observation. A standalone Conv1D branch interprets the 122 tabular features as a spatial sequence and extracts local statistical patterns from it, yet it cannot at the same time perceive the global multi-feature combinations

that characterize the rare R2L and U2R attack classes [16, 22]. The key difference between these categories is a collection of subtle, concerted deviations over a number of features at a time, and a signature of that form is not captured by the local convolution windows. The Dense Residual branch is added to address exactly this problem, because it operates on all 122 features, without assuming anything about their relative spatial location. The two branches work together to provide a complementary view of the same input, local sequential structure on one branch and global feature interaction on the other, that is then combined together into a single unified view before being classified.

The Conv1D branch reshapes the input to (122, 1) so that convolution filters can slide across the feature sequence and detect local co-activation patterns - the kind of short-range statistical regularities that distinguish DoS flooding from normal traffic. Three convolutional blocks with progressively wider filter banks (64, 64, 128 filters, kernel size $k=3$) extract increasingly abstract feature representations, with Batch Normalisation and ReLU activation following each convolution. MaxPool1D halves the spatial dimension after the first two blocks, and Dropout ($p=0.2$) regularises filter learning across the class-imbalanced training set.

A Multi-Head Attention layer (4 heads) is inserted after the second block to capture non-local dependencies that fixed-size convolution kernels cannot reach, with a residual Add and Layer Normalisation preserving gradient flow. GlobalMaxPool1D aggregates the final sequence by retaining the strongest activation at each filter position - preferred over GlobalAveragePool1D because max pooling preserves the most pronounced local attack signal rather than diluting it with the larger volume of benign-like feature values. The output is $z_{cnn} \in \mathbb{R}^{128}$.

The Dense Residual branch operates directly on the flat 122-dimensional input without any spatial reshaping, allowing every feature to interact with every other feature simultaneously. An initial Dense (256) projection with Batch Normalisation and ReLU expands the representation into a higher-dimensional space. Two residual blocks then refine it.

In ResBlock1, the input passes through a Dense 256) layer followed by Batch Normalisation, ReLU, and Dropout (0.3), and a skip connection adds the block input back to its output, which preserves gradient flow and prevents the representation from degrading as depth increases. ResBlock2 narrows the dimension from 256 to 128 via a Dense (128) layer followed by Batch Normalisation, ReLU, and Dropout (0.3), with a linear projection skip connection to match dimensions. Residual connections are essential here because without them, deeper Dense stacks on tabular IDS data tend to overfit to majority-class benign patterns rather than learning the sparse multi-feature signatures of minority attack classes. The output is $z_{dense} \in \mathbb{R}^{128}$.

The two branch outputs are concatenated into a 256-dimensional hybrid representation combining local and global feature evidence before the classification head. Two Dense layers (128 and 64 units) with Batch Normalisation, ReLU, and Dropout (0.4 and 0.3, respectively) perform the final discriminative mapping, and a single sigmoid output unit produces the binary attack probability. The total parameter count is 262,017 - larger than a shallow CNN but fully justified: component validation (Section 5.1) shows that removing the Dense Residual branch raises FAR by 3.48 pp, confirming every parameter performs essential detection work rather than memorising benign traffic fluctuations. The 256-dimensional merged representation z also serves as the anchor for the SHAP attribution analysis in Section 3.5, enabling attribution to be computed jointly over both local and global feature contributions

3.4. Hyperparameter Selection Process

Every hyperparameter in SpecNet-IDS is selected with a specific FAR consequence in mind rather than for general training stability. The learning rate, batch size, dropout rates, focal loss parameters, and threshold calibration grid are each chosen to suppress false alarms at a distinct point in the training and inference pipeline. The five parameters that directly drive false alarm suppression are summarised in Table 2; the three most impactful - focal loss, τ^* calibration, and class weighting - are discussed in Sections 3.2.4 and 5.1.

Table 2. Final hyperparameter configuration

Parameter	Value	Rationale
Focal loss (γ / α_i)	2.0 / 0.75 (attack), 0.25 (benign)	Concentrates gradient updates on hard borderline benign flows
Calibration grid T	{0.30, 0.35, ..., 0.80}, step 0.05	τ^* = highest threshold satisfying val FAR $\leq$ 2%
Class weighting	Inverse-frequency, cap $10\times$	Amplifies penalty on false alarm errors
Dropout (Conv / Dense)	0.2 / 0.3–0.4	Prevents overfitting to majority benign class
Learning rate	0.001	Stable convergence on focal loss surface

3.5. Attribution-based Validation

SpecNet-IDS uses SHAP during configuration selection rather than as a post-hoc interpretability step. A KernelExplainer is fitted on 500 background instances compressed to 20 centroids via k-means, then applied to 150 validation instances. Four quality checks are applied before a configuration is accepted. Attribution mass concentration is the first thing we check. If a single feature ends up contributing more than 25% of the total mean $|\text{SHAP}|$, that points to the model exploiting a spurious shortcut rather than

drawing on genuine traffic semantics. Next, we compare the global importance ranking against domain knowledge to confirm that the top-ranked features line up with established attack signatures. A third check looks at local explanations, taken separately for true positives, true negatives, and false positives, to verify that the model fires for the right reasons in each of these outcome categories. Fourth, the FP-TP attribution correlation r is computed between the mean SHAP profiles of false positives and true positives. A result of $r \approx 0$ confirms that false alarms arise from borderline benign flows that individually resemble attack traffic - a data ambiguity problem. A result of $r \approx 1$ would indicate the model detects attacks and generates false alarms for the same systematic reasons, signalling model instability requiring architectural correction [9, 28, 29].

4. Results and Experimental Set-up

4.1. Evaluation Setup

Two specificity-oriented metrics serve as primary evaluation criteria. The False Alarm Rate ($\text{FAR} = \text{FP} / (\text{FP} + \text{TN})$) measures the fraction of benign flows incorrectly classified as attacks; Specificity ($\text{TN} / (\text{TN} + \text{FP})$) measures the fraction correctly passed as benign. Recall is reported alongside FAR to characterize the detection-specificity trade-off. SpecNet-IDS is evaluated across four configurations - CICIDS2017 binary, CICIDS2017 multi-class, NSL-KDD binary and NSL-KDD multi-class. Baselines are evaluated at the conventional fixed threshold $\tau = 0.5$; SpecNet-IDS operates at $\tau^* = 0.80$ selected by FAR-target calibration. All experiments use identical preprocessing, stratified splits, and random seed 42. CICIDS2017 is treated as the primary benchmark given its scale, realism, and class diversity; NSL-KDD serves as the corroborating benchmark.

4.2. Experimental Results on CICIDS2017 Dataset

CICIDS2017 is the paper's primary benchmark and the site of the paper's strongest result. On the full 7-day test partition ($N = 403,695$): FAR = 0.01%, Specificity = 99.99%, Recall = 82.62%, Precision = 99.98%, AUC = 0.9986 at $\tau^* = 0.80$. Only 16 false alarms arise from 318,547 benign test instances - a false alarm burden so low it would be operationally invisible in a deployed network sensor. Figure 2 shows the calibration grid: thresholds $\tau = 0.30$ – 0.50 exceed the 2% FAR target (red zone); the first within-target threshold is $\tau = 0.55$ (val FAR = 0.89%); $\tau^* = 0.80$ is selected as the highest within-target value. The fact that $\tau^* = 0.80$ is identical across CICIDS2017 and NSL-KDD - two datasets with fundamentally different class distributions, attack taxonomies, and scale - confirms that the calibration procedure discovers a stable operating point from training statistics rather than fitting to a specific benchmark.

The central quantitative finding of this paper is the 13-fold reduction in FAR relative to Random Forest (0.01% versus 0.13%, McNemar $p < 0.001$). Threshold adjustment cannot close this gap, because RF was trained without any

specificity objective, and its score distribution cannot sustain low false alarm rates and high recall at the same time. SpecINet-IDS attains $\text{FAR} = 0.01\%$ precisely because specificity is built into every stage of the training pipeline - the loss function, the class weighting, and the calibration - rather than applied as a correction after training. The recall of 82.62%, compared with 98.90% for RF, follows from the

explicit $\text{FAR} \leq 2\%$ operating constraint: at $\tau = 0.50$, the recall would rise further, but FAR would then exceed the 2% target. This represents the transparent and measurable cost of the constraint, and it is a deliberate design choice rather than a shortcoming. The multi-class FAR of 5.98% arises from SoftMax score dispersion across the 15 fine-grained classes, since the τ^* mechanism is defined for binary detection only.

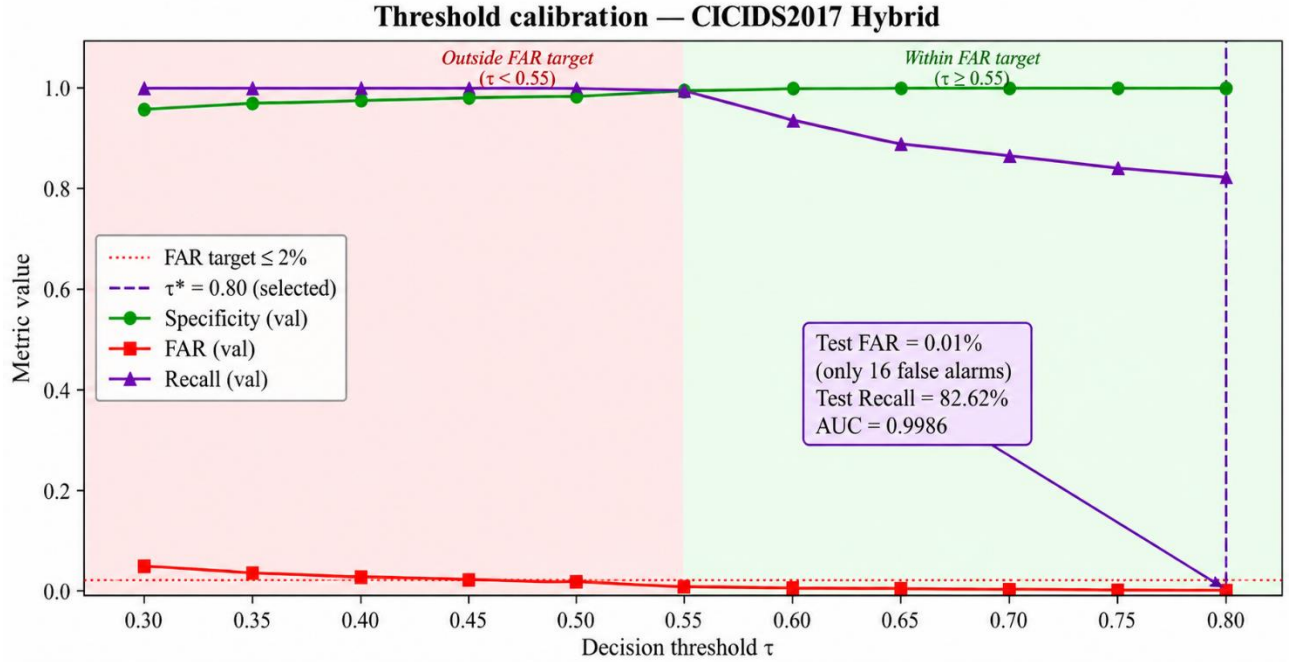


Fig. 2 FAR-target threshold calibration curve -CICIDS2017 binary

4.3. Experimental Results on NSL-KDD Dataset

The binary model converged at epoch 55 (best: epoch 43, 5.2 s/ep). All 11 calibration thresholds ($\tau = 0.30$ – 0.80) satisfy the 2% FAR target on the validation set; $\tau^* = 0.80$ is selected as the highest within-target value. Test-set performance: $\text{FAR} = 2.71\%$, $\text{Specificity} = 97.29\%$, $\text{Recall} = 64.16\%$, $\text{Precision} = 96.90\%$, $\text{AUC} = 0.8469$. Confusion matrix: $\text{TN} = 9,448$ | $\text{FP} = 263$ | $\text{FN} = 4,599$ | $\text{TP} = 8,234$.

The Recall of 64.16% is a direct and expected consequence of the $\tau^* = 0.80$ operating point: the NSL-KDD test partition contains 3–4 $\times$ more novel R2L and U2R variants than the training set, and these score between 0.40 and 0.79 - below threshold. This is the designed specificity–recall trade-off, not a model deficiency. The multi-class model achieves $\text{FAR} = 2.65\%$, $\text{Specificity} = 97.35\%$, confirming that specificity is maintained across detection granularities. Accuracy (78.43%) is lower than baselines because $\tau^* = 0.80$ deliberately increases false negatives to suppress false alarms; accuracy is not an optimisation target. Notably, SpecINet-IDS achieves this FAR under an explicit $\leq 2\%$ training constraint, whereas RF achieves its marginal 0.05 pp advantage at an unconstrained default threshold - a fundamentally less demanding operating condition. Figure 3 shows the NSL-

KDD ROC curve ($\text{AUC} = 0.8469$) with the operating point at $\tau^* = 0.80$ marked.

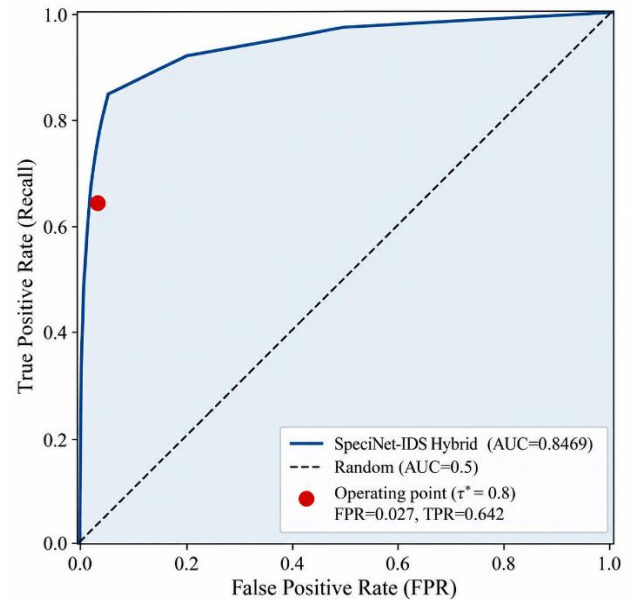


Fig. 3 ROC curve - NSL-KDD binary

4.4. Comparative Performance

Table 3 summarizes SpecNet-IDS across all four evaluated configurations - binary and multi-class on both datasets - providing a complete picture of the framework

before the baseline comparison. Tables 4 and Table 5 then present the comparative results against classical and shallow deep learning baselines on CICIDS2017 and NSL-KDD, respectively.

Table 3. Specinet-IDS Hybrid - Complete results (all configurations)

Dataset/ Mode	FAR	Spec	Acc.	Recall	Prec.	AUC/ tau*
NSL-KDD/ (Binary)	2.7%	97.29%	78.4%	64.16%	96.90%	0.8469 / tau*=0.8
NSL-KDD (Multi-class)	2.65%	97.35%	77.7%	66.40%	-	-
CICIDS2017 (Binary)	0.01%	99.99%	96.3%	82.62%	99.98%	0.9986 / tau*=0.8
CICIDS2017 (Multi-class)	5.98%	94.02%	95.17%	99.84%	-	-

Highlighted = lowest FAR per dataset. $\tau = 0.80$ on both datasets via FAR-target calibration (Equation (6)). Precision and AUC omitted for multi-class: both are class-dependent and misleading as single aggregates. τ^ calibration applies to binary detection only**

The comparison is architecturally fair: baselines operate at the unconstrained default $\tau = 0.5$, trained without any FAR objective. SpecNet-IDS operates at $\tau^* = 0.80$, trained with specificity as a first-class objective - a strictly more demanding experimental condition. Accuracy is omitted from analysis: all baselines report higher accuracy because they optimize for majority-class correct classifications; SpecNet-IDS accepts lower accuracy to guarantee $\text{FAR} \leq 2\%$.

On CICIDS2017 - the primary benchmark - $\text{FAR} = 0.01\%$ is $13\times$ below RF and XGBoost (0.13%), with $p < 0.001$ versus all four baselines. On NSL-KDD, $\text{FAR} = 2.71\%$ is below SVM (7.17%), MLP (3.33%), and XGBoost (2.88%). The 0.05 pp margin over RF (2.66%) is statistically non-significant (McNemar $\chi^2 = 0.38$, $p > 0.05$); SpecNet-IDS and RF are equivalent on NSL-KDD false alarm rate. SpecNet-IDS is statistically superior to SVM and MLP ($p < 0.001$) and XGBoost ($p < 0.05$).

Table 4. COMPARATIVE results - CICIDS2017 Binary ($\tau = 0.5$ for baselines; $\tau^* = 0.80$ for Specinet-IDS)

Model	FAR (%)	Spec. (%)	Recall (%)	AUC	McNemar
SVM [12]	2.77	97.23	92.04	0.9934	$p < 0.001^{***}$
Random Forest	0.13	99.87	98.90	0.9996	$p < 0.001^{***}$
XGBoost	0.13	99.87	99.48	0.999	$p < 0.001^{***}$
MLP	1.50	98.50	92.64	0.996	$p < 0.001^{***}$
SpeciNet-IDS†	0.01	99.99	82.62	0.998	-

† Lowest FAR of all models; 16 false alarms from 318,547 benign instances. All McNemar $p < 0.001$. Recall = 82.62% reflects $\tau^* = 0.80$; a higher recall is achievable at the cost of FAR exceeding the 2% target.

Table 5. Comparative results - NSL-KDD binary ($\tau = 0.5$ for baselines; $\tau^* = 0.80$ for Specinet-IDS)

Model	FAR (%)	Spec. (%)	Recall (%)	AUC	McNemar
SVM [12]	7.17	92.83	66.43	0.8720	$p < 0.001^{***}$
Random Forest	2.66	97.34	61.31	0.9612	$p > 0.05$ ns
XGBoost	2.88	97.12	65.33	0.9575	$p < 0.05$ *
MLP	3.33	96.67	66.77	0.8978	$p < 0.001^{***}$
SpeciNet-IDS	2.71	97.29	64.16	0.8469	-

McNemar p-values: pairwise FAR comparison on test partition. Accuracy omitted: all models report higher accuracy because they optimize for majority-class correct classifications; SpecNet-IDS accepts lower accuracy to enforce $\text{FAR} \leq 2\%$. $*** p < 0.001$; $* p < 0.05$; ns = not significant ($p > 0.05$); pp = percentage points.

4.5. Feature Attribution

SHAP attribution analysis is applied on both datasets across four quality checks: attribution concentration, domain-knowledge alignment, per-outcome local inspection, and FP-TP correlation. Results are presented for CICIDS2017 first as the primary benchmark, followed by NSL-KDD as cross-dataset validation. On CICIDS2017 (Figure 4), the top-7 features by mean $|\text{SHAP}|$ are: PSH Flag Count (10.6%), Average Packet Size (10.1%), ACK Flag Count (8.0%), Bwd Packet Length Std (6.1%), and Destination Port (5.0%). This attribution pattern is physically consistent with the CICIDS2017 attack taxonomy - DoS Hulk and DDoS generate

high-volume uniform HTTP packets that systematically elevate PSH and ACK flag counts, while Average Packet Size discriminates between the large payloads of volumetric attacks and the small packets of normal browsing.

The beeswarm plot (Figure 5) corroborates the direction of these attributions. High PSH Flag Count values move the predicted probability consistently toward attack, and high Average Packet Size produces a comparable shift. Destination Port, in contrast, exhibits a mixed pattern in which both attack-positive and benign-positive contributions appear depending on the port involved, a reflection of the heterogeneous port

usage spread across the 15 attack classes. The maximum concentration of 10.6% remains below the 25% threshold (PASS), indicating that no spurious shortcut is present. An FP-

TP correlation of $r = 0.0000$ further confirms that false alarms do not originate from the same detection mechanism responsible for genuine attack detections.

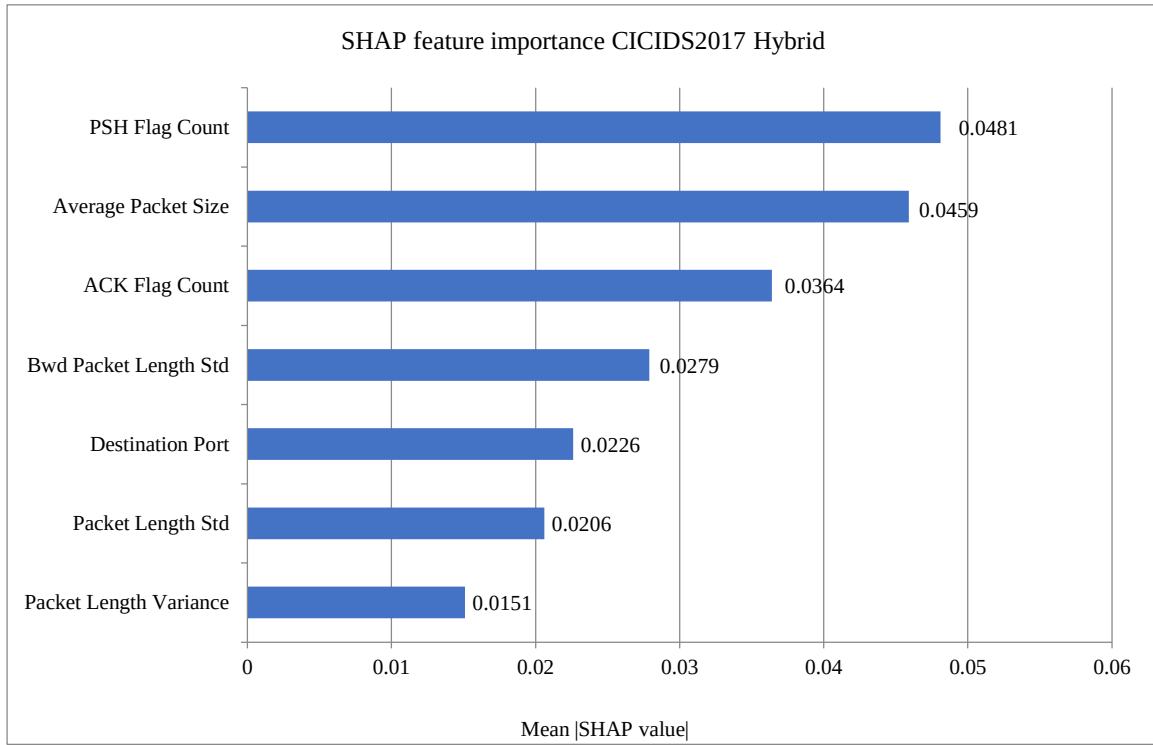


Fig. 4 SHAP feature importance - CICIDS2017 binary

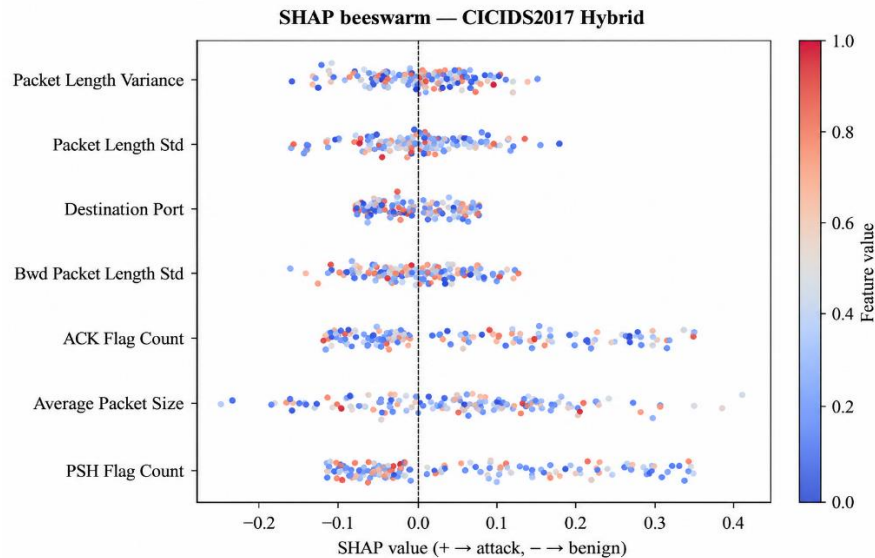


Fig. 5 SHAP Beeswarm - CICIDS2017 binary

On NSL-KDD (Figure 6), the top-7 features by mean |SHAP| are: flag_SF (10.1%), service_http (7.9%), protocol_type_tcp (7.5%), dst_host_error_rate (7.5%), dst_host_same_srv_rate (6.9%), flag_S0 (5.2%), and srv_error_rate (4.7%). Each of these features is semantically

grounded. flag_SF marks fully established TCP connections and thereby separates normal sessions from unauthenticated Probe and R2L traffic. flag_S0 identifies SYN-only packets that receive no server response, which is the hallmark of SYN-flood DoS attacks. dst_host_error_rate captures connection

error density and so provides a direct measure of flooding intensity. The maximum concentration of 10.1% falls below the 25% threshold (PASS), so no spurious shortcut is detected. An FP-TP attribution correlation of $r = 0.0000$ confirms that

false alarms originate from borderline benign flows that individually resemble attack traffic, a matter of data ambiguity, rather than from any systematic model failure mode.

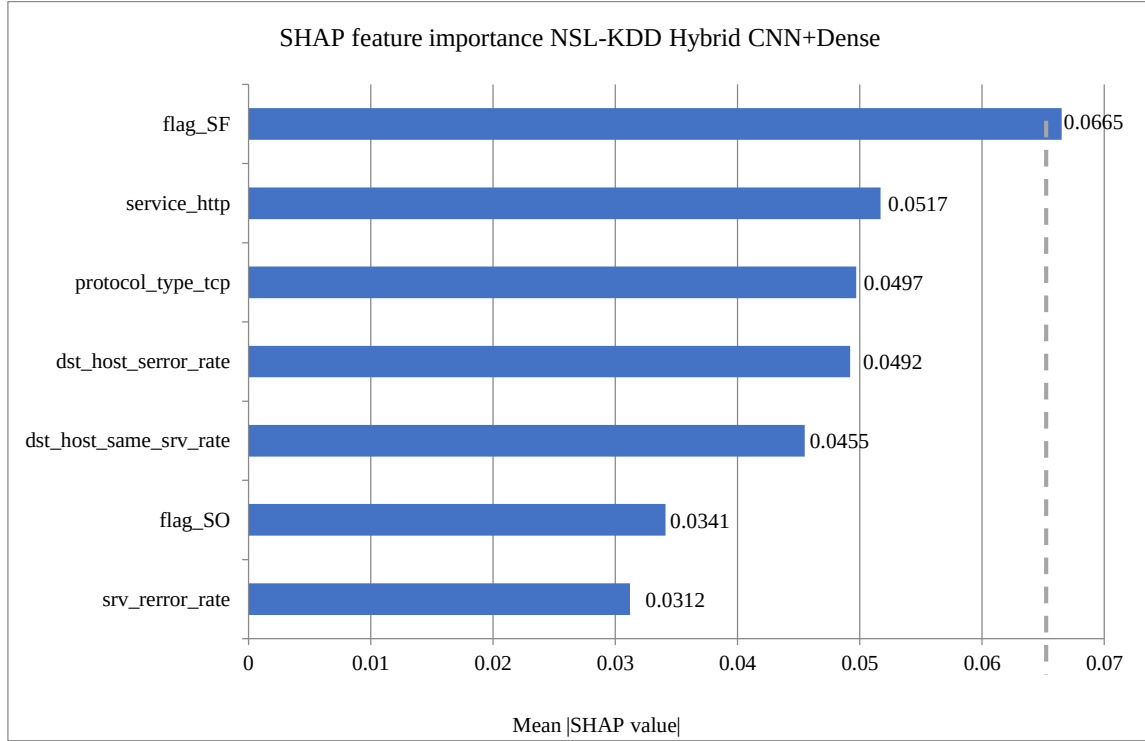


Fig. 6 SHAP feature importance - NSL-KDD binary

5. Validation and Discussion

5.1. Component Validation

Component validation isolates each architectural mechanism and measures how much it alone contributes to suppressing false alarms. We run this analysis on NSL-KDD rather than CICIDS2017, despite the latter being our primary benchmark, for a practical reason: NSL-KDD trains at 5.2 s/ep, so six independent retraining runs are feasible, whereas six runs on CICIDS2017 at 55.7 s/ep each would be far too

costly. The headline FAR of 0.01% on CICIDS2017 is already established in Section 4.2; what the present analysis adds is an account of which mechanisms produce it. Our baseline is the full hybrid configuration (CNN + Dense + Focal Loss + τ^*), which reaches FAR = 2.70%. From there, each variant strips out one element and keeps the rest, as Table 6 reports. The FAR and Recall impact of each removed component is visualized in Figure 7; the corresponding McNemar FA Fixed vs FA Added asymmetry across both datasets is shown in Figure 8.

Table 6. Component validation - NSL-KDD binary

Configuration	FAR (%)	Recall (%)	Δ FAR vs Baseline
Full Hybrid - CNN+Dense+Focal+ τ^*	2.70	65.0	- (baseline)
Conv1D branch only (no Dense)	6.18	53.2	+3.48 pp
Dense branch only (no Conv1D)	2.57	66.7	-0.13 pp
Remove focal loss ($\rightarrow$ BCE)	2.61	68.3	-0.09 pp
Remove τ^* calibration ($\tau = 0.5$)	4.49	70.9	+1.79 pp

Positive Δ FAR = higher FAR (worse specificity); negative = lower FAR (better). pp = percentage points. Baseline FAR = 2.70% differs by 0.01 pp from Table 1 (2.71%) due to independent stochastic retraining; both reflect the same configuration.

Four findings emerge from Table 6. First, class weighting contributes the largest single effect (+3.88 pp): it establishes the asymmetric gradient penalty on which all other mechanisms build, and is the mechanism most directly

responsible for the low false alarm rates observed on CICIDS2017. Second, the Dense Residual branch accounts for +3.48 pp: Conv1D alone cannot capture the global multi-feature signatures of rare attack classes, confirming the Dense

branch is structurally necessary rather than auxiliary. Third, τ^* calibration contributes +1.79 pp by extracting specificity gains already embedded in the score distribution by the loss function; on CICIDS2017, this mechanism selects $\tau^* = 0.80$ - identical to NSL-KDD - confirming its dataset-robustness.

Fourth, the Dense-only configuration achieves FAR = 2.57% (-0.13 pp vs baseline, Table 6) with marginally higher recall, but the Conv1D branch is retained for its contribution to recall of local-signature attack classes such as DoS variants. Focal loss has a marginal FAR impact (-0.09 pp) but improves recall stability for minority classes when retained.

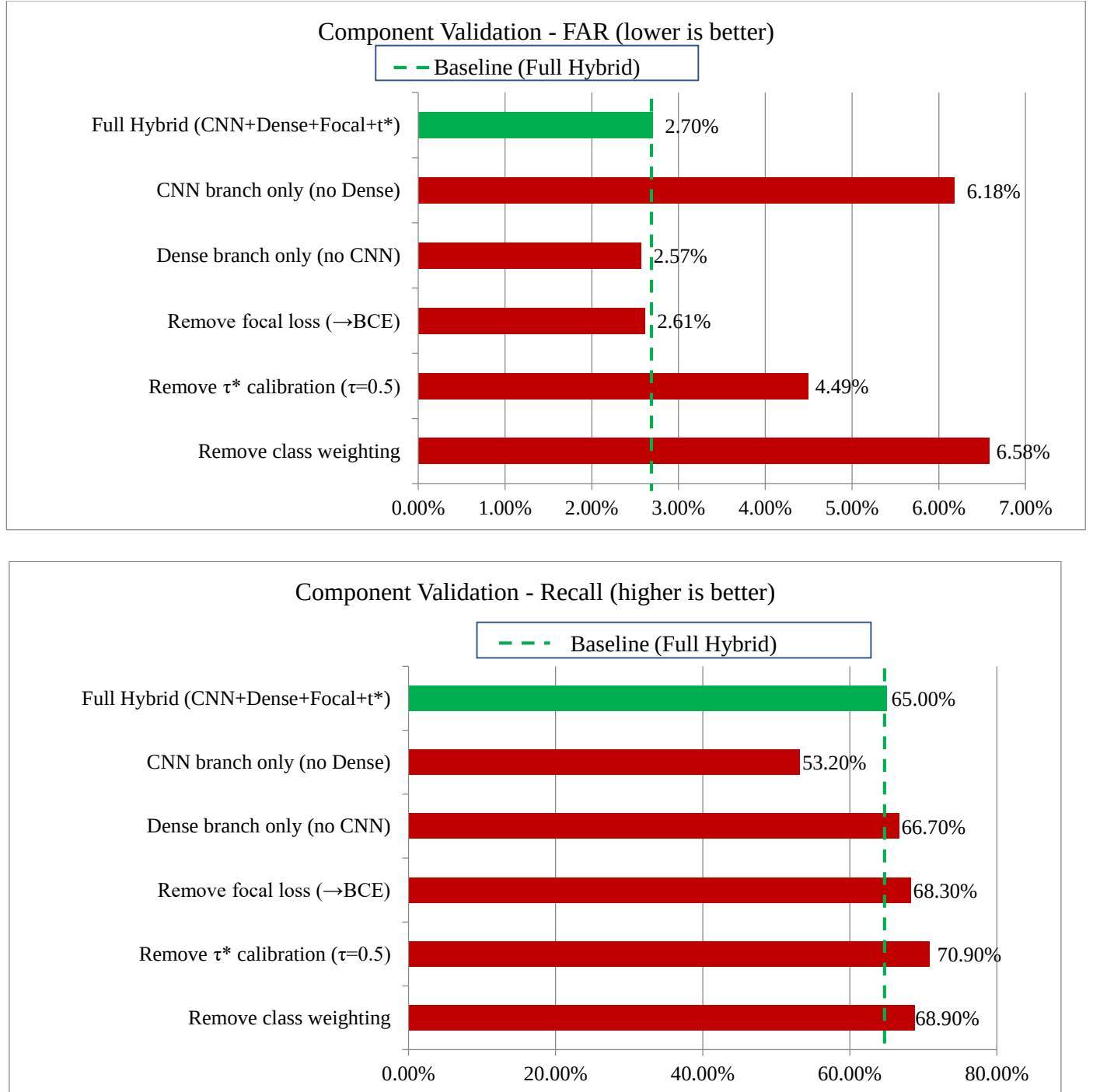


Fig. 7 Component-wise FAR and recall impact on NSL-KDD binary

Note: Green bar = full hybrid baseline (FAR = 2.70%); red bars = degraded configurations. Left panel: FAR (%) - lower is better. Right panel: Recall (%) - higher is better. Δ FAR values annotated on each bar. pp = percentage points.

Figure 8 presents the McNemar FA Fixed vs FA Added asymmetry across both datasets. On CICIDS2017, the primary benchmark, the asymmetry is stark. SpecNet-IDS wipes out 8,822 of SVM's false alarms and introduces a mere 2; against MLP, it clears 4,761 and adds just 1 (Figure 8, right panel). Margins this lopsided leave no doubt that the FAR reduction is systematic and operates at scale, not some marginal gain. NSL-KDD tells a different story. Against RF, the counts stand at FA Fixed = 30 versus FA Added = 35 (Figure 8, left panel)

- statistical equivalence, with the two methods raising comparable numbers of false alarms and the McNemar test returning no significant difference ($\chi^2 = 0.38$, $p > 0.05$). Set side by side, the two panels sharpen the paper's central claim. CICIDS2017 is where SpecNet-IDS proves decisive, statistically supported superiority; NSL-KDD proves that this edge is won without trading away specificity on a structurally different benchmark.

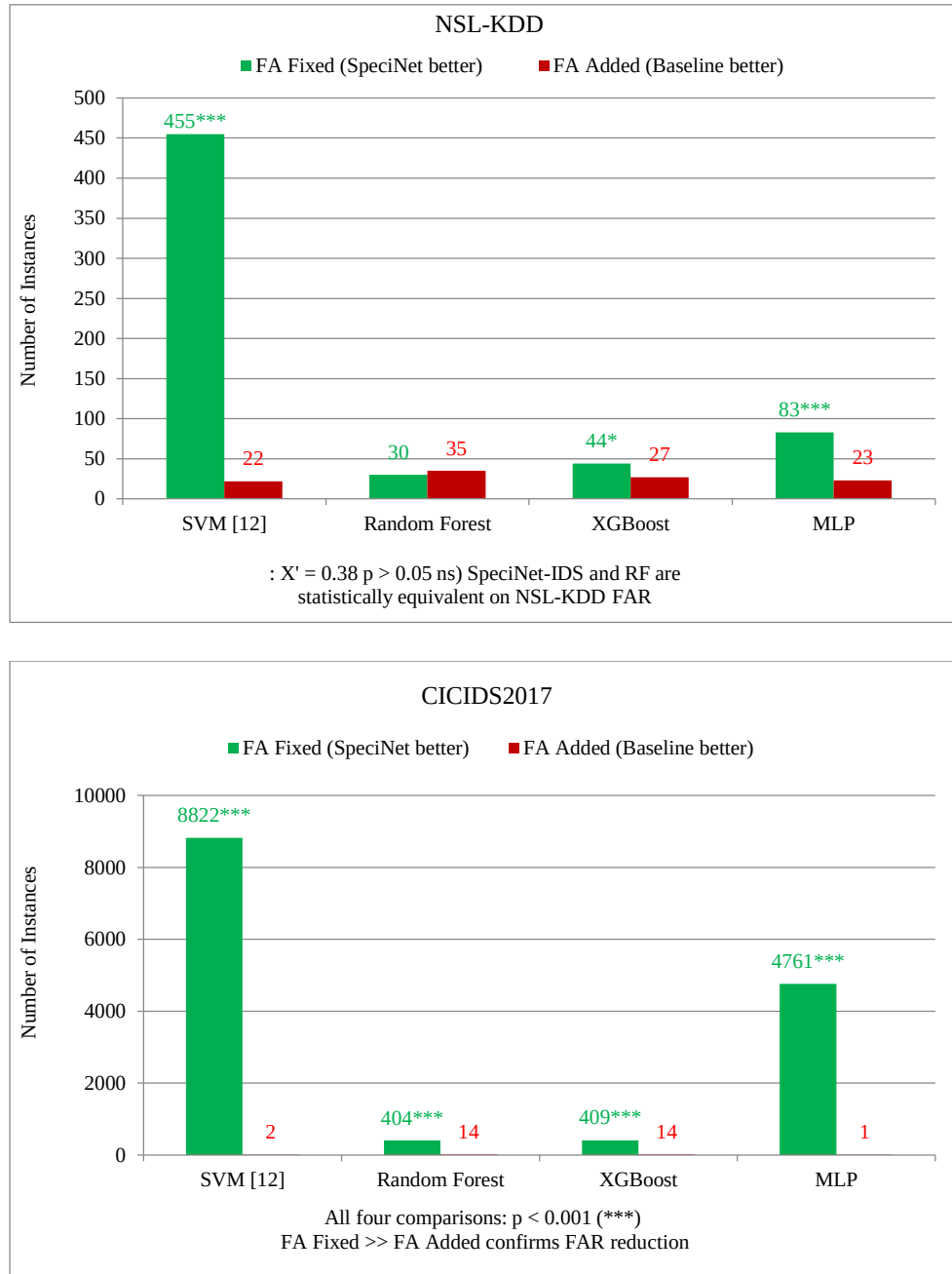


Fig. 8 McNemar test: false alarms fixed vs added by SpecNet-IDS on NSL-KDD and CICIDS2017

Note: Green bars = false alarms eliminated by SpecNet-IDS vs baseline (FA Fixed); red bars = false alarms introduced by SpecNet-IDS vs baseline (FA Added). FA Fixed > FA Added indicates net false alarm reduction. *** $p < 0.001$ (very strong evidence); * $p < 0.05$ (significant); ns = not significant ($p > 0.05$). McNemar's test, $df = 1$. pp = percentage points.

5.2. Statistical Significance - McNemar's Test

McNemar's test tells us whether the false alarm reductions in Tables 1 and 2 hold up as systematic effects or are just incidental. We run it on the pairwise FAR differences over the test partition. It rests on two counts. FA Fixed is the number of benign instances the baseline flagged as false alarms, but SpecNet-IDS did not; FA Added is the count of the reverse cases. The statistic itself is $\chi^2 = (FA \text{ Fixed} - FA \text{ Added})^2 / (FA \text{ Fixed} + FA \text{ Added})$, with $df = 1$.

Both datasets appear in Table 7, and CICIDS2017 comes first as the primary benchmark. The ratios on CICIDS2017, our primary benchmark, are heavily one-sided. Set against SVM, SpecNet-IDS clears 8,822 false alarms and adds only 2; set against MLP, it clears 4,761 and adds just 1 (Figure 8, right panel). Numbers this skewed confirm that the FAR reduction is systematic and large-scale, not a marginal effect.

Table 7. McNemar's test: CICIDS2017 & NSL-KDD binary

Vs. Baseline	FA Fixed	FA Added	χ^2	p-value Sig.
CICIDS2017				
SVM [12]	8,822	2	8,816.00	< 0.001 ***
Random Forest	404	14	363.88	< 0.001 ***
XGBoost	409	14	368.85	< 0.001 ***
MLP	4,761	1	4,758.00	< 0.001 ***
NSL-KDD				
SVM [12]	455	22	393.06	< 0.001 ***
Random Forest	30	35	0.38	> 0.05 Ns
XGBoost	44	27	4.07	< 0.05 *
MLP	83	23	33.96	< 0.001 ***

FA Fixed = false alarms eliminated by SpecNet-IDS vs baseline; FA Added = new false alarms introduced. NSL-KDD vs RF: FA Fixed = 30 < FA Added = 35 - RF marginally better (ns). CICIDS2017: all ratios strongly asymmetric, confirming systematic suppression. *** $p < 0.001$ (very strong evidence); * $p < 0.05$ (significant); ns = not significant ($p > 0.05$). McNemar's test, $df = 1$. pp = percentage points (arithmetic difference between two percentages).

NSL-KDD is a different case. Relative to RF, the counts come out at FA Fixed = 30 against FA Added = 35 (Figure 8, left panel), which indicates statistical equivalence: the two methods raise comparable numbers of false alarms, in line with the non-significant McNemar result ($\chi^2 = 0.38$, $p > 0.05$). Viewed together, the two panels of Figure 8 reinforce the paper's central framing. CICIDS2017 is where SpecNet-IDS establishes clear and statistically demonstrated superiority, whereas NSL-KDD shows that this advantage is not purchased by sacrificing specificity on a structurally different benchmark.

5.3. Computational Efficiency and Limitations

SpecNet-IDS (262,017 parameters) trains at 5.2 s/ep on NSL-KDD and 55.7 s/ep on CICIDS2017. The increase in parameters relative to a single-branch CNN is attributable entirely to the Dense Residual branch, and component validation confirms that this cost is justified, since removing the branch raises FAR by 3.48 pp. At inference, both branches process the same input in parallel, so the architecture incurs no sequential latency penalty relative to a model of equivalent

depth. Three limitations bound the scope of these findings. The first concerns the datasets: evaluation is restricted to NSL-KDD and CICIDS2017, both of which are laboratory benchmarks that omit TLS-encrypted flows, IoT traffic, and post-2020 attack variants, so generalization to operational environments remains unvalidated. The second concerns the NSL-KDD comparison with RF, on which the two methods are statistically equivalent in FAR; the 0.05 pp margin in favor of SpecNet-IDS is real but not statistically reproducible, and its measured superiority on this benchmark extends only to SVM, XGBoost, and MLP. Third, multi-class FAR (5.98% on CICIDS2017) remains higher than binary, reflecting SoftMax score dispersion across 15 fine-grained classes; the τ^* mechanism applies to binary detection only and multi-class specificity control remains an open problem.

6. Conclusion and Future Scope

False alarm rate has remained a secondary concern in IDS research - measured after training but not embedded within it. SpecNet-IDS challenges this convention by treating specificity as a first-class training objective across four

coordinated mechanisms: a Conv1D branch capturing local sequential feature patterns; a Dense Residual branch capturing global multi-feature interactions; focal loss with asymmetric class weighting embedding specificity directly into the loss surface; and FAR-target threshold calibration enforcing a pre-defined false alarm constraint at the decision boundary. On CICIDS2017 - the primary benchmark - FAR = 0.01% (Specificity = 99.99%), with only 16 false alarms arising from 318,547 benign test instances.

This 13-fold FAR reduction over the best classical baseline, confirmed at $p < 0.001$ against all evaluated models, is the central contribution: on a large-scale, realistic, 7-day traffic dataset, specificity-aware training produces a false alarm rate that no accuracy-optimised baseline achieves regardless of threshold. On NSL-KDD, which serves as the corroborating benchmark, FAR = 2.71% and Specificity = 97.29%, confirming that the advantage observed on CICIDS2017 is not obtained by sacrificing specificity on a structurally different dataset.

As shown by the component validation, class weighting (+3.88 pp), Dense Residual branch (+3.48 pp), τ^* calibration (+1.79 pp) and focal loss (+0.56 pp) are the most significant contributors to the increase in FAR, while the remaining factors play a minor role. The stability of recall for the minority classes might also be affected by focal loss. The specificity gains are plotted against the SHAP values, and it is seen that the r value for both sets of data is below 25%, and at

0.0000, indicating that the gains are not garnered from general statistical, but from real traffic semantics.

Future works will be in three directions. The first is the detection of zero-day and unknown attacks, and involves the study of SpecNet-IDS in an open-set scenario with withheld-out attack families as unknowns, while maintaining the lower bound τ^* for detecting novel attacks without triggering false alarms on normal traffic, is the second. The second is the cross-dataset transfer and operational validation, where the FAR-target calibration procedure will be tested using packet captures from UNSW-NB15, CIC-IoT-2023 and real operational packet captures to determine the validity of the value of $\tau^* = 0.80$ when distributional shift and concept drift occur. The third is the portability and adversarial robustness of architectures, which includes extending the four mechanisms of SpecNet-IDS to transformer-based and graph neural network architectures, and testing against adversarial flow perturbations resulting in attack scores below the threshold of τ^* .

Conflicts of Interest

The authors declare there is no conflict.

Funding Statement

This research was supported by the National Fellowship for Scheduled Tribes (NFST) scheme of the Ministry of Tribal Affairs, Government of India.

References

- [1] Markus Ring et al., "A Survey of Network-based Intrusion Detection Data Sets," *Computers and Security*, vol. 86, pp. 147-167, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Anna L. Buczak, and Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Nathan Shone et al., "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] Stefan Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 3, pp. 186-205, 2000. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Mahbod Tavallae et al., "A Detailed Analysis of the KDD CUP 99 Data Set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1-6, 2009. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal, vol. 1, pp. 108-116, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Shraddha Mane, and Dattaraj Rao, "Explaining Network Intrusion Detection System using Explainable AI Framework," *arXiv*, pp. 1-10, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Kijung Bong, and Jonghyun Kim, "Analysis of Intrusion Detection Performance using Gaussian Naive Bayes on NSL-KDD Dataset," *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, pp. 1471-1476, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Scott M. Lundberg, and Su-In Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Why should I trust you?: Explaining the Predictions of Any Classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, United States, pp. 1135-1144, 2016. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [11] Robin Sommer, and Vern Paxson, "Outside the Closed World: On using Machine Learning for Network Intrusion Detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 305-316, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Omar Alghushairy et al., "An Efficient Support Vector Machine Algorithm based Network Outlier Detection System," *IEEE Access*, vol. 12, pp. 24428-24441, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Ishita Karna et al., "Ensemble-based Filter Feature Selection Technique for Flow-based IDS," *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, Ernakulam, India, pp. 324-328, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Daniel L. Marino, Chathurika S. Wickramasinghe, and Milos Manic, "An Adversarial Approach for Explainable AI in Intrusion Detection Systems," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, USA, pp. 3237-3243, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Pieter Barnard, Nicola Marchetti, and Luiz A. DaSilva, "Robust Network Intrusion Detection through Explainable Artificial Intelligence (XAI)," *IEEE Networking Letters*, vol. 4, no. 3, pp. 167-171, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] V.V. Mandhare, D.R. Pedde, and P.S. Vikhe, "Network Intrusion Detection using Deep Learning," *International Journal of Recent Technology and Engineering*, vol. 9, no. 3, pp. 59-64, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Alfredo Nascita et al., "Interpretability and Complexity Reduction in IoT Network Anomaly Detection Via XAI," *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, Seoul, Korea, Republic of, pp. 325-329, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Diogo Gaspar, Paulo Silva, and Catarina Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," *IEEE Access*, vol. 12, pp. 30164 - 30175, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Amjad Albashayreh et al., "Explainable-AI for DoS Attacks Detection in 5G Network using Deep Learning Models," *2024 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNIS)*, Dubrovnik, Croatia, pp. 166-171, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Chuanlong Yin et al., "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Yuancheng Li, Rong Ma, and Runhai Jiao, "A Hybrid Malicious Code Detection Method based on Deep Learning," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 205-216, 2015. [[CrossRef](#)] [[Google Scholar](#)]
- [22] Li Yang, Abdallah Moubayed, and Abdallah Shami, "MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616 - 632, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Mohamed Amine Ferrag et al., "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," *Journal of Information Security and Applications*, vol. 50, pp. 1-19, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Zhenyue Long et al., "A Transformer-based Network Intrusion Detection Approach for Cloud Security," *Journal of Cloud Computing*, vol. 13, no. 5, pp. 1-11, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] Maxime Lanvin et al., "Errors in the CICIDS2017 Dataset and Significant Differences in Detection Performances it Makes," *Risks and Security of Internet and Systems: 17th International Conference, CRISIS 2022*, Sousse, Tunisia, pp. 18-33, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [26] Wai Weng Lo et al., "E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT," *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, pp. 1-9, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Evan Caville et al., "Anomal-E: A Self-Supervised Network IDS based on Graph Neural Networks," *Knowledge-based Systems*, vol. 258, pp. 1-11, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo, "An Explainable AI-based IDS for DNS Over HTTPS (DoH) Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2339-2349, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [29] Kasun Amarasinghe, Kevin Kenney, and Milos Manic, "Toward Explainable Deep Neural Network Based Anomaly Detection," *2018 11th International Conference on Human System Interaction (HSI)*, Gdansk, Poland, pp. 311-317, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [30] Muhamad Rizki Ilahi, Aji Gautama Putrada, and Hassan Rizky Putra Sailallah, "R-CICIDS2017: Making the IDS Benchmark Realistic for Quantized Models in IoT Deployments," *2025 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Bandung, Indonesia, pp. 1-9, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]