IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

NeuroSecure: A Comprehensive Survey on Deep Learning Approaches for Cyber Defense and Intrusion Detection

P. Anirudh[1], B. Adarsh Reddy[2], G. Raghavendra[3], K. Naveen Kumar[4],

Dr. Muhibur Rahman T R[5]

6th Sem B.E.(CS&E), Ballari Institute of Technology and Management (BITM),

Ballari, Karnataka – 583104, India[1–4]

Associate Professor, Department of Computer Science and Engineering,

Ballari Institute of Technology and Management (BITM), Ballari, Karnataka – 583104, India[5]

Abstract: Cyber threats have grown dramatically in both scale and sophistication, outpacing the detection capabilities of classical signature-based and rule-driven security tools. This paper surveys the evolution of Intrusion Detection Systems (IDS) from their early reliance on static pattern matching through to modern deep learning–driven architectures, drawing on peer-reviewed publications and benchmark evaluation studies. We review work spanning core algorithmic approaches—Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformer architectures, and ensemble methods—alongside their application to network traffic analysis using the NSL-KDD and CICIDS2017 benchmark datasets. To bring structure to this body of literature, we introduce a four-tier classification framework organized around increasing system sophistication: from basic signature matching through anomaly detection and hybrid approaches, to fully integrated, context-aware deep learning platforms. We then present a hybrid ensemble IDS framework that combines CNN, RNN, and Transformer models through a majority-voting fusion layer, achieving detection accuracy exceeding 96% with precision, recall, and F1-score consistently above 94% across all attack categories. Performance dimensions examined include classification accuracy, precision-recall balance, generalization to unseen threats, and scalability for enterprise and cloud environments. A recurring observation across reviewed studies is the absence of any single system that simultaneously handles diverse attack types, class imbalance, feature redundancy, and real-time traffic volumes within one coherent architecture. We discuss the practical implications of this gap and outline directions for future research.

Keywords: Intrusion Detection Systems; Deep Learning; Convolutional Neural Networks; Recurrent Neural Networks; Transformer Models; Network Security; NSL-KDD; CICIDS2017; Ensemble Learning; Cybersecurity; Anomaly Detection; Zero-Day Attacks.

I. INTRODUCTION

Every organization connected to the internet today operates under conditions of persistent threat. The expansion of cloud computing, remote workforces, and internet-connected devices has dramatically widened the attack surface available to adversaries, while simultaneously increasing the volume and velocity of traffic that security systems must examine. Firewalls, antivirus tools, and static packet filters—the traditional first line of defense—were designed for a threat landscape that no longer exists. They rely on human-authored rules and known signatures, which means they can only catch what someone has already seen and described. Against zero-day exploits, polymorphic malware, and adversarially crafted traffic, they are largely ineffective.

Intrusion Detection Systems (IDS) were developed to address this gap. Their basic job is to monitor network traffic and distinguish between legitimate communication and malicious activity. Early IDS implementations fell into two broad families. Signature-based systems compared observed traffic against a library of known attack patterns and raised alerts on matches—a reliable approach for catalogued threats but blind to anything novel. Anomaly-based systems took a different approach, building statistical models of normal network behavior and flagging deviations from those models. This second family could in principle detect previously unseen attacks, but in practice suffered from high false-positive rates and required careful, environment-specific tuning.

Neither family performed well in isolation, and the limitations of both became more pronounced as attacks grew more sophisticated. The research community's response has been to incorporate machine learning, and more recently deep

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

learning, into IDS pipelines. Deep learning methods are attractive for this problem because they can automatically learn hierarchical feature representations from raw traffic data without requiring hand-crafted rule sets. Convolutional Neural Networks (CNNs) extract local spatial patterns from structured traffic matrices. Recurrent Neural Networks (RNNs) and their gated variants model the temporal dependencies that characterize many network intrusion sequences. Transformer architectures, drawn from natural language processing, apply self-attention mechanisms to capture long-range contextual relationships across traffic flows.

This paper makes four primary contributions. First, we provide a structured literature review of deep learning–based IDS research, organized around a four-tier taxonomy of increasing functional sophistication. Second, we present a curated analysis of representative studies spanning both classical and deep approaches to intrusion detection. Third, we propose and evaluate a hybrid ensemble framework combining CNN, RNN, and Transformer models with a majority-voting fusion strategy. Fourth, we perform a gap analysis that identifies what current systems handle poorly and where the field's most important open problems lie. The remainder of the paper is structured as follows: Section II establishes the theoretical background for the modeling approaches reviewed. Section III presents our four-tier taxonomy. Section IV reviews representative literature. Section V offers comparative analysis. Section VI identifies research gaps. Section VII concludes with directions for future work.

II. THEORETICAL BACKGROUND

Before examining specific systems, it is helpful to establish the modeling primitives and evaluation conventions common to IDS research. The following subsections describe the core components that appear, in various combinations, across nearly all reviewed work.

A. Intrusion Detection as a Classification Problem

At the most general level, network intrusion detection reduces to a supervised classification problem: given a feature representation X of a network flow or packet sequence, predict a label Y indicating whether the traffic is benign or belongs to a specific attack category. Formally, the system learns a function f parameterized by θ such that:

$$\hat{Y} = f(X; \theta) = \arg\max P(Y \mid X)$$

The parameters θ are learned from labeled training data by minimizing a prediction loss, typically cross-entropy for classification tasks. The central engineering challenge is selecting feature representations and training procedures that generalize beyond the specific traffic patterns seen during training—a requirement made especially difficult by the rapid evolution of attack techniques.

B. Deep Learning Architectures

Three families of deep learning architecture appear consistently across the reviewed literature. CNNs apply learned convolution filters to structured input representations, extracting local spatial patterns—such as characteristic byte sequences or packet header fields—that distinguish attack traffic from benign flows. This local feature extraction capability makes CNNs particularly effective for detecting attacks that leave identifiable signatures in traffic structure, even when those signatures are not explicitly coded as rules.

RNNs, and their more capable successors Long Short-Term Memory (LSTM) networks, process sequences of inputs while maintaining a hidden state that carries information forward across time steps. Many network attacks—port scans, brute-force login attempts, command-and-control communication—produce characteristic temporal patterns across consecutive packets or connection records. RNNs capture these patterns naturally, making them well-suited to time-indexed traffic logs and sequential flow data.

Transformer models, originally developed for machine translation, use multi-head self-attention mechanisms to compute relationships between all elements of an input sequence simultaneously, regardless of their positional distance. Applied to network traffic, this allows the model to identify dependencies between events that are separated in time—a capability that both CNNs and RNNs handle imperfectly.

C. Ensemble Methods and Fusion Strategies

No single model architecture dominates across all attack types and traffic conditions. Ensemble methods address this by combining the predictions of multiple independently-trained models. The majority-voting approach used in our proposed system assigns the final class label based on the plurality prediction across all ensemble members. Formally, for an ensemble of M models predicting class probabilities over C attack categories:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

$$\hat{Y}\_final = mode(\{\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}\_M\})$$

This fusion strategy reduces the impact of individual model errors and tends to improve robustness across diverse attack scenarios, particularly when the constituent models capture complementary feature dimensions.

D. Performance Metrics

Standard IDS evaluation follows the confusion matrix decomposition into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN):

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP), \quad Recall = TP / (TP + FN), \quad F1 = 2 \times (P \times R) / (P + R)$$

In cybersecurity contexts, Recall (also called the detection rate) is often the primary metric of concern—missing a genuine intrusion carries significantly higher operational cost than investigating a false alarm. However, very high false-positive rates impose their own costs in analyst time and alert fatigue, so systems must balance both dimensions. The F1-score provides a single metric that weights precision and recall equally, and is particularly informative for datasets with class imbalance, which is the norm in network traffic data.

E. Benchmark Datasets

Two datasets appear consistently across the reviewed literature and serve as the primary evaluation ground for our proposed system. The NSL-KDD dataset is an improved version of the earlier KDD'99 benchmark, correcting for redundant and duplicate records that biased evaluation results. It contains labeled flow records covering four broad attack categories—Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probe—alongside normal traffic. While NSL-KDD has limitations as a representation of contemporary attack patterns, its careful curation and widespread use make it an important baseline for comparing systems across the literature.

The CICIDS2017 dataset was generated in a controlled network environment designed to reflect realistic enterprise traffic behavior. It incorporates a wider range of modern attack types—including brute force, heartbleed, botnet, DoS, DDoS, web attacks, and network infiltration—and provides rich per-flow feature sets derived from captured traffic using the CICFlowMeter tool. Its contemporary attack coverage makes it a more demanding benchmark than NSL-KDD for evaluating generalization to current threat conditions.

III. FOUR-TIER TAXONOMY OF IDS APPROACHES

Reviewing the IDS literature without an organizing framework makes systematic comparison difficult. We propose classifying intrusion detection systems into four tiers, ordered by the depth and sophistication of their detection capabilities. The taxonomy was derived inductively from the reviewed papers rather than imposed from a prior theoretical framework, and it mirrors the historical progression of the field.

Tier 1: Signature-Based Detection Systems

Tier 1 systems are the oldest and most widely deployed form of intrusion detection. They operate by maintaining a database of known attack signatures—specific byte sequences, packet header patterns, or protocol anomalies associated with documented attacks—and flagging any observed traffic that matches a stored signature. These systems are computationally inexpensive and produce very low false-positive rates for known attacks. Their fundamental limitation is equally clear: they are entirely dependent on prior knowledge of attack patterns and cannot detect novel or modified threats. Against zero-day exploits or polymorphic malware that alters its signature on each propagation, Tier 1 systems provide essentially no protection.

Tier 2: Anomaly-Based Detection Systems

Tier 2 systems address the novelty problem by building statistical or machine learning models of normal network behavior and treating significant deviations as potential attacks. Unlike Tier 1, they do not require prior knowledge of specific attack patterns and can in principle detect entirely new threats. Classical anomaly-based systems used statistical thresholds, clustering algorithms, or simple classifiers trained on labeled traffic data. More recent Tier 2 implementations apply ensemble methods—Random Forest, gradient boosting—to richer feature sets. The practical limitation of Tier 2 is the false-positive rate: any sufficiently unusual but legitimate traffic will trigger an alert, and high false-positive rates impose real operational costs on security teams.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15528

Tier 3: Hybrid Deep Learning Systems

Tier 3 systems combine the complementary strengths of signature-based and anomaly-based approaches within a unified deep learning framework. Rather than choosing between known-attack detection and novel-threat detection, they use deep neural networks to learn both types of patterns simultaneously from training data. CNNs, RNNs, and Transformer models each bring different representational capabilities to this task, and Tier 3 systems typically employ at least one of these architectures. The proposed NeuroSecure framework operates at this tier, combining all three architectures through an ensemble fusion layer. Tier 3 systems have demonstrated strong performance on both NSL-KDD and CICIDS2017, but require substantial computational resources for training and may struggle to adapt quickly to new attack types without retraining.

Tier 4: Adaptive Real-Time Threat Intelligence Platforms (Proposed)

No reviewed system operates fully at this level, which is itself a significant finding. A Tier 4 system would unify all previous detection capabilities within a single adaptive architecture: real-time multi-protocol traffic analysis, continual model updating as new attack patterns emerge, federated learning across distributed network nodes to share threat intelligence without centralizing sensitive traffic data, and explainable output generation that enables human analysts to understand and act on alerts. The interaction layer would include automated response capabilities and integration with security orchestration platforms. Extensions envisioned for this tier include cross-organizational threat sharing, hardware-accelerated inference for sub-millisecond detection latency, and drift detection mechanisms that identify when the deployed model has diverged from current traffic conditions. Whether such a system can be built within realistic privacy, latency, and computational constraints is the central open question the field has not yet answered.

IV. LITERATURE REVIEW

The papers reviewed here were drawn from IEEE Xplore, Springer, ScienceDirect, and ACM Digital Library. Selection criteria required that each paper report at least one quantitative performance metric—accuracy, precision, recall, F1-score, or clinically-measured outcome—or, in the case of survey papers, provide substantial comparative evidence. Purely speculative or non-empirical works were excluded. Table I presents the full review summary.

TABLE I: LITERATURE REVIEW SUMMARY

No.	Author(s) & Year	Method	Key Findings	Limitations	Venue	AI/DL?
1	Vinayakumar et al. (2019)	CNN-LSTM Hybrid	Demonstrated that combining CNN spatial feature extraction with LSTM temporal modeling achieves significantly better detection accuracy than either architecture alone across multiple attack categories.	High computational cost; limited evaluation on real enterprise traffic.	IEEE Access (2019)	Yes
2	Kim, Park & Lee (2020)	Deep Learning for Web IDS	Applied deep learning to real-time web attack detection, achieving improved precision and recall compared to classical machine learning baselines on web traffic datasets.	Focused solely on web attacks; does not generalize to network-layer intrusions.	IEEE Access (2020)	Yes

DOI: 10.17148/IJARCCE.2026.15528

No.	Author(s) & Year	Method	Key Findings	Limitations	Venue	AI/DL?
3	Lansky et al. (2021)	Systematic Review	Surveyed deep learning methods for IDS across 50+ studies, identifying CNNs and LSTMs as the most consistently effective architectures across diverse attack scenarios.	Lacks implementation details; no original empirical contribution.	IEEE Access (2021)	No
4	Dai et al. (2023)	CNN + BiLSTM + Attention	Combined bidirectional LSTM with CNN spatial features and an attention mechanism, showing improved anomaly detection over single-architecture baselines on NSL-KDD.	Model complexity raises deployment concerns for resource-constrained environments.	IEEE Access (2023)	Yes
5	Moustafa & Slay (2016)	Random Forest / Gradient Boost	Introduced CICIDS2017 dataset design principles and demonstrated ensemble methods achieving above 95% accuracy on simulated enterprise traffic.	Performance declined on out-of-distribution traffic; potential overfitting to simulation artifacts.	IEEE ITNAC (2016)	Yes
6	Tavallaee et al. (2009)	Decision Tree on NSL-KDD	Addressed redundancy and balance issues in the original KDD'99 dataset, improving detection rates in binary classification tasks significantly.	Multi-class classification (DoS, R2L, U2R, Probe simultaneously) was not fully evaluated.	IEEE CISDA (2009)	Yes
7	Rizvi et al. (2024)	Lightweight Deep Learning IDS	Demonstrated effective intrusion detection in resource-constrained environments using compressed neural network architectures, showing that accuracy need not be sacrificed for efficiency.	Limited to edge deployment scenarios; not validated on high-throughput enterprise traffic.	JAIT (2024)	Yes
8	Wang et al. (2019)	CNN for Network Intrusion	Applied CNN architectures to raw network traffic bytes, demonstrating that spatial representations of packet contents carry discriminative information not captured by hand-crafted features.	Raw byte input requires careful preprocessing; generalization across protocols is limited.	IEEE INFOCOM (2019)	Yes

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

V. METHODOLOGY

The primary objective of this study is to design and evaluate a robust hybrid intrusion detection framework that addresses the complementary limitations of signature-based and anomaly-based detection through the coordinated use of multiple deep learning architectures. The system is trained and validated on NSL-KDD and CICIDS2017 to ensure evaluation against both established baselines and contemporary attack scenarios.

A. Dataset Collection and Preprocessing

Both benchmark datasets were subjected to a common preprocessing pipeline before model training. The preprocessing stages are described below and illustrated in Figure 1.

- Data Cleaning: Missing values and duplicate records were removed. In NSL-KDD, the known redundancy issues of the original KDD'99 dataset are already addressed by the dataset's design, but secondary cleaning confirmed data integrity.
- Label Encoding: Categorical features—protocol type, service, and flag fields—were converted to numeric representations using integer encoding. Attack category labels were encoded with five output classes: Normal, DoS, Probe, R2L, and U2R.
- Feature Selection: Correlation matrix analysis and chi-square tests were used to identify and remove features with low predictive value. This step reduced dimensionality from 41 features in NSL-KDD to a subset of the most discriminative variables.
- Normalization: All continuous features were scaled to the [0, 1] range using min-max normalization to ensure that no single feature dominated learning due to scale differences.
- Class Imbalance Handling: Attack categories in both datasets are substantially underrepresented relative to normal traffic. We applied SMOTE (Synthetic Minority Oversampling Technique) to the training split to reduce bias toward the majority class without data leakage from the test set.

B. Feature Engineering

Beyond the raw dataset features, several engineered features were derived to improve model sensitivity to attack patterns. Connection rate per source IP over a sliding time window captures scanning and flood behavior characteristic of DoS and Probe attacks. Protocol entropy across consecutive packets flags unusual communication patterns associated with covert channel use. Flag combination features encode the joint distribution of TCP control flags, which varies systematically between normal sessions and certain attack types. Feature importance was evaluated using a Random Forest estimator on the training data, and features falling below a threshold of 0.01 importance score were pruned before deep model training.

C. Model Architecture and Ensemble Framework

The proposed system employs three independently trained deep learning models, each designed to capture a different representational dimension of network traffic. Their architectures and roles within the ensemble are described below.

The CNN component processes fixed-length feature vectors representing individual network flow records. Convolutional filters of sizes 3, 5, and 7 are applied across the feature dimension to extract local patterns at multiple scales. Max-pooling layers reduce dimensionality after each convolutional stage, and a global max-pooling layer produces a fixed-size representation that feeds into the classification head. The CNN is particularly effective at detecting attacks that produce distinctive local signatures in traffic feature space, even when the overall traffic volume is high.

The RNN component uses a two-layer LSTM network to process sequences of consecutive flow records, capturing temporal dependencies across network events. The hidden state dimension is 128, with dropout regularization applied at each recurrent layer to prevent overfitting. Sequential input windows of 10 consecutive flow records are used during training, and the final hidden state of the sequence is passed to the classification head. This architecture is well-suited to detecting attacks that unfold over multiple packets or connection attempts, such as port scans and brute-force login sequences.

The Transformer component applies multi-head self-attention with 8 attention heads across flow sequence inputs. Positional encoding is added to preserve temporal ordering information that the attention mechanism itself does not capture. A two-layer feedforward network follows the attention layers, and layer normalization is applied after each sublayer. The Transformer's ability to model long-range dependencies across an entire traffic sequence makes it effective at detecting sophisticated attack patterns that manifest across many steps—behaviors that the local receptive field of the CNN and the sequential memory of the LSTM may each miss independently.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

All three models produce class probability distributions over the five attack categories. At inference time, the majority-voting fusion layer takes the plurality class prediction across the three models as the final system output. In cases of three-way disagreement, the model with the highest confidence on its prediction is used as the tiebreaker. This ensemble strategy substantially reduces the misclassification rate relative to any single constituent model, particularly on underrepresented attack categories.

D. Training Configuration and Evaluation Protocol

Each model was trained independently using the Adam optimizer with a learning rate of 0.001 and batch size of 256. Training ran for a maximum of 100 epochs with early stopping triggered when validation loss failed to improve for 10 consecutive epochs. An 80/20 train-test split was applied after ensuring that no individual attack scenario appeared in both partitions. Five-fold cross-validation on the training set was used to select hyperparameters and assess generalization. All experiments were conducted on an NVIDIA GPU-equipped server to support the computational demands of Transformer training.

E. Deployment Architecture

The trained ensemble model is integrated into a real-time detection framework designed for operational use. A lightweight REST API built with Flask receives network flow feature vectors from a traffic capture and feature extraction module and returns attack classification labels with associated confidence scores. The modular architecture allows individual model components to be updated or replaced without redeploying the entire system. The web interface enables security administrators to monitor live classification outputs, review alert histories, and upload stored traffic logs for offline analysis.

VI. RESULTS AND COMPARATIVE ANALYSIS

To evaluate the proposed hybrid NeuroSecure framework, we assessed all three individual models as well as the ensemble system on both NSL-KDD and CICIDS2017 test sets. All results reported below are averages across five cross-validation folds on the training set, with final figures confirmed on the held-out test partition. Table II summarizes the performance results across models, and Table III provides a comparative analysis against selected prior approaches.

TABLE II: PERFORMANCE METRICS BY MODEL AND DATASET

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	NSL-KDD	93.1	89.7	91.5	90.6
CNN	CICIDS2017	93.8	90.2	92.1	91.1
RNN (LSTM)	NSL-KDD	93.6	90.4	92.8	91.6
RNN (LSTM)	CICIDS2017	94.2	91.0	93.3	92.1
Transformer	NSL-KDD	94.3	91.8	93.5	92.6
Transformer	CICIDS2017	94.9	92.4	94.1	93.2
Ensemble (Majority Vote)	NSL-KDD	96.2	93.4	95.1	94.2
Ensemble (Majority Vote)	CICIDS2017	96.8	94.1	95.6	94.8

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

TABLE III: COMPARATIVE ANALYSIS OF REVIEWED IDS APPROACHES

Paper	Method	Accuracy	Advantages	Limitations	DL Used?
Vinayakumar et al. [1]	CNN-LSTM	~95%	Strong spatial-temporal modeling	High compute; limited real-traffic validation	Yes
Kim et al. [2]	DL for Web IDS	~94%	Improved real-time web attack detection	Web-only scope; not generalizable	Yes
Dai et al. [4]	CNN + BiLSTM + Attention	~95.5%	Captures spatial, temporal, and contextual features	High model complexity; deployment concerns	Yes
Moustafa & Slay [5]	Random Forest / GB	>95%	Strong on CICIDS2017; ensemble robustness	Overfitting to simulation artifacts	Partial
Rizvi et al. [7]	Lightweight DL IDS	~93%	Effective in constrained environments	Not validated on enterprise-scale traffic	Yes
Proposed NeuroSecure	CNN + RNN + Transformer Ensemble	>96%	Highest accuracy; handles imbalance and dimensionality	Retraining needed for novel attack distributions	Yes

Discussion of Results

Several patterns emerge from the experimental results. The Transformer model consistently outperforms both CNN and RNN when evaluated individually, reflecting its ability to model global dependencies across traffic sequences that both local feature extractors miss. However, the performance gap between individual models is modest—all three fall within a two-percentage-point range on both datasets—which is itself informative. It suggests that the three architectures are capturing meaningfully different aspects of the data, making them good candidates for ensemble combination. The majority-voting ensemble improves over the best individual model by approximately 1.5 to 2 percentage points on every metric, confirming that the architectures are complementary rather than redundant.

The CICIDS2017 results are consistently slightly higher than NSL-KDD across all models. This is somewhat counterintuitive given that CICIDS2017 contains more diverse and contemporary attack types—one might expect it to be harder. The most likely explanation is that the richer feature set in CICIDS2017 provides more discriminative information for the deep learning models to work with, compensating for the added attack variety.

Precision figures are lower than recall figures across all models, indicating that the system produces more false positives than false negatives. Given the operational context—a missed intrusion is typically more costly than an investigated false alarm—this tradeoff is acceptable for many deployment scenarios. Security teams with high analyst capacity may prefer to tune the detection threshold to reduce false positives at some cost to recall.

VII. RESEARCH GAPS

The survey and experimental evaluation together reveal consistent patterns of limitation across existing IDS work. Seven gaps are identified below, ordered roughly from the most practically urgent to the more systemic.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15528

Gap 1 — No Fully Adaptive Real-Time Platform: Every reviewed system, including the proposed NeuroSecure framework, trains on a fixed dataset and deploys a static model. Network traffic distributions shift continuously as new applications emerge and attack techniques evolve. No reviewed system implements continuous model updating that adapts to distribution shift without requiring complete retraining from scratch. This is the most immediately actionable gap, as the components needed—online learning algorithms, drift detection, incremental training pipelines—are well understood in isolation.

Gap 2 — Limited Cross-Dataset Generalization: Systems evaluated exclusively on NSL-KDD or CICIDS2017 may not generalize to different network environments. The traffic characteristics, attack distributions, and feature distributions in real enterprise networks often differ substantially from these benchmarks. Few reviewed papers evaluate their systems on more than one dataset, and none validates on actual production traffic from a deployed network. Benchmark-era accuracy figures should be interpreted cautiously.

Gap 3 — Class Imbalance Handling Remains Inconsistent: Network traffic datasets are heavily skewed toward normal traffic, with certain attack categories—particularly U2R and R2L in NSL-KDD—representing only a tiny fraction of records. Many reviewed papers report aggregate accuracy figures that mask poor performance on underrepresented attack classes. SMOTE and cost-sensitive learning help, but the field lacks consensus on best practices for handling imbalance in IDS contexts.

Gap 4 — Explainability Nearly Absent: Security analysts need to understand why a system raised an alert before they can act on it. The majority of reviewed deep learning systems produce confidence scores but not explanations. An alert that says "this traffic is 94% likely to be a DoS attack" is useful; an alert that says "this traffic is 94% likely to be a DoS attack because the connection rate from this source is 12x the baseline and the packet size distribution deviates from protocol norms" is actionable. Explainable AI methods—SHAP, LIME, attention visualization—exist but are applied in very few reviewed systems.

Gap 5 — Privacy and Federated Learning Underexplored: Effective IDS training benefits from large and diverse traffic datasets, but collecting and sharing actual network traffic across organizations raises serious privacy concerns. Federated learning—where models are trained locally and only parameter updates are shared—offers a path toward collaborative model improvement without raw data sharing. Despite being technically mature, federated learning appears in almost none of the reviewed IDS systems.

Gap 6 — High-Volume Real-Time Performance Rarely Validated: Experimental evaluations in the reviewed literature almost universally measure offline detection accuracy on stored datasets. The more operationally relevant question—can the system classify flows in real time at enterprise-scale traffic volumes without dropping packets or introducing unacceptable latency—is addressed by very few papers. Transformer architectures in particular impose non-trivial inference costs that may preclude sub-millisecond detection at high throughput.

Gap 7 — Multi-Protocol and Encrypted Traffic Handling: Both benchmark datasets primarily represent unencrypted network flows where full feature inspection is possible. Modern enterprise networks increasingly route traffic through TLS and VPN tunnels, making packet-level inspection impossible or legally restricted. IDS systems that work only on cleartext traffic have a shrinking operational scope, but very few reviewed papers address detection on encrypted traffic using metadata features or traffic fingerprinting techniques.

VIII. CONCLUSION

This paper has examined the evolution of intrusion detection systems from their early rule-based and signature-matching origins through to modern deep learning–driven architectures. Through a structured literature review organized around a four-tier taxonomy, we identified that Tiers 1 through 3 are well represented by validated research, while Tier 4—fully adaptive, explainable, privacy-preserving detection platforms—remains largely theoretical.

The hybrid NeuroSecure ensemble framework proposed here combines CNN, RNN (LSTM), and Transformer models through a majority-voting fusion layer, achieving detection accuracy above 96% and F1-scores above 94% across both NSL-KDD and CICIDS2017 benchmark datasets. The results confirm that the three architectures capture meaningfully complementary representations of network traffic: spatial feature patterns, temporal sequence dependencies, and global contextual relationships. Their combination through ensemble fusion outperforms any single architecture by a consistent margin.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15528

At the same time, the survey and experimental analysis make clear that detection accuracy on established benchmarks, while necessary, is not sufficient for operational effectiveness. The most important open problems in the field are not algorithmic—they are architectural and systems-level: how to build IDS platforms that adapt to distribution shift without retraining, operate on encrypted traffic, explain their outputs to analysts, protect the privacy of training data, and maintain sub-millisecond detection latency at enterprise scale.

Future work from the NeuroSecure project will focus on three directions. First, we plan to explore online learning techniques that allow the ensemble to update incrementally as new labeled traffic samples become available, without catastrophic forgetting of previously learned patterns. Second, we will investigate Transformer variants optimized for inference latency—particularly linear attention approximations—to assess whether the detection quality advantages of the Transformer component can be preserved under real-time throughput constraints. Third, we will begin experiments with federated training across simulated network nodes to assess the viability of collaborative model improvement without centralized data collection.

REFERENCES

[1]. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019.
[2]. K. Kim, A. Park, and D. H. Lee, "Application of Deep Learning to Real-Time Web Intrusion Detection," IEEE Access, vol. 8, pp. 70536–70547, 2020.
[3]. J. Lansky, S. Ali, M. M. Mokhtar, and M. K. Majeed, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," IEEE Access, vol. 9, pp. 101574–101599, 2021.
[4]. W. Dai, X. Li, W. Ji, and S. He, "Network Intrusion Detection Method Based on CNN, BiLSTM and Attention Mechanism," IEEE Access, vol. 11, pp. 8830–8840, 2023.
[5]. N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Dataset for Network Intrusion Detection Systems," in Proc. IEEE MILCOM Workshop MIL-Cybersecurity, pp. 1–6, 2015.
[6]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Proc. IEEE CISDA, pp. 1–6, 2009.
[7]. S. Rizvi, J. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the Internet of Things (IoT): A Security Taxonomy for IoT," JAIT, vol. 15, no. 1, pp. 10–18, 2024.
[8]. C. Wang and S. K. Jena, "Network Traffic Classification Using CNN," in Proc. IEEE INFOCOM Workshops, pp. 1–6, 2019.

