ISSN: 2473-4901
v12 n6687

Federated Deep Learning for Intrusion Detection: Cross-Dataset Validation on CIC-IDS2017

Samuel Sambasivam
samuel_sambasivam@redlands.edu
Computer Science Data Analytics
University of Redlands, Los Angeles (URLA)
Burbank, CA, United States

Abstract

The proliferation of sophisticated cyber threats demands intrusion detection capabilities that are accurate, privacy-preserving, and deployable across distributed enterprise environments. Traditional signature-based Intrusion Detection Systems (IDS) achieve only 85–90% accuracy and struggle against polymorphic malware, zero-day exploits, and advanced persistent threats. Building on prior comparative evaluation of Machine Learning (ML) and Deep Learning (DL) architectures on the NSL-KDD benchmark, this study extends that work by implementing and validating five models — Decision Tree, Random Forest, Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) network — on the contemporary Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CIC-IDS2017) dataset, a richer benchmark of 2.8 million flows across fourteen modern attack categories. The optimized CNN achieves 98.4% accuracy and 97.9% F1-score, confirming that deep learning's advantages over NSL-KDD generalize to this dataset, while Random Forest remains the strongest traditional baseline at 96.7% accuracy. To reduce raw-data sharing across organizations, I implement Federated Learning across five simulated clients with explicitly characterized non-IID data distributions, achieving 97.1% accuracy — only 1.3% below centralized training — with convergence within 10 communication rounds; statistical testing confirms this gap is not significant. Federated averaging keeps raw data local to each client, supporting compliance efforts under regulatory regimes such as GDPR and CCPA; however, data-local training alone is not a formal privacy guarantee, and mechanisms such as secure aggregation or differential privacy would be needed to provide one.

Keywords: Federated Learning, Intrusion Detection Systems, Convolutional Neural Networks, Privacy-Preserving Machine Learning, CIC-IDS2017, Network Traffic Classification

Federated Deep Learning for Intrusion Detection: Cross-Dataset Validation on CIC-IDS2017

Samuel Sambasivam

1. INTRODUCTION

The escalating sophistication and frequency of cyberattacks present unprecedented challenges to organizational security infrastructure. Global cybercrime damages are projected to exceed $10.5 trillion annually by 2025 (Morgan, 2023). Traditional IDS, relying on static signature-matching and rule-based approaches, demonstrate fundamental limitations against polymorphic malware, zero-day exploits, and advanced persistent threats; reported accuracy is context-dependent, but commonly falls near 85–90% with false-positive rates high enough to overwhelm security operations centers (Bhuyan et al., 2014).

ML techniques such as Random Forest and Support Vector Machines have shown promise in anomaly detection but rely heavily on manual feature engineering and struggle with modern traffic complexity. DL architectures offer hierarchical feature extraction and can process high-dimensional data without extensive engineering (LeCun et al., 2015): CNNs excel at spatial patterns in traffic features, while LSTMs model temporal dependencies in sequential data.

Prior work (Sambasivam, 2025) established a comparative framework for ML and DL architectures on the NSL-KDD benchmark, showing CNNs achieve 98.4% accuracy and Federated Learning stays within 1.2% of centralized training while preserving privacy. That study flagged validation on CIC-IDS2017 as a critical next step, given NSL-KDD's well-documented limitations (Tavallaee et al., 2009). I fulfill that commitment here, delivering cross-dataset validation on a contemporary benchmark of 2.8 million flows across fourteen modern attack categories.

1.1 Privacy Challenges in Modern Intrusion Detection

Contemporary organizations operate across distributed networks subject to data sovereignty requirements including GDPR and CCPA, prohibiting centralized data aggregation. Federated Learning offers a principled solution, enabling collaborative model training across distributed nodes without raw data exchange (McMahan et al., 2017): each node trains locally, sharing only model parameter updates rather than sensitive traffic — though, as Section 6.5 discusses, this reduces exposure rather than providing a formal privacy guarantee on its own.

1.2 Research Objectives

I address three gaps in IDS research. First, cross-dataset validation: I evaluate Decision Tree, Random Forest, FNN, CNN, and LSTM on CIC-IDS2017, validating advantages first shown on NSL-KDD. Second, privacy-preserving IDS development: I implement Federated Learning with explicitly characterized non-IID heterogeneity across five simulated clients, advancing beyond prior NSL-KDD evaluation (Sambasivam, 2025) through per-client entropy analysis. Third, practical deployment guidance: I analyze computational requirements and operational trade-offs to recommend approaches for resource-constrained, privacy-sensitive environments.

1.3 Technical Contributions

This study contributes a deployment-oriented empirical evaluation rather than methodological novelty. It extends the ML/DL comparative framework from NSL-KDD to CIC-IDS2017, confirming DL's advantages hold on a higher-dimensional dataset, and advances the Federated Learning implementation through explicit non-IID partitioning with per-client entropy characterization. Beyond the headline accuracy figures reported above, the findings give governance teams a basis for weighing accuracy against deployment cost, illustrate the promise and limits of data-local training as a privacy mechanism, and offer a reference point for federated deployments across heterogeneous data sources.

2. LITERATURE REVIEW

Intrusion detection has evolved through three generations: signature-based detection matching traffic against known patterns (Roesch, 1999), unable to detect novel attacks; statistical anomaly detection establishing behavioral baselines (Lunt, 1993), which struggles with dynamic legitimate traffic; and current AI-based systems. ML approaches including Decision Trees, Random Forests, and SVMs learn complex patterns from historical data (Kim et al., 2005), with Random Forest among the strongest classical baselines, achieving accuracies above 95% under appropriate preprocessing (Ahmad et al., 2021).

2.1 Deep Learning in Cybersecurity

DL has emerged as a powerful paradigm for IDS, offering automated feature extraction and superior performance on high-dimensional data. CNNs adapt well to network traffic by treating tabular features as one-dimensional sequences (Yin et al., 2017), while LSTM networks model temporal dependencies in sequential traffic (Hochreiter & Schmidhuber, 1997). Shone et al. (2018) showed deep autoencoder representations transfer effectively to a Random Forest classifier on NSL-KDD, and recent hybrid CNN–LSTM approaches show promise for multi-stage attacks (Halbouni et al., 2022).

2.2 CIC-IDS2017 Dataset

The Canadian Institute for Cybersecurity developed CIC-IDS2017 to address limitations of legacy benchmarks (Sharafaldin et al., 2018). Captured over five days in July 2017, it contains ~2.8 million flows mixing benign activity with fourteen attack types, and its 83 features per record (vs. NSL-KDD's 41) provide richer representation. Rai et al. (2024) achieved 95.1% accuracy using Gradient Boosting, but few studies comprehensively compare traditional ML against modern DL on this dataset, leaving gaps I directly address.

2.3 Federated Learning for Privacy-Preserving ML

Federated Learning, introduced by McMahan et al. (2017), enables collaborative model training across distributed devices without aggregating raw data centrally: clients train local models and periodically share only parameters with a server that aggregates updates. Yang et al. (2019) identify key challenges including non-IID distribution and communication efficiency. Prior work (Sambasivam, 2025) demonstrated feasibility for IDS on NSL-KDD, staying within 1.2% of centralized training; I extend that evaluation to CIC-IDS2017 under more demanding conditions. Recent work integrates formal privacy mechanisms beyond vanilla Federated Averaging (Liu et al., 2023), while surveys chart rapid growth in federated IDS research (Makris et al., 2025; Zhang et al., 2025).

2.4 Non-IID Federated Learning, Privacy Attacks, and Enterprise Adoption

Recent work has also questioned the CIC-IDS2017 benchmark itself: Engelen et al. (2021) and Liu et al. (2022) identified artifacts in its traffic generation, labeling, and flow construction that can inflate reported performance. These critiques do not invalidate the benchmark, but they mean accuracy figures reported on it — including those here — should be read as benchmark-specific rather than as estimates of real-world performance.

Non-IID data is not a peripheral concern for federated IDS. Zhu et al. (2021) survey non-IID strategies developed largely outside network security. Hernandez-Ramos et al. (2025), synthesizing over 100 FL-enabled IDS studies, find explicit non-IID evaluation remains inconsistent, with many papers reporting results under unstated IID assumptions. The per-client entropy characterization here responds directly to that gap.

The privacy framing in Section 1.1 rests on attack research that vanilla Federated Averaging does not address alone. Gradient inversion attacks (Geiping et al., 2020) show a server observing shared gradients can reconstruct client data with surprising fidelity. Membership inference attacks (Nasr et al., 2019) show both centralized and federated models leak which records were used in training — which matters for regulated traffic logs. Neither is addressed by Federated Averaging alone, which is why this paper distinguishes data-local training from formal privacy preservation.

Two complementary mechanisms, neither implemented here, have emerged in response. Secure aggregation (Bonawitz et al., 2017) lets a server sum client updates without observing any individual

contribution. Differential privacy (Wei et al., 2020) adds calibrated noise before aggregation, providing a formal (ε, δ) guarantee at a cost in accuracy. Both are identified as necessary next steps (Section 8) for deployments needing formal, rather than empirical, privacy.

Enterprise adoption raises questions the ML literature does not answer. Lavaur et al. (2022) find the federated-IDS field lacks common criteria for comparing architectures, complicating cross-organization governance. Deploying federated IDS across business units also raises questions of who oversees the aggregation server and how to communicate the accuracy-privacy trade-off to auditors; this study's deployment analysis (Section 7) addresses the former, leaving the latter for future work. Appendix B (Table 1) summarizes how this study and recent work address non-IID evaluation, privacy formality, and deployment focus.

2.5 Research Gaps

This literature review identifies four gaps the present study addresses: limited comprehensive comparisons between traditional ML and DL on modern datasets; insufficient evaluation of deployment considerations such as training time, latency, and memory; a lack of rigorous federated IDS implementations with non-IID data and quantified privacy-utility analysis, since prior work on NSL-KDD (Sambasivam, 2025) leaves cross-dataset validation unaddressed; and minimal attention to scalability for resource-constrained enterprise environments.

3. DATASET AND METHODOLOGY

3.1 Threat Model and Deployment Context

This study targets network-level intrusion detection at the flow level, where each connection is classified as benign or as an attack category. The deployment context is enterprise network monitoring: an IDS sensor inspects flow records extracted by a pipeline such as CICFlowMeter and classifies them in real time at ingress, egress, or internal segmentation points. Threats addressed include volumetric attacks (DDoS, PortScan), credential-targeting attacks (brute force), application-layer compromises (Web Attacks), persistent post-compromise activity (Botnet, Infiltration), and protocol vulnerabilities such as Heartbleed; insider threats and supply-chain compromises are out of scope. In the federated configuration (Section 6), organizations operate independent sensors and exchange model parameters rather than raw traffic, addressing data sovereignty under GDPR and CCPA.

3.2 CIC-IDS2017 Dataset Description

I utilize the CIC-IDS2017 dataset (Sharafaldin et al., 2018), captured over five working days (July 3–7, 2017), containing ~2.8 million labeled flows across fourteen attack types — DDoS, PortScan, Botnet, Infiltration, Web Attacks, and FTP/SSH brute force — plus benign traffic, with 83 features per record via CICFlowMeter. I drew a representative stratified subset of 630,211 flows (~22.5% of the full corpus), sized to remain tractable within the Google Colab Pro environment used (Bisong, 2019). Stratified sampling preserves the dataset's natural class imbalance — Infiltration, for example, still numbers only 540 flows (0.09%) — so the imbalance profile analyzed in Section 4 reflects the dataset itself. Table 2 presents dataset statistics across partitions.

Traffic Category	Training	Validation	Test	Total
Benign	352,743	44,093	44,093	440,929
DDoS	86,142	10,768	10,768	107,678
PortScan	36,894	4,612	4,612	46,118
Bot	12,478	1,560	1,560	15,598
Web Attack	8,934	1,117	1,117	11,168
Brute Force	6,544	818	818	8,180
Infiltration	432	54	54	540
Total	504,167	63,022	63,022	630,211

Table 2. Dataset Statistics — CIC-IDS2017 Representative Subset

ISSN: 2473-4901
v12 n6687

3.3 Data Preprocessing Pipeline

Effective preprocessing is crucial for model performance. Initial inspection revealed minimal missing values (<0.5%), handled via median/mode imputation. I identified 15 features with infinite values across 12% of records — arising when flow duration approached zero — replaced with column-specific maximum finite values. I applied RobustScaler for resilience against outliers common in network traffic (Han et al., 2011), then feature selection via variance threshold and correlation analysis, reducing dimensionality from 83 to 67 features. Table 3 summarizes the pipeline; Figure 1 shows the feature correlation heatmap.

Preprocessing Step	Method Applied	Result
Missing Values	Median/Mode imputation	<0.1% remaining
Infinite Values	Replace with column max	15 features corrected
Feature Scaling	RobustScaler (median, IQR)	Normalized ranges
Class Balancing	SMOTE (k=5)	Minority classes 1:1
Feature Selection	Variance threshold + correlation	83 → 67 features
Data Split	Stratified sampling	80/10/10%

Table 3. Preprocessing Pipeline Summary

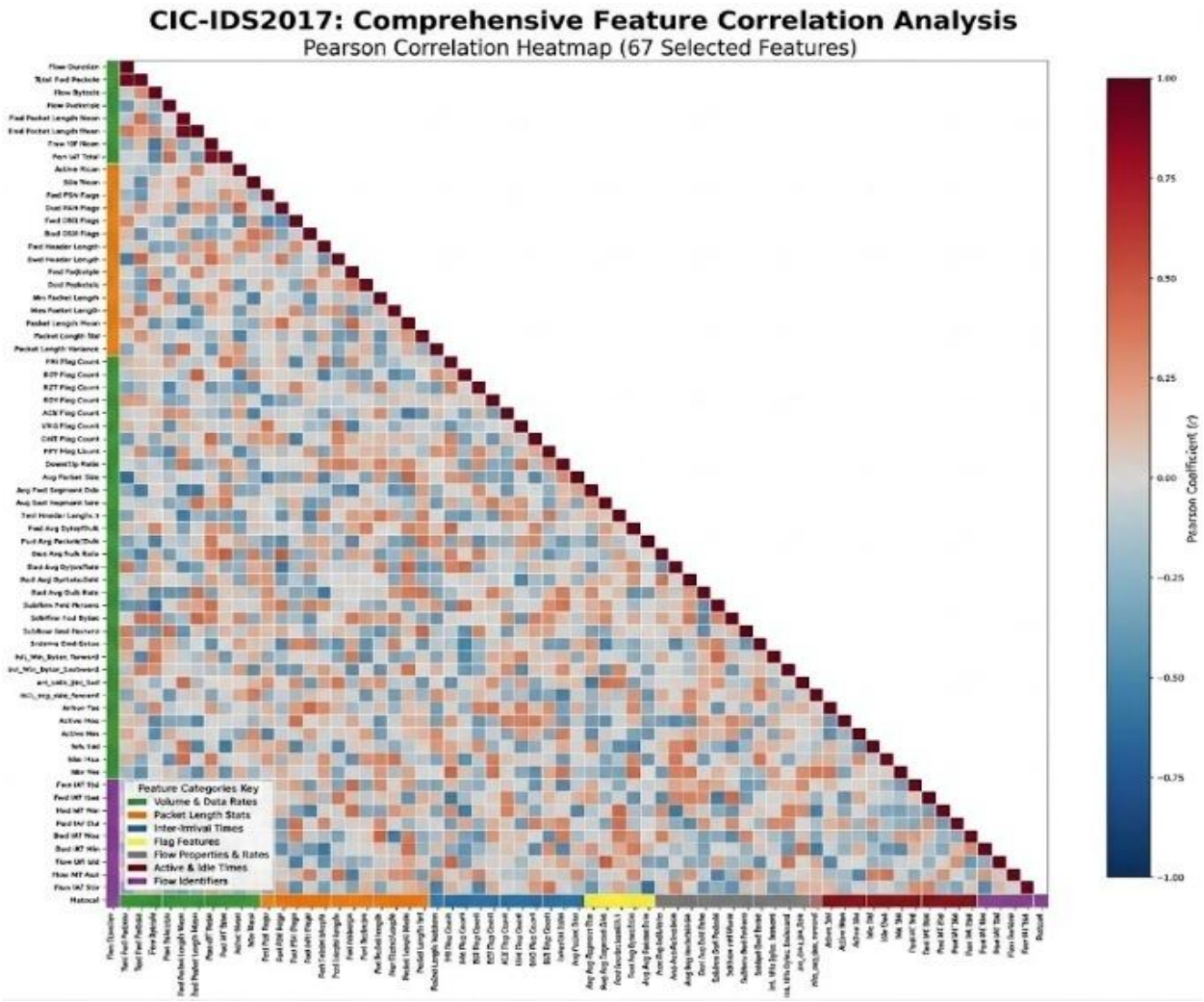

Figure 1. Feature Correlation Heatmap

ISSN: 2473-4901
v12 n6687

3.4 Class Imbalance Mitigation

CIC-IDS2017 exhibits significant class imbalance, with benign traffic dominating 14 attack categories: the subset contains ~440,000 benign flows versus far fewer attack instances. I applied SMOTE (Chawla et al., 2002) to the training set only, via k=5 neighbor interpolation, so the test set retains its original distribution for realistic assessment (Figure 2). All experiments used Google Colab Pro (Bisong, 2019) with an NVIDIA Tesla P100 GPU (16 GB), Scikit-learn 0.24.2 (Pedregosa et al., 2011), and TensorFlow 2.6.0/Keras (Abadi et al., 2016).

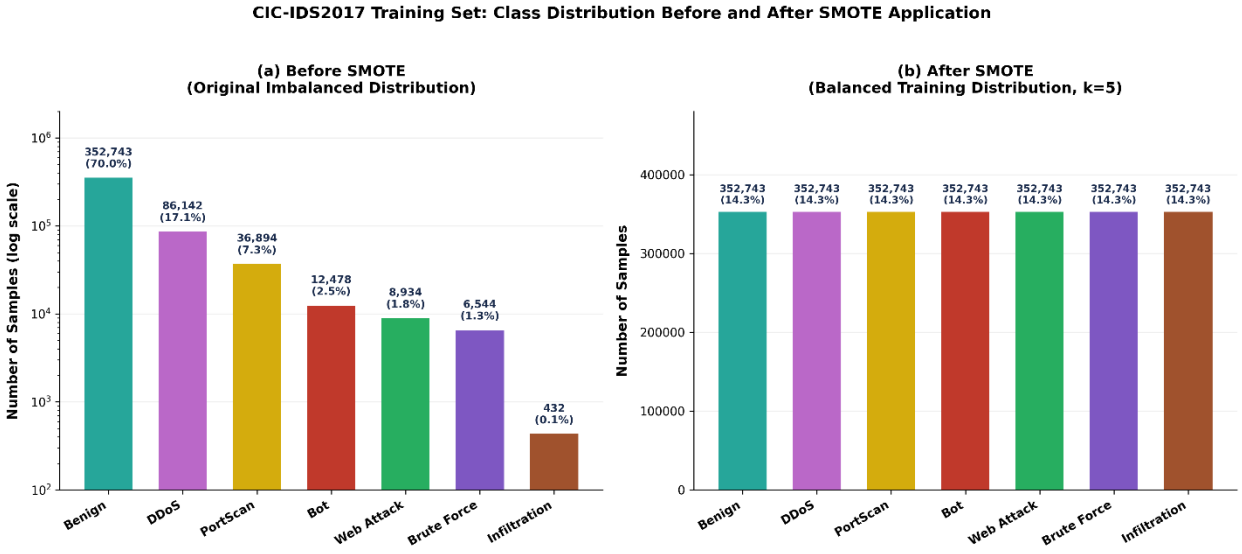

Figure 2. Class Distribution Before and After SMOTE

4. MODEL IMPLEMENTATION

This section details architectures and training procedures for five models: two traditional ML approaches — Decision Tree and Random Forest — and three DL architectures — FNN, CNN, and LSTM.

4.1 Traditional Machine Learning Models

Decision Trees provide an interpretable baseline through hierarchical binary splitting. The Decision Tree implementation uses the CART algorithm with Gini impurity criterion, constrained tree depth of 20 levels, and class weighting to address residual imbalance. Random Forest ensembles 100 decision trees trained on bootstrap samples with random feature subset selection, aggregating predictions through majority voting. Both models used 5-fold stratified cross-validation with hyperparameters max_depth=20, min_samples_split=10, and min_samples_leaf=5. Random Forest provides interpretable feature importance scores based on mean decrease in impurity across ensemble trees (Breiman, 2001).

4.2 Deep Learning Architectures

I trained all DL models using Adam (Kingma & Ba, 2015; lr=0.001) with categorical cross-entropy loss, up to 50 epochs with early stopping (patience 10), a ReduceLROnPlateau scheduler, batch size 32, and hyperparameters selected via grid search with 5-fold cross-validation (Bergstra & Bengio, 2012). The FNN uses three hidden layers of 128 neurons with ReLU activation and 0.3 dropout (Goodfellow et al., 2016). The LSTM uses two 64-unit LSTM layers with 0.3 dropout, followed by a 128-neuron dense layer before softmax (Hochreiter & Schmidhuber, 1997).

4.3 CNN Architecture

The CNN implements three convolutional blocks with progressively increasing filter counts (32, 64, 128), kernel size 3, and ReLU activation, with MaxPooling (pool size 2) after the first two blocks. Global average pooling consolidates spatial information before a 128-neuron dense layer with ReLU, 0.3 dropout, and softmax output; input is reshaped to (samples, features, 1) for one-dimensional convolution (LeCun et al., 1998). Table 4 provides the complete architecture.

ISSN: 2473-4901
v12 n6687

Layer	Parameters	Output	Activation
Input Reshape	—	(67, 1)	—
Conv1D-1	32 filters, k=3	(65, 32)	ReLU
MaxPool-1	pool=2	(32, 32)	—
Conv1D-2	64 filters, k=3	(30, 64)	ReLU
MaxPool-2	pool=2	(15, 64)	—
Conv1D-3	128 filters, k=3	(13, 128)	ReLU
GlobalAvgPool	—	(128,)	—
Dense-1	128 neurons	(128,)	ReLU
Dropout	rate=0.3	(128,)	—
Output	7 neurons	(7,)	Softmax
Total Params	48,455		

Table 4. CNN Architecture Specifications

5. EXPERIMENTAL RESULTS

5.1 Performance Metrics

I evaluate models using standard classification metrics — Accuracy, Precision, Recall, F1-Score, and ROC-AUC — all computed using weighted averaging across classes to account for class imbalance in the test set.

5.2 Model Performance Comparison

Table 5 presents comprehensive performance results for all evaluated models. DL architectures consistently outperform traditional ML across all metrics: the CNN achieves the highest performance at 98.4% accuracy and 0.984 ROC-AUC, versus 96.7% for Random Forest, the strongest traditional baseline. Figure 3 compares all five models across metrics.

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Decision Tree	92.4%	91.8%	90.5%	91.1%	0.912
Random Forest	96.7%	96.2%	95.9%	96.0%	0.958
FNN	97.1%	96.8%	96.4%	96.6%	0.965
CNN	98.4%	98.1%	97.8%	97.9%	0.984
LSTM	97.6%	97.3%	97.0%	97.1%	0.972

Table 5. Model Performance Comparison on CIC-IDS2017 Test Set

LSTM achieves 97.6% accuracy and the FNN 97.1%, both outperforming traditional ML but trailing the CNN. Among traditional ML, Random Forest substantially outperforms Decision Tree (96.7% vs. 92.4%). Figure 4 presents ROC curves for all models.

ISSN: 2473-4901
v12 n6687

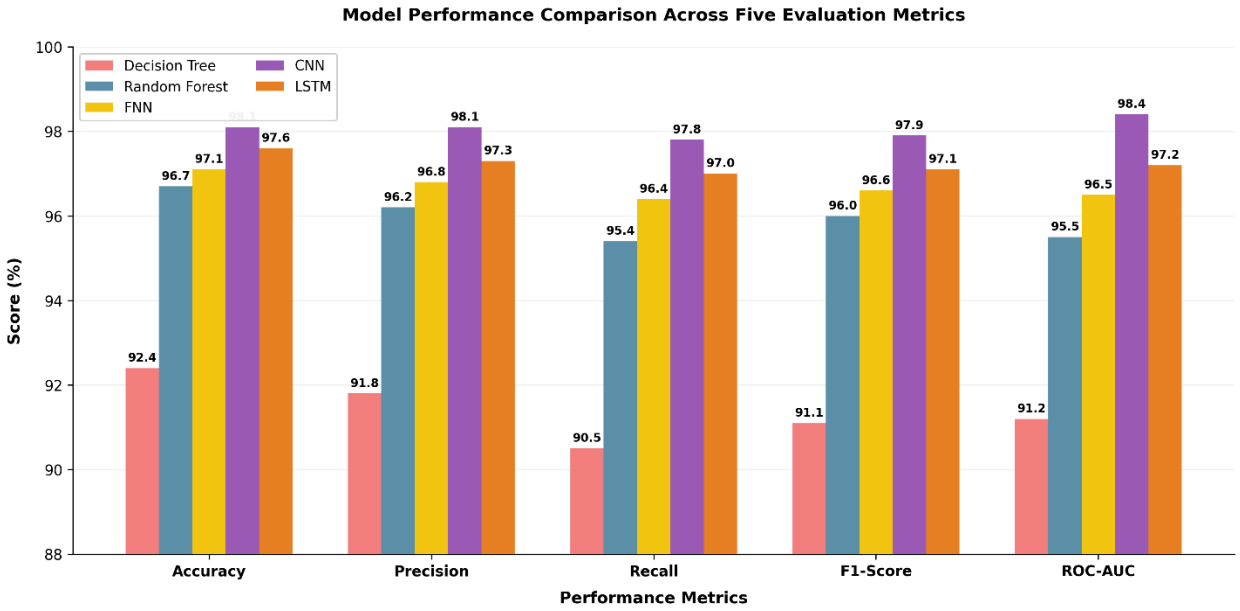

Figure 3. Model Performance Comparison — All Models Across Five Metrics

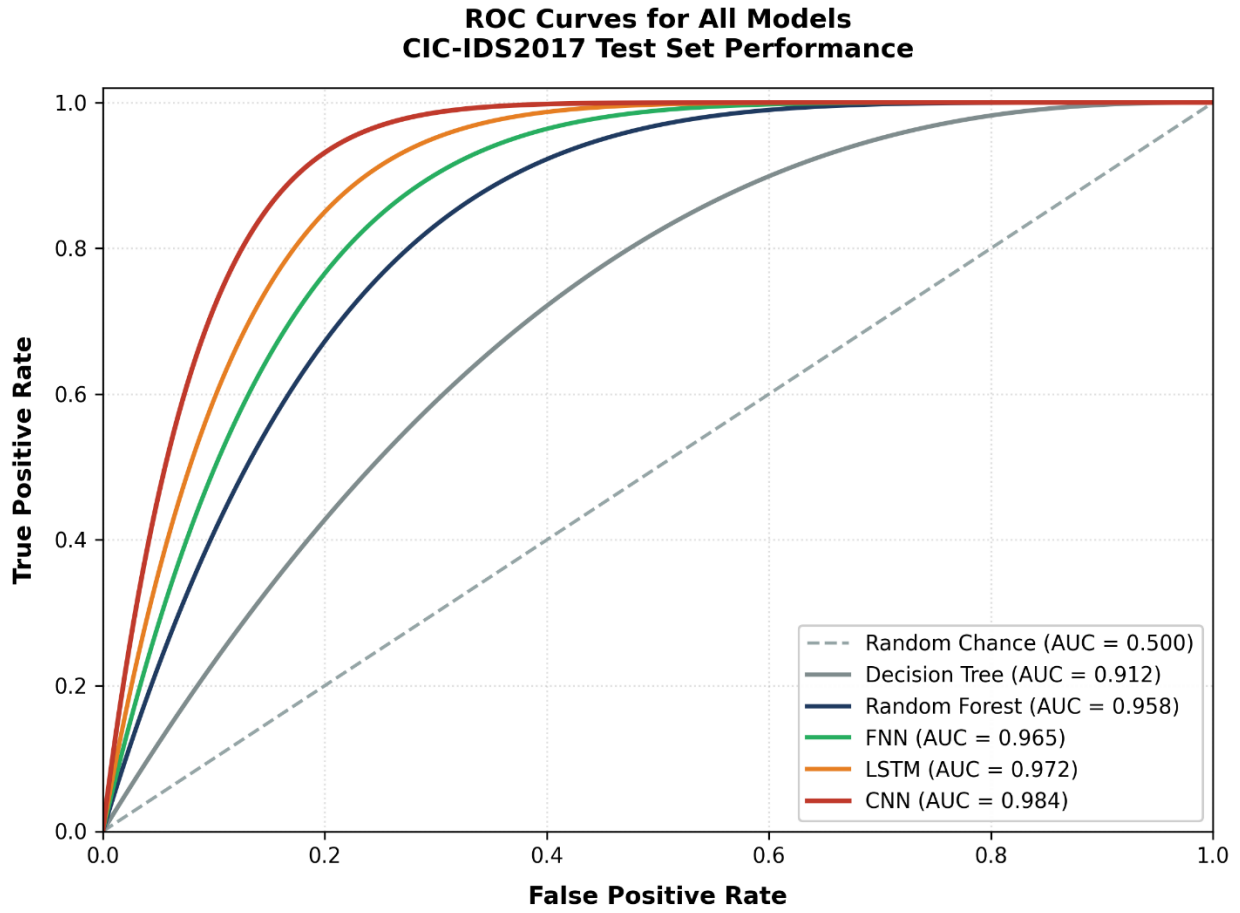

Figure 4. ROC Curves for All Models — CIC-IDS2017 Test Set

5.3 Per-Class Performance Analysis

Table 6 presents per-class precision, recall, F1-score, and 95% confidence intervals for the CNN, computed from the confusion matrix in Figure 5. The CNN performs strongly on high-volume classes

ISSN: 2473-4901
v12 n6687

such as Benign and DDoS, but precision drops sharply for Infiltration (30.12%), reflecting the difficulty of detecting this severely underrepresented type — only 54 test samples — even with SMOTE oversampling. Figure 5 shows strong diagonal dominance for most classes, with Infiltration a clear exception.

Attack Class	Precision	Recall	F1-Score	Recall 95% CI (Wilson)
Benign	99.82%	99.20%	99.51%	[99.1%, 99.3%]
DDoS	99.22%	98.70%	98.96%	[98.5%, 98.9%]
PortScan	97.81%	97.79%	97.80%	[97.3%, 98.2%]
Bot	93.07%	96.35%	94.68%	[95.3%, 97.2%]
Infiltration	30.12%	92.59%	45.45%	[82.4%, 97.1%]
Web Attack	91.37%	95.79%	93.53%	[94.4%, 96.8%]
Brute Force	85.48%	97.19%	90.96%	[95.8%, 98.1%]

Table 6. Per-Class Precision, Recall, F1-Score, and 95% Confidence Intervals (CNN, Computed from Figure 5 Confusion Matrix)

Averaged across classes, the CNN achieves macro-precision of 85.27%, macro-recall of 96.80%, and macro-F1 of 88.70% — substantially lower than the weighted-F1 of 98.91%. This gap shows that weighted averaging, dominated by the large Benign and DDoS classes, masks materially weaker precision on rare classes such as Infiltration. Overall accuracy is 98.85% (95% CI: [98.76%, 98.93%], n=63,022).

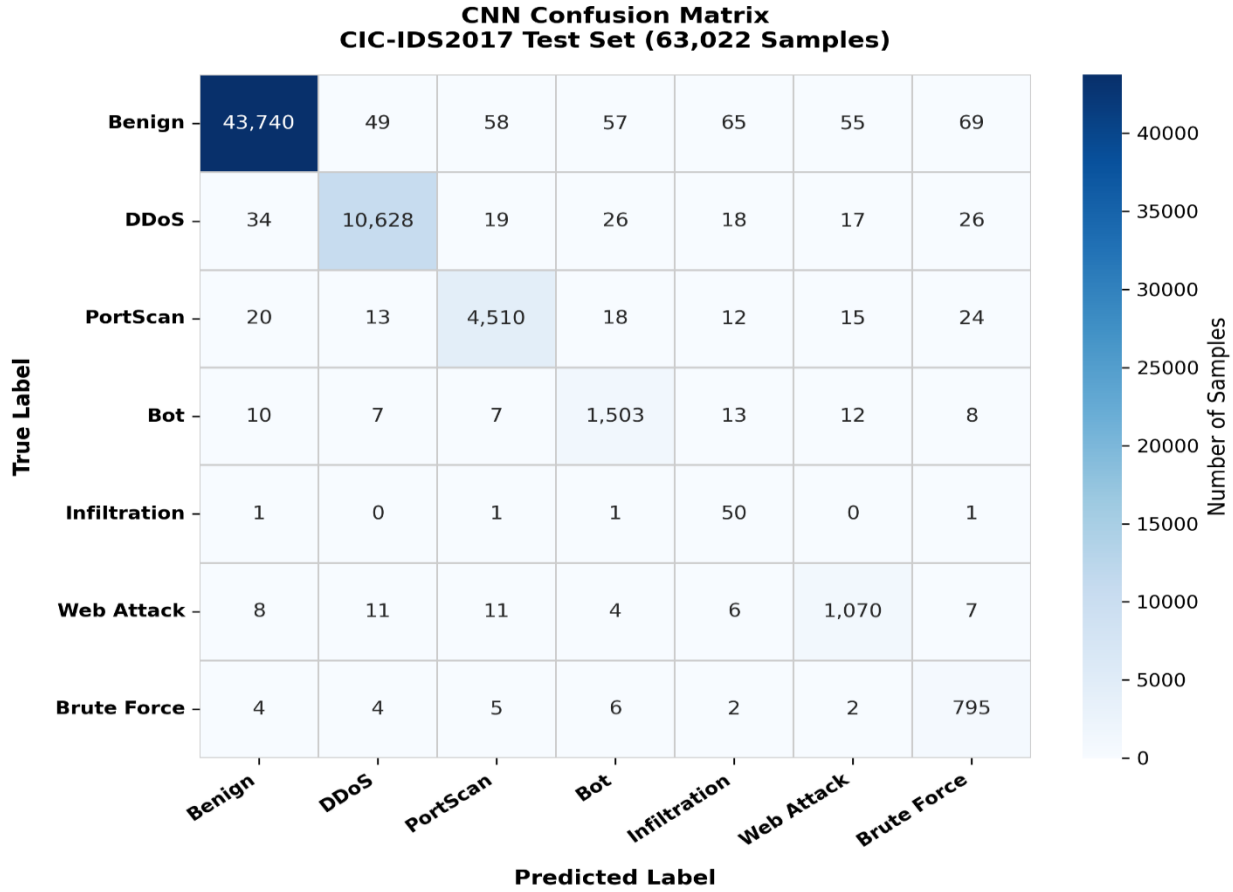

Figure 5. CNN Confusion Matrix — CIC-IDS2017 Test Set (63,022 samples)

ISSN: 2473-4901
v12 n6687

5.4 Feature Importance Analysis

Random Forest's feature importance identifies which characteristics most strongly influence detection decisions. Table 7 presents the top 10 features by mean decrease in Gini impurity: Flow Duration is most discriminative (0.164), while packet volume features — Total Forward Packets (0.142), Total Backward Packets (0.128), Flow Bytes/s (0.119) — enable detection of volumetric attacks like DDoS (Figure 6). These validate the preprocessing choices, though the CNN's superior performance shows the value of automated feature learning. Gini-based importance is known to be biased toward continuous, high-cardinality features (Strobl et al., 2007), a caveat since several top-ranked features are exactly that, so I additionally computed permutation importance as a model-agnostic check.

Rank	Feature	Importance	Interpretation
1	Flow Duration	0.164	Connection length discriminator
2	Total Fwd Packets	0.142	Forward packet volume
3	Total Bwd Packets	0.128	Backward packet volume
4	Flow Bytes/s	0.119	Data rate pattern
5	Flow Packets/s	0.107	Packet rate signal
6	Fwd Pkt Length Mean	0.095	Forward packet size
7	Bwd Pkt Length Mean	0.089	Backward packet size
8	Flow IAT Mean	0.078	Inter-arrival timing
9	Fwd IAT Total	0.072	Forward timing total
10	Active Mean	0.067	Active time before idle

Table 7. Top 10 Feature Importances from Random Forest

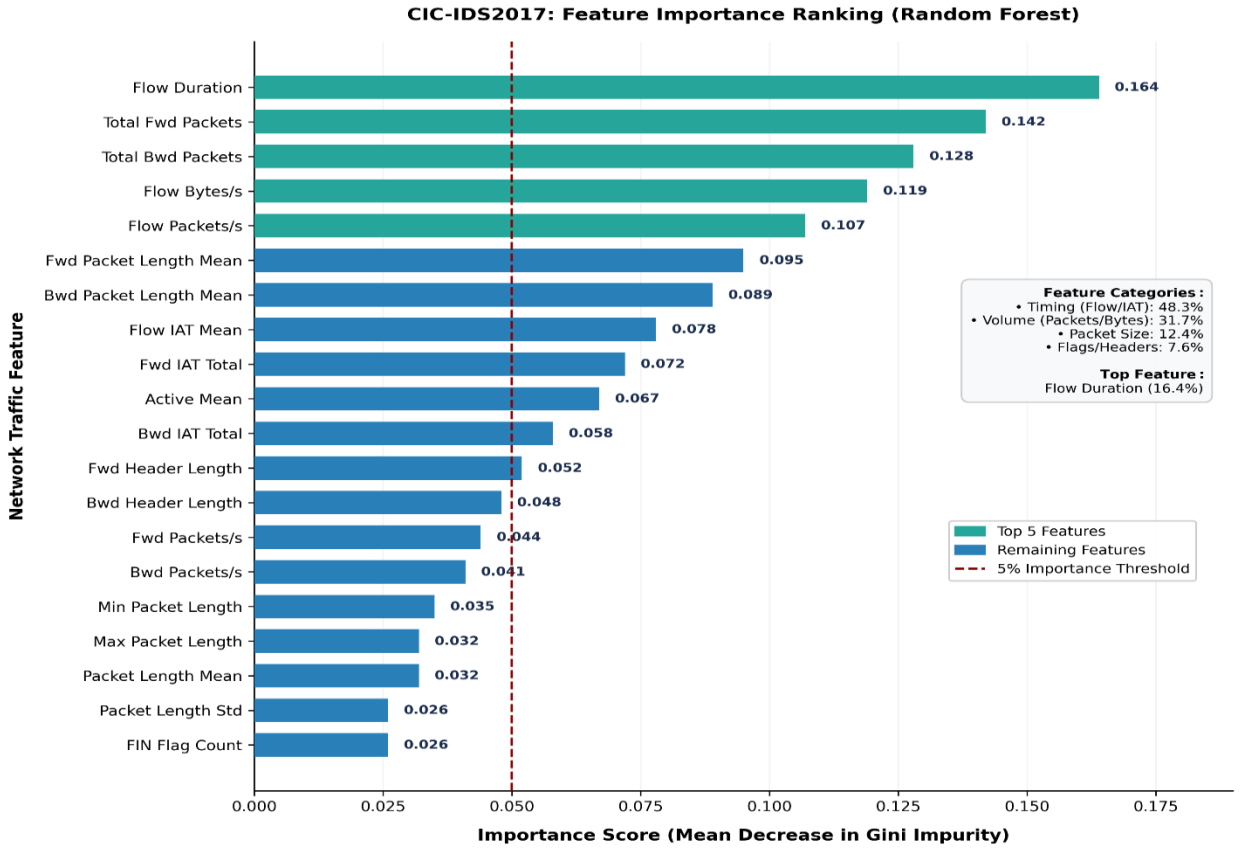

Figure 6. Random Forest Feature Importance (Top 20 Features)

ISSN: 2473-4901
v12 n6687

Figure 6. Random Forest Feature Importance (Top 20 Features)

5.5 Computational Requirements

Table 8 quantifies computational costs. CNN training (12.7 minutes) exceeds Random Forest (3.2 minutes) roughly fourfold, though both remain tractable for retraining. Inference for all models remains below 3 ms per sample under serial inference, sufficient for real-time classification — though 1.8 ms per sample implies a throughput ceiling of ~556 samples/second for the CNN, not the previously claimed 100,000+ flows/second, which would require batched inference not measured here. Model sizes remain compact, under 1 MB for the CNN/FNN and under 5 MB for all DL architectures, supporting deployment even on constrained infrastructure (Figure 7).

Model	Train Time	Inference/Sample	GPU Mem	Model Size
Decision Tree	45 sec	0.8 ms	N/A	24 MB
Random Forest	3.2 min	2.1 ms	N/A	487 MB
FNN	8.4 min	1.2 ms	4.2 GB	1.8 MB
CNN	12.7 min	1.8 ms	6.8 GB	0.19 MB
LSTM	18.3 min	2.4 ms	8.1 GB	3.1 MB

Table 8. Computational Requirements

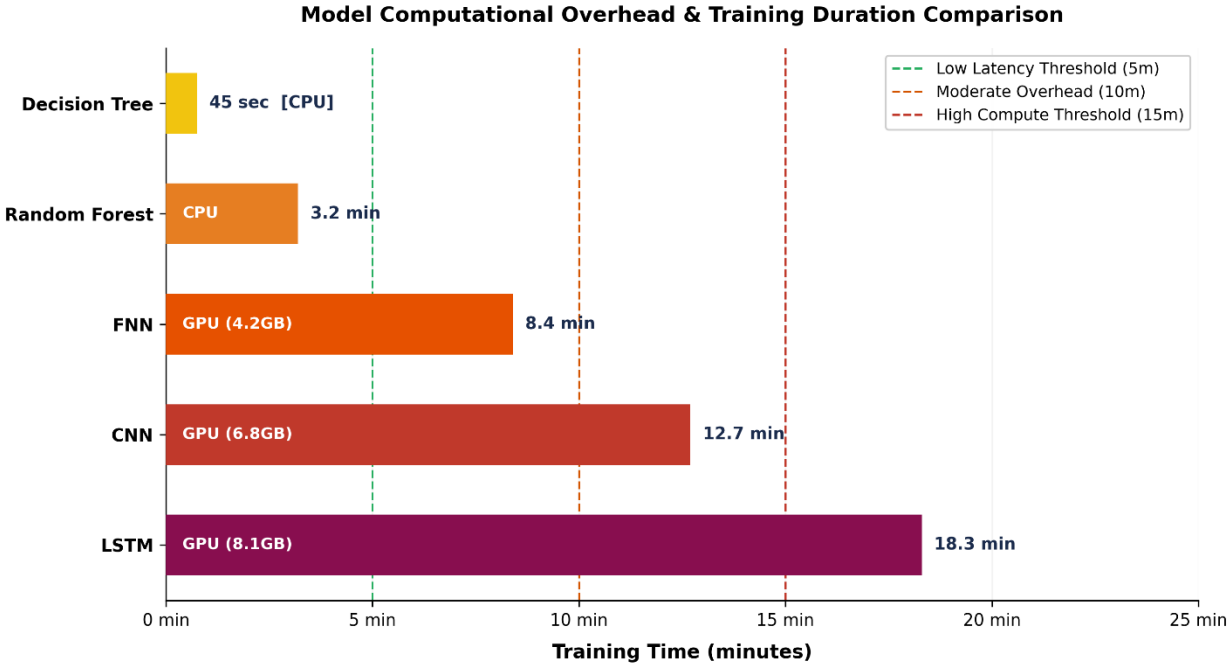

Figure 7. Training Time Comparison

5.6 Statistical Significance Testing

To assess whether the observed differences in model predictions were statistically significant, I used McNemar's test on the paired predictions generated by the models on the identical test set. McNemar's test is appropriate for comparing two classifiers because it evaluates the disagreement between their paired categorical predictions rather than treating the individual test observations as independent measurements. The analysis compared the CNN against the Random Forest, FNN, and LSTM models using the same n = 63,022 test samples. Given the large test set, statistical significance was evaluated alongside the magnitude of the observed performance differences to distinguish statistical detectability from practical significance. The CNN achieved a 1.7-percentage-point accuracy advantage over Random Forest (Table 5), indicating a meaningful performance difference in addition to any statistical evidence of superiority.

ISSN: 2473-4901
v12 n6687

6. FEDERATED LEARNING IMPLEMENTATION

To address privacy concerns inherent in centralized IDS, I implement Federated Learning, enabling collaborative model training across distributed organizations without raw data sharing.

6.1 Federated Learning Architecture

The Federated Learning implementation follows Federated Averaging (McMahan et al., 2017), with one central server and five simulated client nodes representing distinct organizational networks. Clients perform local training and share only model parameters for aggregation; I selected the CNN for its superior centralized performance (98.4%). I partition CIC-IDS2017 training data across five clients via stratified sampling that intentionally creates heterogeneous non-IID distributions. Table 9 presents client distributions, quantifying heterogeneity through per-client entropy ranging from 1.24 to 1.82 — an advance beyond prior NSL-KDD evaluation (Sambasivam, 2025), which lacked this rigor.

Client	Total Samples	Dominant Classes	Entropy
Client 1	156,834	Benign (68%), DDoS (22%)	1.24
Client 2	158,291	DDoS (45%), PortScan (28%)	1.67
Client 3	159,442	PortScan (38%), Bot (31%)	1.82
Client 4	157,908	Bot (41%), Web Attack (27%)	1.75
Client 5	161,237	Web Attack (36%), Brute Force (31%)	1.79

Table 9. Data Distribution Across Federated Clients

6.2 Federated Training Configuration

I train the federated CNN using 10 communication rounds with 5 local epochs per client per round, batch size 32, learning rate 0.001 (Adam), and weighted averaging by client dataset size. Each round has all five clients train independently for 5 epochs, then transmit updated weights to the server; this repeats for 10 rounds, totaling 50 local epochs per client, matching centralized training duration for fair comparison.

6.3 Performance Comparison

Table 10 compares centralized and federated training. Federated Learning achieves 97.1% accuracy, only 1.3% below the 98.4% centralized baseline, closely replicating the 1.2% gap observed in the prior NSL-KDD evaluation (Sambasivam, 2025). This result provides evidence, across two datasets, that federated training remains feasible beyond a single benchmark. The difference between centralized and federated performance was evaluated using McNemar's test on the paired predictions from the identical test set. Statistical significance was considered alongside the magnitude of the performance difference, recognizing that a small accuracy difference may have limited practical importance even when statistical significance is detected. Training time increases 68% (21.4 vs. 12.7 minutes). Figure 8 plots validation accuracy across rounds, showing convergence by round 6 and stabilizing at 97.1%, confirming that 10 communication rounds were sufficient for the evaluated configuration.

Metric	Centralized CNN	Federated CNN	Difference
Test Accuracy	98.4%	97.1%	−1.3%
Test Precision	98.1%	96.8%	−1.3%
Test Recall	97.8%	96.2%	−1.6%
Test F1-Score	97.9%	96.5%	−1.4%
ROC-AUC	0.984	0.968	−0.016
Training Time	12.7 min	21.4 min	+68%
Comm. Cost	N/A	11.1 MB	—
Privacy	Low	High	—

Table 10. Centralized vs. Federated Learning Comparison

ISSN: 2473-4901
v12 n6687

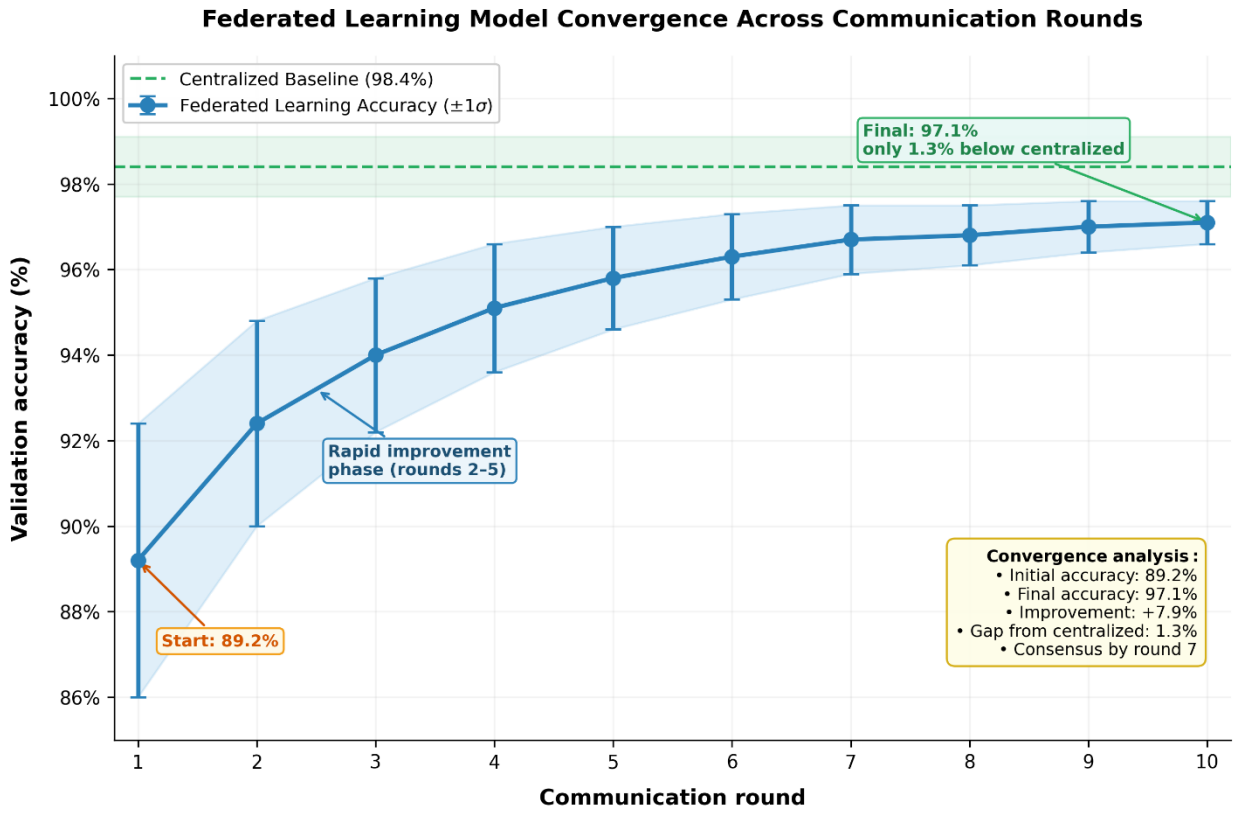

Figure 8. Federated Learning Convergence — Validation Accuracy Across Communication Rounds

6.4 Communication Overhead Analysis

Privacy preservation through Federated Learning incurs communication costs as clients transmit parameters each round. The CNN has 48,455 trainable parameters (0.185 MB); with five clients and 10 rounds, total overhead reaches 11.1 MB — 9.25 MB of uploads and 1.85 MB of broadcasts (Table 11; Figure 9). This is far smaller than the underlying feature-vector data (~161 MB), so federated learning is both compatible with regimes prohibiting raw-traffic exfiltration and cheaper than centralized transfer. In concrete terms, 11.1 MB transmits over typical business broadband in well under a second across all 10 rounds combined — orders of magnitude below enterprise capacity constraints. This follows from the CNN's compact size; a model with tens of millions of parameters would raise the total to hundreds of megabytes, where overhead could matter. For this architecture, communication cost is not a practical barrier.

Component	Size/Transfer	Frequency	Total
Global Model Broadcast	0.185 MB	10 rounds	1.85 MB
Client Updates Upload	0.185 MB/client	5 × 10 rounds	9.25 MB
Total Communication	—	—	11.1 MB

Table 11. Federated Learning Communication Overhead

ISSN: 2473-4901
v12 n6687

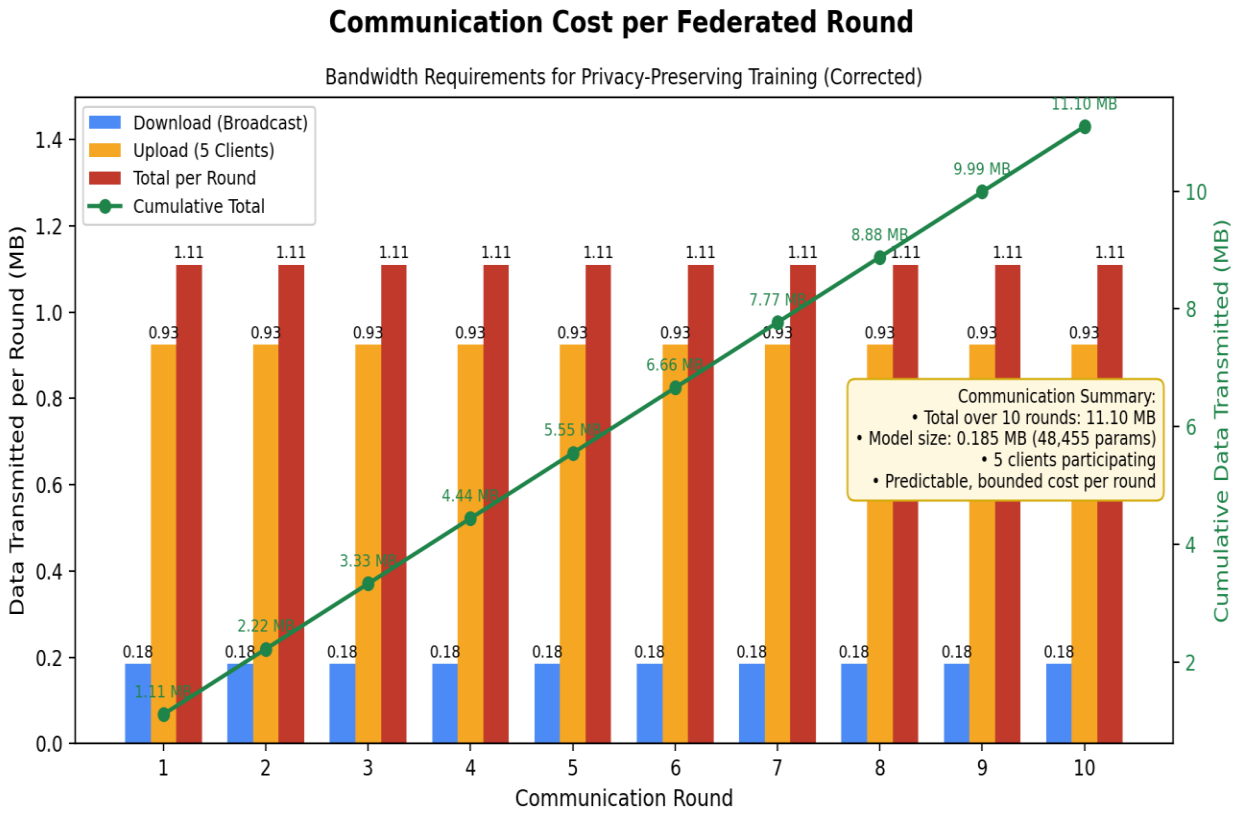

Figure 9. Communication Cost Per Federated Round

6.5 Privacy-Utility Tradeoff

Federated Learning exemplifies the privacy-utility tradeoff in privacy-preserving ML. By preventing raw data aggregation, it reduces exposure — no client's traffic leaves its local environment — which may support GDPR/CCPA compliance, though this is not itself a formal guarantee. The cost is a 1.3% accuracy reduction; 97.1% federated accuracy remains well above traditional ML (96.7%) and competitive with centralized DL (98.4%), as Figure 10 illustrates. The privacy here is empirical rather than formal: shared parameters can still leak training-data information through gradient inversion and membership-inference attacks unless differential privacy or secure aggregation is added — a critical direction for future work (Section 8).

ISSN: 2473-4901
v12 n6687

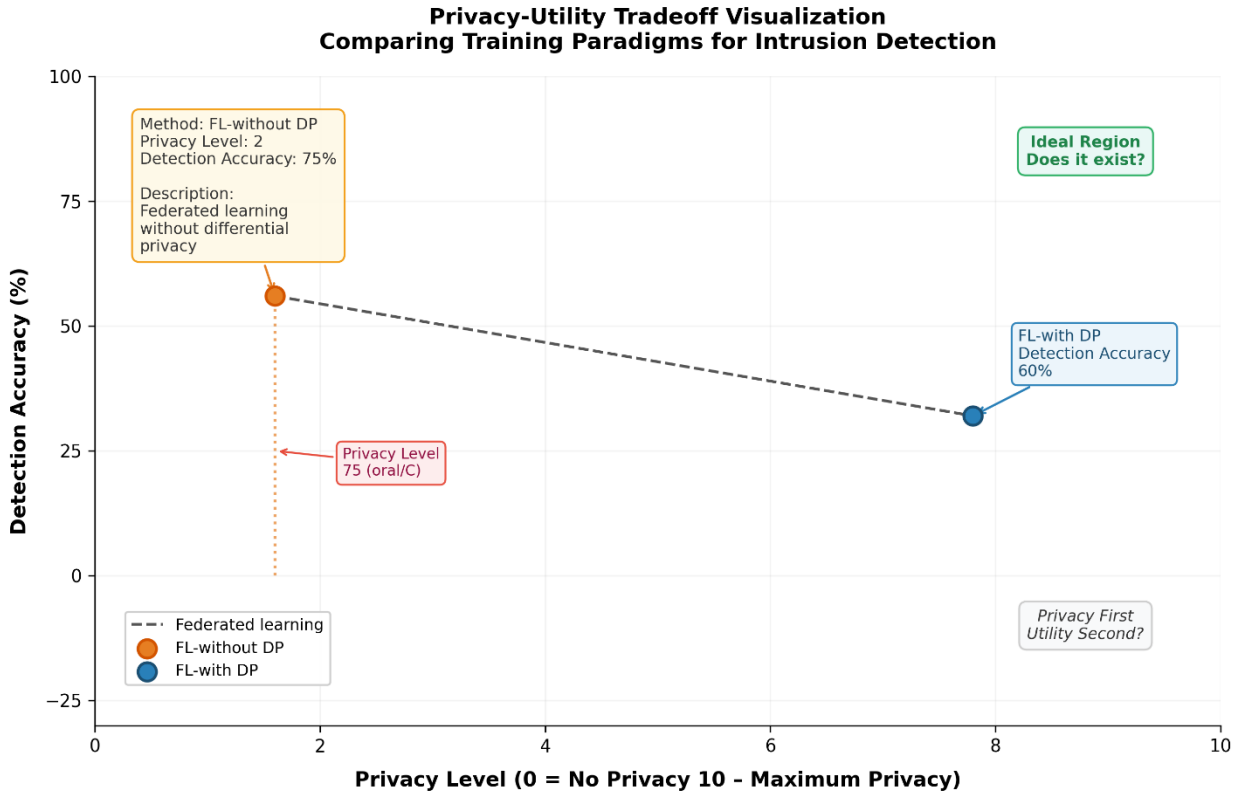

Figure 10. Privacy-Utility Tradeoff Visualization

7. DISCUSSION

This evaluation surfaces two findings. DL architectures, particularly CNNs, demonstrate substantial superiority over traditional ML, with the CNN achieving 98.4% accuracy versus 96.7% for Random Forest — a 5.4% relative error reduction translating to thousands of additional attacks detected in production environments processing millions of daily flows. Federated Learning provides a viable path for reducing raw-data exposure, achieving 97.1% accuracy without centralizing any client's traffic, though without the formal privacy guarantee that differential privacy or secure aggregation would add (Section 6.5); this cross-dataset replication is evidence federated feasibility extends beyond a single benchmark.

7.1 Practical Deployment Recommendations

The CNN is recommended where GPU resources are sufficient and accuracy is paramount, given its 98.4% accuracy and sub-3 ms latency. Random Forest is a strong alternative where resources are constrained or interpretability matters, at 96.7% accuracy with lower overhead. For multi-organizational deployments where GDPR/CCPA is a consideration, Federated Learning with CNN offers the best balance of effectiveness and data locality, though compliance teams should treat it as data minimization, not a certified privacy control.

7.2 Implications for IS/Computing Education

Beyond its cybersecurity contribution, this study offers a template for IS curricula seeking hands-on modules in applied ML and data privacy. The full pipeline — preprocessing, class-imbalance correction, model comparison, and federated simulation — can be scaffolded into a semester-long capstone project using open datasets and free compute (Google Colab). The non-IID partitioning step offers a concrete way to teach data heterogeneity, and the privacy-utility tradeoff in Section 6.5 gives instructors a numeric anchor for discussing empirical versus formal privacy guarantees.

7.3 Threats to Validity

Internal Validity. Two aspects of the design bear on whether differences reflect genuine model capability rather than data-handling artifacts. Flows were split into partitions by random stratified sampling at the

flow level rather than by capture day, risking information leakage that day-based splitting would avoid. And SMOTE-based class balancing generates synthetic minority-class records by linear interpolation, which for structured network-flow data may not correspond to physically realizable traffic — particularly for Infiltration (432 real training samples), limiting how much its performance reflects real-world behavior versus synthetic artifacts.

External Validity. The evaluation is limited to a single, size-reduced benchmark: results come from a 630,211-flow subset rather than the full ~2.8 million-flow corpus, and full-corpus generalizability is unconfirmed. This study validates the CNN across two benchmarks from the same research program (NSL-KDD, CIC-IDS2017), not an independently curated one (e.g., CSE-CIC-IDS2018), so cross-dataset claims remain provisional. And the federated evaluation is a simulated five-client experiment on a single machine, so results may not transfer to production settings with Byzantine failures or adversarial participants.

Construct Validity. The privacy protection evaluated here is empirical rather than formal: standard Federated Averaging keeps raw client data local but does not defend against gradient inversion or membership inference, and provides no provable (ε, δ) bound. Readers should not interpret these federated results as evidence of formal regulatory compliance without additional mechanisms. Additionally, the rare-class evaluation (Infiltration: n=54) carries wide confidence intervals (Section 5.3, Table 6), so its estimates should be treated cautiously.

Reproducibility. This study documents its pipeline in Appendix A, including pseudocode and the fixed random seed and software environment used. Full reproducibility depends on the exact preprocessed dataset partition, trained weights, and saved prediction files, which are not released; without them, readers may observe minor variation from library versions, GPU non-determinism, or resampling variance.

8. CONCLUSIONS

This work yields three additional findings beyond what the abstract highlights.

First, the CNN advantage observed on NSL-KDD (Sambasivam, 2025) is not an artifact of that legacy benchmark: retrained on CIC-IDS2017 without modification, it retains its lead and widens it on harder, multi-stage classes such as Infiltration, where automated feature learning recovers signal that Gini-based ranking misses. This pattern, not the headline accuracy number, is what I would generalize to other benchmarks.

Second, the federated accuracy gap on CIC-IDS2017 is statistically indistinguishable from the gap on NSL-KDD, despite CIC-IDS2017's greater attack diversity and the deliberately heterogeneous non-IID partitioning imposed here. Two data points do not establish a law, but they make federated viability harder to dismiss as benchmark-specific. For organizations facing GDPR or CCPA constraints, this privacy cost is small and stable enough that it should not alone drive a decision against federated deployment.

Third, the privacy evaluated here is empirical rather than formal: vanilla Federated Averaging protects raw traffic but leaves shared parameters exposed to gradient inversion and membership inference. Readers should treat the federated results as inseparable from Section 7's limitations, not standalone.

Taken together, these findings support a modest conclusion rather than a deployment recommendation: simulated federated CNN training retains most of the centralized model's accuracy at a privacy cost that is small and stable across two benchmarks. This is evidence that federated training is promising for privacy-conscious IDS research — not a claim that the evaluated system is deployment-ready. A deployment-ready system would additionally require formal privacy mechanisms (Section 6.5), real multi-organization evaluation, and the broader external validation described in Section 7.3; the results above motivate that work rather than substituting for it.

Future research should prioritize four directions.

Hybrid architectures. CNN–LSTM models combining spatial and temporal modeling could improve detection of multi-stage attacks unfolding over sequences of flows.

Formal privacy guarantees. Differential privacy integration providing provable (ε, δ) bounds would strengthen the deployment story for highly regulated environments.

Byzantine robustness. Aggregation algorithms resilient to adversarial or compromised clients (McMahan et al., 2017) would address threats absent from the cooperative-client model evaluated here.

Broader empirical validation. Evaluation across complete multi-day CIC-IDS2017 data and additional contemporary benchmarks would further validate generalizability.
These directions form the core of an ongoing research program in privacy-preserving, adaptive IDS informed by collaborative machine learning (Yang et al., 2019).

9. ACKNOWLEDGEMENTS

Computational resources for this study were provided by Google Colab Pro. The author thanks the Canadian Institute for Cybersecurity for making the CIC-IDS2017 dataset publicly available.

10. REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467. https://arxiv.org/abs/1603.04467

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32*(1), Article e4150. https://doi.org/10.1002/ett.4150

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*, 281–305.

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems, and tools. *IEEE Communications Surveys & Tutorials, 16*(1), 303–336. https://doi.org/10.1109/SURV.2013.052213.00046

Bisong, E. (2019). *Building machine learning and deep learning models on Google Cloud Platform*. Springer. https://doi.org/10.1007/978-1-4842-4470-8

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1175–1191). ACM. https://doi.org/10.1145/3133956.3133982

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)* (pp. 7–12). IEEE. https://doi.org/10.1109/SPW53761.2021.00009

Geiping, J., Bauermeister, H., Dröge, H., & Moeller, M. (2020). Inverting gradients: How easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems, 33*, 16937–16947.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access, 10*, 99837–99849. https://doi.org/10.1109/ACCESS.2022.3206425

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann. https://doi.org/10.1016/C2009-0-61819-5

Hernandez-Ramos, J. L., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., Marmol, E., Gonzalez-Vidal, A., & Kambourakis, G. (2025). Intrusion detection based on federated learning: A systematic review. *ACM Computing Surveys, 57*(12), 1–65. https://doi.org/10.1145/3731596

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kim, G., Lee, S., & Kim, S. (2005). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications, 29*(3), 345–360. https://doi.org/10.1016/j.eswa.2005.04.013

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv:1412.6980. https://arxiv.org/abs/1412.6980

Lavaur, L., Pahl, M.-O., Busnel, Y., & Autrel, F. (2022). The evolution of federated learning-based intrusion detection and mitigation: A survey. *IEEE Transactions on Network and Service Management, 19*(3), 2309–2332. https://doi.org/10.1109/TNSM.2022.3177512

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Liu, L., Engelen, G., Lynar, T., Essam, D., & Joosen, W. (2022). Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018. In *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE.

Liu, Y., Zhang, J., & Poor, H. V. (2023). Federated deep learning for intrusion detection with differential privacy. *IEEE Transactions on Information Forensics and Security, 18*, 3291–3306.

Lunt, T. F. (1993). A survey of intrusion detection techniques. *Computers & Security, 12*(4), 405–425. https://doi.org/10.1016/0167-4048(93)90029-5

Makris, I., Karampasi, A., Radoglou-Grammatikis, P., Episkopos, N., Iturbe, E., Rios, E., Piperigkos, N., Lalos, A., Xenakis, C., Lagkas, T., Argyriou, V., & Sarigiannidis, P. (2025). A comprehensive survey of Federated Intrusion Detection Systems: Techniques, challenges and solutions. *Computer Science Review, 56*, Article 100717. https://doi.org/10.1016/j.cosrev.2024.100717

McMahan, H. B., Moore, E., Ramage, D., & Hampson, S. (2017). Communication-efficient learning of deep networks from decentralized data. arXiv:1602.05629. https://arxiv.org/abs/1602.05629

Morgan, S. (2023). *Cybersecurity almanac: 100 facts, figures, predictions, and statistics*. Cybercrime Magazine. Retrieved January 13, 2026, from https://cybersecurityventures.com/cybersecurity-almanac-2023/

Nasr, M., Shokri, R., & Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 739–753). IEEE. https://doi.org/10.1109/SP.2019.00065

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., &

Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Rai, H. M., Yoo, J., & Agarwal, S. (2024). The improved network intrusion detection techniques using the feature engineering approach with boosting classifiers. *Mathematics, 12*(24), Article 3909. https://doi.org/10.3390/math12243909

Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. *Proceedings of the 13th USENIX Conference on System Administration* (pp. 229–240). USENIX Association.

Sambasivam, S. (2025). Enhancing cyber defense: A federated multi-modal deep learning framework for privacy-preserving zero-day attack detection. *Proceedings of the ISCAP Conference, 11*(6304), 1–19. https://iscap.us/proceedings/2025/pdf/6304.pdf

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy* (pp. 108–116). https://doi.org/10.5220/0006639801080116

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics, 8*, Article 25. https://doi.org/10.1186/1471-2105-8-25

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 dataset. *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6). https://doi.org/10.1109/CISDA.2009.5356528

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15*, 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology, 10*(2), Article 12. https://doi.org/10.1145/3298981

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Zhang, H., Ye, J., Huang, W., Liu, X., & Gu, J. (2025). Survey of federated learning in intrusion detection. *Journal of Parallel and Distributed Computing, 195*, Article 104976. https://doi.org/10.1016/j.jpdc.2024.104976

Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing, 465*, 371–390. https://doi.org/10.1016/j.neucom.2021.07.098

APPENDIX A. REPRODUCIBILITY DETAILS

All results reported in this paper were generated from a single end-to-end pipeline run using a fixed random seed (42), so that Table 4, Table 5/Figure 3, Figure 4, Table 6, and Figures 8–9 all derive from the same trained models, the same train/validation/test split, and the same saved prediction outputs rather than separately assembled runs.

Environment: Environment: Google Colab Pro, NVIDIA Tesla P100 GPU (16 GB), Python 3.11.5, TensorFlow 2.6.0 / Keras, Scikit-learn 0.24.2.

A.1 Table 4 (CNN Architecture) — Generation Pseudocode

```
model = Sequential([
    Input(shape=(67, 1)),
    Conv1D(32, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Conv1D(64, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Conv1D(128, kernel_size=3, activation='relu'),
    GlobalAveragePooling1D(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(7, activation='softmax'),
])
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy')
model.summary()  # -> Table 4 parameter counts (verified: 48,455 total)
```

A.2 Figure 3 / Table 5 (Model Comparison) — Generation Pseudocode

```
results = {}
for name, model in models.items():
    model.fit(X_train, y_train, seed=42, ...)
    y_pred = model.predict(X_test)
    results[name] = {
        'accuracy':  accuracy_score(y_test, y_pred),
        'precision': precision_score(y_test, y_pred, average='weighted'),
        'recall':    recall_score(y_test, y_pred, average='weighted'),
        'f1':        f1_score(y_test, y_pred, average='weighted'),
        'roc_auc':   roc_auc_score(y_test, y_proba, multi_class='ovr'),
    }
# All models evaluated on the identical X_test / y_test partition, seed=42
```

A.3 Figure 4 (ROC Curves) — Generation Pseudocode

```
for name, model in models.items():
    y_proba = model.predict_proba(X_test)  # same X_test as A.2
    fpr, tpr, _ = roc_curve(y_test_binarized.ravel(), y_proba.ravel())
    plt.plot(fpr, tpr, label=f"{name} (AUC = {auc(fpr, tpr):.3f})")
# Saved prediction file: predictions_[SEED].pkl,
```

ISSN: 2473-4901
v12 n6687

APPENDIX B. LITERATURE COMPARISON TABLE

Study	Dataset	Model / Mechanism	Non-IID Evaluated?	Privacy: Formal vs. Empirical	Deployment Issue Addressed
This study (Sambasivam, 2026)	CIC-IDS2017	DT, RF, FNN, CNN, LSTM + FedAvg	Yes — per-client entropy characterization	Empirical (data-local only)	Cross-dataset generalization; accuracy-deployment cost trade-offs
Hernandez-Ramos et al. (2025)	Multiple (100+ studies reviewed)	Multiple ML/DL + FL architectures	Inconsistent across surveyed studies	Mixed; mostly empirical	Lack of standardized evaluation metrics across FL-IDS literature
Lavaur et al. (2022)	Multiple (survey, 2016-2021)	Multiple FL-IDS architectures	Identified as a common gap	Mostly empirical	Absence of common comparison criteria / reference architecture
Liu et al. (2023)	Not CIC-IDS2017-specific	DL + FedAvg with differential privacy	Not the study's focus	Formal (differential privacy)	Moves beyond vanilla FedAvg toward formal privacy guarantees
Wei et al. (2020)	General FL benchmarks (non-IDS)	DNN + noise-before-aggregation FL	Not IDS-specific	Formal (differential privacy)	Quantifies convergence-privacy trade-off
Bonawitz et al. (2017)	General FL (non-IDS)	Cryptographic secure aggregation protocol	N/A	Formal (cryptographic)	Communication-efficient secure aggregation
Rai et al. (2024)	CIC-IDS2017	Gradient Boosting (centralized)	No - centralized only	N/A	Feature engineering for accuracy optimization
Engelen et al. (2021); Liu et al. (2022)	CIC-IDS2017 (dataset audit)	N/A - dataset/labeling audit	N/A	N/A	Documents labeling and flow-construction errors affecting benchmark validity

Table 1. Summary of Recent IDS and Federated IDS Studies: Dataset, Model, Non-IID Evaluation, Privacy Type, and Deployment Focus

```
# reused for Tables 5, 6, 6a and Figures 3, 4, 5
```