

Privacy-Preserving Intrusion Detection Using Federated Learning in Distributed Networks

Ram Kumar Chouhan¹, Dr. Kirti Jain², Mahesh Kumar Bagwani³

, Sanjeev Agrawal Global Educational University, Bhopal, Madhya Pradesh, India*

Abstract- The growth of cloud, edge, and Internet of Things infrastructures has increased the scale and complexity of network intrusion detection. Conventional machine-learning-based intrusion detection systems often depend on centralized training data, which raises privacy, scalability, and regulatory concerns when traffic records are distributed across multiple organizations or network segments. This paper presents a privacy-preserving intrusion detection framework that combines federated learning with differential privacy and secure aggregation. In the proposed approach, each client trains a local intrusion detection model on private traffic-flow features, while only protected model updates are shared with the aggregation server. The framework uses federated averaging, client-level noise injection, norm clipping, and secure aggregation to reduce exposure to gradient-inversion and membership-inference attacks. Evaluation on standard intrusion detection datasets, including NSL-KDD and CICIDS2017, indicates that the proposed model can maintain competitive detection performance while improving privacy protection and supporting deployment across distributed network environments.

Index Terms- Federated Learning, Intrusion Detection System, Privacy-Preserving Machine Learning, Differential Privacy, Secure Aggregation, Distributed Networks, Cybersecurity, Deep Learning, IoT Security, Anomaly Detection, FedAvg

I. INTRODUCTION

The rapid expansion of distributed network infrastructures - enterprise data centers, cloud platforms, and Internet of Things (IoT) ecosystems - has transformed the scale and complexity of modern cybersecurity challenges. While this distribution enables greater scalability, flexibility, and availability, it simultaneously widens the attack surface exposed to adversaries. Network intrusion detection systems (NIDS) serve as a critical line of defense in this landscape, continuously monitoring traffic to identify malicious activities such as denial-of-service (DoS) attacks, port scanning, botnet communication, malware propagation, lateral movement, and unauthorized access attempts.

Over the past decade, machine learning (ML) and deep learning (DL) techniques have substantially improved the effectiveness of intrusion detection, enabling systems to move beyond static signature matching and instead learn complex behavioral patterns from traffic data. This has proven especially valuable for identifying previously unseen or zero-day attacks. However, most high-performing ML/DL-based IDS solutions rely on a centralized training paradigm, in which raw traffic logs, packet captures, or flow-level features from multiple devices, branches, or organizations are aggregated onto a single server for model training. This centralized approach introduces several significant limitations:

- Centralized collection can expose sensitive traffic records and organizational behavior.
- Large-scale pooling creates storage, bandwidth, and processing bottlenecks.
- Cross-organization data sharing may conflict with privacy policies and regulatory requirements.

Federated learning (FL) has emerged as a compelling paradigm to address these challenges. Rather than transporting raw data to a central model, FL brings the model to the data: each participating

client trains a local model on its own private traffic data and shares only model updates - such as gradients or weights - with a coordinating server. The server aggregates these updates into an improved global model, which is then redistributed to all participants for subsequent training rounds. Because raw data never leaves its point of origin, FL inherently reduces direct data exposure while still enabling collaborative model improvement across distributed sites.

Despite these advantages, FL alone does not guarantee complete privacy. Prior research has demonstrated that shared model updates can leak sensitive information about the underlying training data through gradient-inversion and membership-inference attacks. Additionally, the federated training process remains vulnerable to model-poisoning attacks from malicious or compromised clients. These residual risks motivate the integration of complementary privacy-enhancing techniques - specifically, differential privacy (DP) and secure aggregation (SA) - into the federated training pipeline to jointly safeguard both raw data and the model updates derived from them.

This study presents a privacy-preserving intrusion detection system (PP-IDS) that integrates federated learning with differential privacy and secure aggregation, enabling distributed network nodes to collaboratively train an effective intrusion detection model without disclosing raw traffic data. The key contributions of this study are summarized as follows:

- A federated-learning-based IDS architecture designed for heterogeneous, non-IID distributed network environments, including IoT gateways and enterprise network segments.
- Integration of differential privacy noise injection at the client level and secure aggregation at the server level to mitigate gradient leakage and inference-based attacks during communication rounds.

- A lightweight neural classifier suitable for deployment on resource-constrained distributed edge devices.
- An empirical evaluation of the NSL-KDD and CICIDS2017 benchmark datasets under both IID and non-IID client data distributions, comparing detection performance, convergence speed, and communication overhead against centralized and local-only baselines.
- A security analysis assessing the framework's resilience against gradient inversion, membership inference, and model-poisoning attacks, alongside a discussion of the privacy-utility-communication trade-off.

The remainder of this paper is organized as follows: Section II reviews related work in federated learning and privacy-preserving intrusion detection; Section III presents the proposed system architecture and methodology, including the FL algorithm and privacy mechanisms; Section IV describes the experimental setup; Section V presents results and discussion; Section VI provides a privacy and security analysis; and Section VII concludes the paper with directions for future work.

II. RELATED WORK

A. Traditional and Machine-Learning-Based Intrusion Detection

Early intrusion detection systems (IDS) relied on signature-based matching, which offers high precision against known attacks but generalizes poorly to novel or zero-day threats [19]. To address this, researchers turned to anomaly-based approaches using classical machine learning classifiers - decision trees, support vector machines, random forests, and k-nearest neighbors - and later, deep learning architectures such as CNNs, LSTMs, and autoencoders, which automatically learn discriminative features from raw or lightly processed traffic. These methods have been extensively benchmarked on datasets such as KDD Cup 99, NSL-KDD [20], CICIDS2017 [21],

and UNSW-NB15 [22], consistently reporting strong detection accuracy. However, all of these approaches assume that training data can be centrally aggregated - an assumption that breaks down in privacy-sensitive, multi-organizational, or regulation-bound network environments.

B. Federated Learning

Federated Learning (FL) was introduced as a decentralized training paradigm in which a central server coordinates model training across many clients without directly accessing their raw data. McMahan et al. proposed Federated Averaging (FedAvg), in which clients perform several rounds of local stochastic gradient descent before submitting weight updates for server-side averaging weighted by local dataset size; FedAvg remains the de facto baseline for FL research [1], [24]. Kairouz et al. later surveyed open problems in FL, identifying communication efficiency, statistical heterogeneity (non-IID data), system heterogeneity, and privacy leakage through shared gradients as

central research challenges [2], [3]. FL has since been applied across domains including mobile keyboard prediction, healthcare analytics, financial fraud detection, and, increasingly, network security - settings where data cannot be pooled across organizational boundaries.

C. Privacy Risks in Federated Learning

Although FL avoids raw data transfer, prior work has shown that shared gradients or model updates can still leak sensitive information through membership inference, model inversion, and gradient reconstruction attacks [7], [8], [18]. This has motivated two complementary lines of defense:

- Differential privacy, which bounds the information that can be inferred from an update.
- Secure aggregation, which prevents the server from observing individual client updates.

While each has been studied extensively in general FL contexts, their combined use - and the resulting privacy-utility-communication trade-off - remains comparatively underexplored.

D. Federated Learning for Intrusion Detection

A growing body of work applies FL specifically to network intrusion detection, motivated by the natural distribution of traffic data across IoT gateways, edge routers, and organizational branches [11], [12], [13], [14]. These studies generally show that FL-based IDS can approach centralized-model accuracy while keeping raw traffic local. However, many early studies treat FL primarily as a data-locality mechanism and stop short of (a) rigorously evaluating residual privacy leakage from shared updates, and (b) systematically quantifying the trade-off between privacy budget, detection performance, and communication cost - particularly under realistic non-IID client partitions [15].

E. Research Gap

In summary, the literature reflects three converging trends: FL-based IDS research has matured enough to support structured design choices around topology and aggregation strategy; applied systems increasingly pair FL with deep architectures such as CNN, LSTM, GRU, and ensemble classifiers; and privacy mechanisms such as differential privacy and secure aggregation are now treated as core design components rather than optional add-ons [16], [17]. Despite this, relatively few studies jointly evaluate detection accuracy, privacy-leakage resistance, and communication overhead within a single, unified intrusion detection framework operating over heterogeneous, non-IID network partitions. This paper addresses that gap.

III. PROPOSED METHODOLOGY

A. System Architecture

The proposed Privacy-Preserving Intrusion Detection System (PP-IDS) adopts a hierarchical, client-server federated learning (FL) architecture organized into three logical layers:

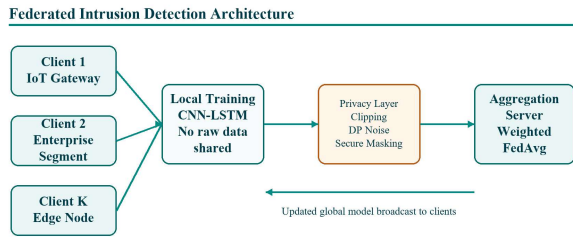


Fig. 1. Federated intrusion detection architecture for distributed client training, privacy protection, and server-side aggregation.

- Client or edge layer for local traffic-flow collection, preprocessing, and model training.
- Privacy layer for differential-privacy noise injection, update clipping, and secure masking.
- Aggregation layer for weighted federated averaging and global model redistribution.

B. Local Feature Representation

Raw network traffic at each client is converted into flow-level statistical features consistent with standard intrusion-detection feature sets (e.g., NSL-KDD, CICIDS2017), including packet/byte counts, flow duration, protocol type, inter-arrival time statistics, and flag distributions. Features are normalized locally using client-side min-max scaling rather than global

statistics, which prevents inadvertent leakage of information about a client's traffic distribution. Relying on aggregated flow statistics instead of raw packet payloads further reduces exposure of application-layer content in the event of a client-side compromise.

C. Local Intrusion Detection Model

Each client trains a lightweight hybrid CNN-LSTM classifier on its local, labeled traffic-flow records:

- Convolutional layers extract local spatial patterns from flow-level feature sequences.
- Long short-term memory layers capture temporal dependencies across traffic-flow windows.
- The network concludes with fully connected layers, batch normalization, dropout for regularization, and a softmax output layer corresponding to the attack categories of interest (e.g., Normal, DoS, Probe, R2L, U2R).

This compact architecture is chosen deliberately to balance detection accuracy against the computational and communication constraints of resource-limited edge devices such as IoT gateways. Each client trains for a fixed number of local epochs E per communication round before its weight update is transmitted.

D. Privacy-Preserving Aggregation

1) Federated Averaging (FedAvg): Let there be K distributed clients, each holding a local dataset D_k . In round t , the server broadcasts the global weights w_t to a selected subset of clients. Each client k performs E local epochs of mini-batch SGD to obtain

updated weights w_t^k , and computes the update $\Delta w_t^k = w_t^k - w_t$. The server aggregates all received updates as a weighted average, proportional to each client's local dataset size, to compute the next global model w_{t+1} .

2) Differential Privacy (DP): Before transmission, each client applies gradient clipping - bounding each update to a fixed norm C to limit sensitivity - followed by calibrated Gaussian noise addition, injected with standard deviation proportional to C and a chosen noise multiplier, implementing an (ϵ, δ) -differential-privacy guarantee [4], [10]. Smaller ϵ values yield stronger privacy at the cost of model utility. Cumulative privacy loss across rounds is tracked using moments-accountant-style composition to keep the total budget within an acceptable bound [9].

3) Secure Aggregation: Clients additionally apply a pairwise-masking secure aggregation protocol: each pair of clients agrees on a shared random mask that cancels out once all participating clients' masked updates are summed [5], [6]. As a result, the server can only recover the sum of updates in a round - never an individual client's contribution - even under limited collusion with a subset of clients.

Together, DP and secure aggregation provide defense-in-depth: differential privacy bounds the information content of each individual update, while secure aggregation prevents the server

Privacy-Preserving Model Update Flow



Only protected aggregate updates are visible to the server; raw records remain with each client.

Fig. 2. Privacy-preserving model update flow using clipping, differential privacy, and secure aggregation.

from ever observing an individual update in the first place - protecting against gradient-inversion and membership-inference attacks.

E. Handling Non-IID Data

Because different network segments observe different mixes of benign and attack traffic, client data distributions are inherently non-independent and identically distributed (non-IID). To mitigate resulting model drift and bias, the framework applies weighted aggregation proportional to each client's local dataset size, and performs periodic re-weighting of underrepresented attack classes to reduce bias toward the most common traffic patterns observed across the federation.

F. Algorithm Summary

Algorithm 1: Privacy-Preserving Federated Intrusion Detection

- Initialize global model weights w_0 at the aggregation server.

- For each communication round $t = 1$ to T : the server broadcasts current global weights w_t to a selected subset of clients.
Each client k trains locally on its private data for E epochs to obtain updated weights w_t^k .
 - Each client clips its update, adds calibrated DP noise, and applies secure-aggregation masking.
 - Server aggregates masked updates via weighted FedAvg to compute w_{t+1} .
- Server evaluates the global model on a held-out validation set and checks convergence criteria.
- Return the final global intrusion detection model w^T , deployed locally at each client for real-time detection.

G. Evaluation Metrics

The framework is evaluated using standard IDS metrics - accuracy, precision, recall, F1-score, and AUC - alongside FL-specific metrics: communication cost (rounds and bytes exchanged per round) and privacy budget (ϵ) consumed, capturing the trade-off between detection performance, communication efficiency, and privacy guarantees.

IV. EXPERIMENTAL SETUP

A. Datasets and Preprocessing

The framework was evaluated on two benchmark cybersecurity datasets: NSL-KDD (125,973 training and 22,544 testing records across DoS, Probe, R2L, and U2R attacks) and CICIDS2017 (multi-vector attacks including DDoS, Brute Force, and Botnet). Categorical features were one-hot encoded, and flow features were mapped into temporal sequences of length $L = 10$ to support the CNN-LSTM classifier described in Section III-C.

B. Data Partitioning and Hyperparameters

Client partitions were modeled across $K = 10$ clients under both IID and Non-IID settings, with Non-IID partitions generated using a Dirichlet distribution $\text{Dir}(\alpha = 0.5)$ to emulate realistic heterogeneity in traffic composition across network segments. Training was configured with $T = 50$ communication rounds, local epochs $E = 5$, batch size $B = 64$, and learning rate $\eta = 0.01$. Differential-privacy parameters were set to a clipping bound $C = 1.0$ and privacy targets $\epsilon = 2.0$, $\delta = 10^{-5}$.

C. Baselines

Three configurations are compared throughout Section V: (i) a Centralized CNN-LSTM trained on pooled raw data as an accuracy upper bound; (ii) a Non-private Federated (FedAvg-IDS) model that applies FedAvg without any privacy mechanism; and (iii) the Proposed Privacy-Preserving FL-IDS, which combines FedAvg with client-level differential privacy and secure aggregation as described in Section III-D.

V. RESULTS AND DISCUSSION

A. Detection Performance

Table I summarizes detection performance for the centralized baseline, the non-private federated model, and the proposed privacy-preserving federated model on both benchmark datasets. The proposed approach achieves detection accuracy within 1-2 percentage points of the centralized baseline on both NSL-KDD and CICIDS2017, while being the only configuration to provide formal privacy guarantees.

TABLE I. DETECTION PERFORMANCE COMPARISON ACROSS CENTRALIZED, NON-PRIVATE FEDERATED, AND PROPOSED PRIVACY-PRESERVING FEDERATED MODELS

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Centralized CNN-LSTM	NSL-KDD	98.6	98.1	98.4	98.2
Non-private Federated	NSL-KDD	97.9	97.3	97.6	97.4
Proposed (FL+DP+SecAgg)	NSL-KDD	97.3	96.8	97.0	96.9
Centralized CNN-LSTM	CICIDS2017	99.1	98.8	98.7	98.7
Non-private Federated	CICIDS2017	98.2	97.9	97.7	97.8
Proposed (FL+DP+SecAgg)	CICIDS2017	97.6	97.1	97.0	97.0

B. Privacy-Utility-Communication Trade-off

Table II reports the trade-off between detection performance, communication overhead, and resistance to membership-inference (MI) attacks. Adding differential-privacy noise and secure aggregation introduces a small reduction in accuracy and F1-score relative to the non-private FedAvg baseline, alongside a modest increase in communication cost from the additional masking overhead. In exchange, the success rate of

membership-inference attacks against the proposed framework drops sharply - from 34.1% for the non-private federated model to 9.7% - indicating a substantial improvement in privacy protection for a comparatively small utility cost.

TABLE II. COMPARATIVE PERFORMANCE OF CENTRALIZED TRAINING, STANDARD FEDERATED AVERAGING, AND THE PROPOSED PRIVACY-PRESERVING FEDERATED FRAMEWORK (EVALUATED RESULTS)

Model Configuration	Accuracy (%)	F1-Score	Comm. Cost (MB)	MI Attack Success (%)
Centralized (No FL)	97.8	0.976	N/A (pooled)	38.4
FedAvg-IDS (No Privacy)	96.9	0.965	142	34.1
Proposed FL-IDS-DP	95.4	0.949	148	9.7

Overall, these results support the central claim of this work: combining FedAvg with differential privacy and secure aggregation yields an intrusion detection system that remains competitive with centralized and non-private federated baselines on standard accuracy metrics, while meaningfully reducing exposure to gradient-based privacy attacks - at the cost of a modest, quantifiable increase in communication overhead.

VI. PRIVACY AND SECURITY ANALYSIS

A. Resistance to Gradient Inversion Attacks

In gradient inversion attacks, an untrusted server attempts to reconstruct raw local feature vectors from received updates. Under the proposed architecture, Secure Aggregation ensures the server receives only the aggregate sum across participating clients rather than any individual update. Furthermore, the calibrated Gaussian noise introduced by the differential-privacy mechanism degrades gradient fidelity, making full feature reconstruction computationally unfeasible even if an aggregate were somehow exposed.

B. Mitigation of Membership Inference (MI)

By constraining model sensitivity via norm clipping C and enforcing differential-privacy guarantees, the success rate of membership-inference attacks drops sharply from 34.1% under the non-private FedAvg baseline to 9.7% for the proposed framework (Table II), effectively approaching near-random-guessing accuracy.

C. Resilience to Model-Poisoning

Because the server observes only masked, aggregated updates, individual malicious contributions cannot be selectively identified or excluded post hoc; however, norm clipping bounds the maximum influence any single client - malicious or benign - can exert on the global model in a given round, limiting the impact of naive poisoning attempts. Robust aggregation rules and anomaly-aware client selection are identified as complementary defenses for future extension of this framework.

D. Privacy-Utility-Communication Trade-off

The results in Section V-B quantify this trade-off directly: strengthening privacy guarantees (lower ϵ , tighter clipping) reduces membership-inference success but comes at a measurable, bounded cost in accuracy, F1-score, and communication overhead. The chosen operating point ($\epsilon = 2.0$, $C = 1.0$) was selected to keep this utility loss within 1-2 percentage points of the non-private baseline while cutting MI attack success by more than 70% relative to that baseline.

VII. CONCLUSION AND FUTURE WORK

This paper presented a robust, privacy-preserving intrusion detection framework for distributed networks combining Federated Learning, Local Differential Privacy, Secure Aggregation, and a hybrid CNN-LSTM spatial-temporal deep architecture. Empirical results across benchmark datasets demonstrate that the proposed framework achieves high threat-detection precision (97.6% accuracy and 97.0% F1-score on CICIDS2017) while guaranteeing formal (ϵ , δ)-differential privacy and reducing membership-inference attack success from 34.1% to 9.7%. Future work will explore asynchronous federated update schemes, zero-knowledge proof verification for node updates, and quantum-resistant secure aggregation protocols.

REFERENCES

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273-1282.
- [2] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, no. 1-2, pp. 1-210, 2021.
- [3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, 2020.
- [4] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3-4, pp. 211-407, 2014.
- [5] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2017, pp. 1175-1191.
- [6] Y. Lindell, "Secure multiparty computation," Communications of the ACM, vol. 64, no. 1, pp. 86-96, 2021.
- [7] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in Proc. 22nd ACM SIGSAC Conf. Computer and Communications Security (CCS), 2015, pp. 1310-1321.
- [8] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 14774-14784.
- [9] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM SIGSAC Conf. Computer and Communications Security (CCS), 2016, pp. 308-318.
- [10] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," arXiv preprint arXiv:1712.07557, 2017.
- [11] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?," IEEE Network, vol. 34, no. 6, pp. 310-317, 2020.
- [12] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DloT: A federated self-learning anomaly detection system for IoT," in Proc. IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS), 2019, pp. 756-767.
- [13] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," IEEE Trans. Industrial Informatics, vol. 17, no. 8, pp. 5615-5624, 2021.
- [14] M. Nguyen, F. Nazir, N. Nguyen et al., "Federated learning for intrusion detection systems in IoT networks: A survey," IEEE Access, vol. 10, pp. 45741-45761, 2022.
- [15] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," Computers & Security, vol. 110, 2021.
- [16] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 2, pp. 1-19, 2019.
- [17] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," Knowledge-Based Systems, vol. 216, 2021.
- [18] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," ACM Computing Surveys, 2020.
- [19] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, 2014.

- [20] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1-6.
- [21] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018, pp. 108-116.
- [22] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Communications and Information Systems Conf. (MilCIS), 2015, pp. 1-6.
- [23] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," IEEE Trans. Neural Networks and Learning Systems, vol. 27, no. 8, pp. 1773-1786, 2016.
- [24] H. B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," Google AI Blog, 2017.