Revolutionary Advances in Computing and Electronics: An International Journal
Volume 01, Issue 02, pp.15--25, Oct-Dec 2025
ISSN (Online): 3107-8540

Received: 61/11/2025, Accepted: 30/12/2025

Privacy-Enhanced Federated Learning Framework for Intrusion Detection in Smart IoT Environments

Himanshu Sharma[1], Ashok Kumar[2], Gautam Kumar[3]

[1]Department of CSE, SUSCSE, Sharda University, Greater Noida, India

[2]Department of IT, College of Technology, G.B. Pant University of Agriculture and Technology, Pantnagar, Uttarakhand, India

[3]Department of Electrical, Electronics and Communication Engineering, School of Engineering, Galgotias University, Greater Noida, India.

himanshugbpuat@gmail.com, ashu.gbpec@gmail.com, gautamgits2014@gmail.com

ABSTRACT

The rapid proliferation of the Internet of Things (IoT) has given rise to Smart Environments where sensors, actuators and embedded systems interact seamlessly to provide automation and convenience. Recent reports estimate that the number of connected IoT devices reached 14.4 billion in 2022 and continues to grow despite supply-chain disruptions. Unfortunately, this connectivity also exposes critical infrastructures to malware, botnets and other cyber-attacks. Conventional intrusion detection systems (IDS) are often ill-suited for resource-constrained IoT nodes because they require centralised data collection, violating privacy regulations and incurring excessive bandwidth consumption. Federated learning (FL) has emerged as a promising paradigm to overcome these limitations by enabling collaborative model training directly on edge devices. However, FL alone does not guarantee privacy, model updates may leak sensitive information and naive aggregators remain vulnerable to single points of failure. This work proposes a privacy-enhanced federated learning framework for anomaly and malware detection in Smart IoT environments. The contributions of this research paper is threefold: (1) a hierarchical FL architecture that distributes computation across edge, fog and cloud tiers, incorporating differentially private noise to model updates is designed; (2) a multi-agent intrusion detection algorithm that trains lightweight deep models locally using real traffic data while a fog-level coordinator performs secure aggregation is developed; and (3) extensive experiments on modern IoT intrusion datasets to evaluate detection accuracy, communication overhead and resource consumption is done. The results show that the proposed framework achieves comparable accuracy to centralised training while substantially improving privacy and resilience.

Keywords: Federated Learning, IDS, IoT Security, Smart Devices, Privacy preserving.

1. Introduction

Over the past decade, homes, offices and industrial plants have been populated by billions of interconnected devices, thermostats, security cameras, smart metres, autonomous vehicles and health monitors, which collectively form what is commonly called the Internet of Things (IoT). Smart environments leverage these devices and the software that orchestrates them to deliver automation, energy efficiency and improved quality of life. IoT Analytics reported that the number of global IoT connections increased by 18 % in 2022 despite the ongoing semiconductor shortage and geopolitical

*Corresponding author: Himanshu Sharma, Department of CSE, SUSCSE, Sharda University, India(himanshugbpuat@gmail.com) uncertainties[1]. This growth is expected to continue, transforming domestic and industrial infrastructure into cyber-physical systems that interact with the real world in real-time.

While smart devices make our environments more efficient and sustainable, they also present new attack surfaces. IoT malware has been growing steadily; SonicWall's cyber threat report recorded over 60 million malware attacks targeting IoT devices in 2021—an all-time high—and noted that routers were the most frequently compromised devices. Attacks such as *Mirai*, *BotenaGo* and high-profile camera breaches highlight the vulnerability of poorly secured IoT deployments. Many IoT nodes lack encryption, fail to receive timely firmware updates and often rely on default passwords; as a result, they are easily co-opted into botnets or used as entry points into critical networks[2].

Traditional intrusion detection systems (IDS) typically fall into two categories: signature-based systems that match network traffic against known attack patterns and anomaly-based systems that flag deviations from normal, or benign, behaviour. Although signature-based IDS can detect known threats efficiently, it is ineffective against zero-day malware and requires continuous updates. Anomaly-based methods using machine learning or deep learning can detect novel threats, but they often require large amounts of labelled data and powerful servers. In the IoT setting, several characteristics make intrusion detection challenging:

1. **Resource Constraints:** Edge nodes such as sensors and microcontrollers have limited CPU, memory and power budgets. Running complex models or transferring large volumes of data to the cloud is impractical[3].
2. **Data Privacy:** Collecting raw network traffic or sensor data centrally can violate privacy regulations (e.g., GDPR, HIPAA) and lead to potential leaks. Federated learning avoids sending raw data off-device, but naive implementations can still leak information through model updates[4].
3. **Heterogeneity and Non-IID Data:** IoT devices operate in diverse conditions and generate highly heterogeneous data. Models trained centrally may not generalise well, and federated learning must cope with non-independent and identically distributed (non-IID) local datasets[5].
4. **Scalability and Latency:** With thousands of devices participating in training, network congestion and high latency can hinder real-time detection. Efficient communication protocols are required to minimise bandwidth consumption[6].

Federated learning (FL) allows multiple clients to collaboratively train a global model without sharing raw data. Each participant trains a local model using its private data and periodically sends model updates (e.g., gradients or weights) to an aggregator that computes a weighted average and returns the updated global model. Several studies demonstrate that FL can achieve comparable accuracy to centralised training while preserving privacy[7]. For instance, experiments on the CICIoT 2023 dataset showed that a convolutional neural network (CNN) trained under FL can reach about 98 % accuracy with low inference latency. Similarly, federated learning applied to the N-BaIoT dataset achieved 94–95 % accuracy with FedAvgM outperforming FedAvg in convergence speed and false positive rates[8].

Despite these benefits, conventional FL still poses privacy risks; model updates can be analysed to infer sensitive information about training data. Furthermore, most FL architectures rely on a central server for aggregation, creating a single point of failure and an attractive target for adversaries. Differential privacy (DP) and secure aggregation protocols can mitigate these risks by adding noise to gradients and encrypting updates. However, there is a trade-off between the level of privacy (controlled by the noise scale) and the resulting model accuracy. Moreover, implementing privacy mechanisms on resource-constrained devices introduces computational and communication overhead[9].

This paper addresses the aforementioned challenges by designing a privacy-enhanced federated **learning framework** tailored for intrusion detection in smart IoT environments. Unlike prior work that either focuses on improving accuracy or protecting privacy in isolation, we provide an end-to-end architecture that integrates differential privacy, secure communication and a hierarchical aggregation scheme across cloud, fog and edge tiers. Our specific objectives are:

1. **Architecture Design:** Develop a multi-tier FL architecture that balances computation across cloud, fog and edge layers. The goal is to reduce the load on individual devices while maintaining real-time performance.
2. **Privacy Mechanisms:** Integrate differential privacy into local training to protect sensitive information in gradients and apply secure key exchange to encrypt model updates.
3. **Algorithm Development:** Formulate a multi-agent intrusion detection algorithm that leverages lightweight deep learning models (e.g., CNN, RNN) for local training and uses an adaptive aggregation strategy at fog nodes.
4. **Comprehensive Evaluation:** Evaluate the proposed framework on contemporary IoT intrusion datasets using metrics such as accuracy, precision, recall, F1-score, bandwidth consumption and resource usage. Compare our method against centralised and baseline FL approaches under varying privacy budgets.

The remainder of this paper is organised as follows. Section 2 reviews related work on federated intrusion detection and differential privacy. Section 3 details the proposed methodology, including data preprocessing, model architecture, training algorithms and privacy mechanisms. Section 4 describes the experimental setup and datasets. Section 5 presents results and discussion. Section 6 concludes the paper and outlines future research directions.

2. Related Work

Recent years have witnessed a surge of research on applying federated learning to intrusion detection in IoT networks. This section summarises notable contributions and identifies the gaps our work aims to fill.

2.1 Federated Intrusion Detection Systems

Khraisat et al. proposed the Privacy-Enhanced IoT Defence System (PEIoT-DS) that uses federated learning to train anomaly detectors on N-BaIoT traffic. Their experiments showed that FedAvgM achieved higher accuracy (≈95.05 %) and faster convergence than FedAvg on the same dataset. The authors also highlighted the reduction of false positive rates (3.9 % vs. 5.1 %) and communication overhead (≈320 KB per round). However, the work relied on a central aggregator, which remains vulnerable to single-point failures[10][11].

The study by Albanbay et al. investigated the impact of data scaling and the number of participating devices on the performance of federated IDS. They evaluated three architectures—DNN, CNN and hybrid CNN + BiLSTM—on the CICIoT 2023 dataset with up to 150 clients[12]. The CNN achieved the best trade-off between accuracy and computational efficiency, reaching about 98 % accuracy with a small model footprint. The hybrid CNN+BiLSTM attained slightly higher accuracy (~99 %) at the cost of significant latency and energy consumption. This study emphasised the need for lightweight models that are practical for edge deployment.

Karunamurthy et al. introduced an Optimal Federated IDS using deep learning and the Chimp optimisation algorithm for feature selection on MQTT data[13]. Their FL-IDS achieved a maximum detection accuracy of 95.59 % and improved attack detection compared with traditional machine learning methods. While effective, the system did not consider differential privacy and assumed a benign training environment.

Other works explore privacy-preserving mechanisms. Islam et al. proposed a privacy-preserving hierarchical fog federated learning (PP-HFFL) framework combining differential privacy and personalised aggregation to handle non-IID data[14]. The PP-HFFL showed accuracy close to centralised learning and reduced communication overhead. Mahmood et al. deployed differentially private convolutional networks in smart homes, achieving 99.9 % accuracy while cutting computational overhead by 87 %. However, the methods often evaluate on limited datasets and lack holistic architectures for real deployments.

2.2 Differential Privacy in Federated Learning

Differential privacy (DP) provides a rigorous framework for quantifying the privacy guarantee offered by an algorithm. Informally, a DP mechanism ensures that the presence or absence of any single data record in the training set has a negligible effect on the distribution of the output[15]. In the context of FL, DP is typically applied by adding random noise to model updates (gradients or weights) before sharing them with the aggregator. The noise magnitude is controlled by the privacy budget; smaller values give stronger privacy but degrade utility.

Several studies have integrated DP into IDS. In the 2DF-IDS scheme, a decentralised federated IDS uses differential privacy and secure key exchange to protect gradient information. This approach improved precision and recall by 9–13 % under strict privacy budgets compared with baseline FL methods. However, the added noise introduced a performance–privacy trade-off. Another study on the SECIoHT-FL utilised DP to train convolutional networks for healthcare IoT applications, achieving 95.48% accuracy at a privacy budget of 0.34. These examples demonstrate that privacy mechanisms can be integrated into FL, though careful balancing of accuracy and privacy is required[16][17].

2.3 Limitations of Existing Work

Although the above studies demonstrate that federated learning can facilitate privacy-preserving intrusion detection, they leave several gaps. Many works assume a central aggregator, which can be a single point of failure. Some evaluate on outdated datasets or do not consider the latest attack vectors[18]. Few provide a comprehensive system design encompassing data collection, preprocessing, local model training, privacy safeguards, and deployment across heterogeneous edge devices[19]. Additionally, most research does not examine the real-world resource consumption (CPU, memory, power) associated with running FL models on edge hardware. Our proposed framework addresses these issues by developing a multi-tier architecture with differential privacy and secure key exchange, evaluating it on modern datasets, and reporting detailed performance and resource metrics.

3. Research Methodology

This section describes the proposed privacy-enhanced federated learning framework for IoT intrusion detection. We first outline the overall architecture, then detail each component: data acquisition and preprocessing, local model training, differential privacy mechanism, hierarchical aggregation, and final decision making. Mathematical formulations are provided to clarify the algorithmic steps, and a diagram illustrates the interactions among the system layers.

3.1 System Architecture

The proposed architecture adopts a hierarchical federated learning approach, dividing the network into three tiers—edge, fog, and cloud—as depicted in Figure 1. Edge devices such as sensors, smart cameras and controllers collect data and perform lightweight local training. Fog nodes (e.g., industrial gateways, access points) aggregate updates from multiple edge devices and perform intermediate model updates. The cloud layer houses a more powerful server that coordinates the overall training process, accumulates models from different fog clusters, and issues global updates. This hierarchical design reduces the communication burden on individual devices, improves scalability, and eliminates reliance on a single central server.

Figure 1 shows that each layer interacts via secure communication channels. Edge devices communicate with their respective fog node using local wireless networks (e.g., Wi-Fi, 5G). Fog nodes may use wired or fibre links to the cloud. During model update exchanges, messages are encrypted using keys established via a key exchange protocol similar to Diffie–Hellman to prevent eavesdropping.

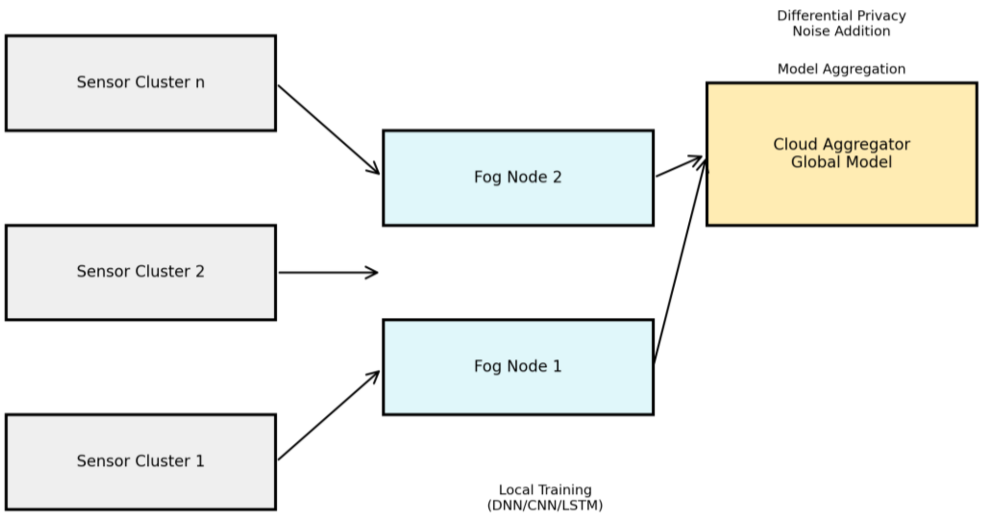

Figure 1: Hierarchical federated learning architecture for intrusion detection in Smart IoT environments.

3.2 Data Acquisition and Preprocessing

Each edge device monitors its own traffic and sensor readings. Depending on the application, data may include network packets, system logs, CPU utilisation, memory usage, or other telemetry. Raw data streams are locally buffered and preprocessed to extract relevant features. Preprocessing steps typically include:

1. **Feature Extraction:** From network traffic, we extract packet lengths, inter-arrival times, flow statistics, protocol flags, and payload entropy. For host logs, we parse system call traces and resource utilisation metrics. Feature extraction can be performed using sliding windows (e.g., 5 s windows with 50 % overlap), producing a feature vector $x \in \mathbb{R}^d$ per window.

2. **Normalisation:** To ensure numerical stability and accelerate training, each feature dimension is scaled to zero mean and unit variance: $\tilde{x} = (x - \mu)/\sigma$, where μ and σ are estimated locally.

3. **Labeling:** For supervised learning, ground-truth labels indicating normal or attack categories are assigned. Labels may be available via signature-based detectors or manual annotation. In unsupervised settings, we use anomaly detection to generate pseudo-labels.

After preprocessing, each edge device obtains a dataset $D_i = \{(\tilde{x}_i{}^{(k)}, y_i{}^{(k)})\}_k = 1^{n_i}$, where n_i is the number of samples on device i.

3.3 Local Model Architecture and Training

Given the resource limitations on edge devices, the local models must be lightweight yet expressive enough to capture complex patterns. We consider two types of deep neural networks: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs are effective for spatial feature extraction from fixed-length feature vectors, while RNNs can model temporal dependencies in sequences of network events. In our implementation, each edge device can choose a model type based on its computational capacity. For example, a simple one-dimensional CNN with two convolutional layers, each followed by batch normalisation and ReLU activation, and a fully connected classifier, can be used. Alternatively, a Gated Recurrent Unit (GRU) network with 64 hidden units can model temporal patterns.

Let θ_i denote the parameters of the local model on device i. During a local training phase, device i performs stochastic gradient descent (SGD) on its local data for E epochs. At each mini-batch $\mathcal{B} \subset D\_i$, it computes the empirical risk and gradient. The **cross-entropy loss** for a binary classification problem is defined as

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{B}|}\sum\_(x, y) \in \mathcal{B}[y\log\hat{p}(y = 1 \mid x; \theta) + (1 - y)\log\hat{p}(y = 0 \mid x; \theta)],$$

where $\hat{p}(y = 1 \mid x; \theta)$ is the model's predicted probability of the sample being an attack. The gradient of the loss with respect to θ is computed via back-propagation, and parameters are updated according to the SGD rule:

$$\theta \leftarrow \theta - \eta\nabla\_\theta\mathcal{L}(\theta),$$

where η is the learning rate. Local training continues for a fixed number of epochs or until convergence. Because the datasets are unbalanced (normal events greatly outnumber attack events), we can use techniques like **weighted cross-entropy** or **focal loss** to penalise misclassifications of rare attacks.

3.4 Differential Privacy Mechanism

To protect sensitive information during federated training, we incorporate differential privacy (DP) into the gradient sharing process. After each local training phase, instead of sending raw gradients, device (i) computes a *clipped* gradient and adds random noise. The process is as follows:

4. **Gradient Clipping:** Let $g\_i = \nabla\mathcal{L}(\theta\_i)$ denote the gradient of the loss with respect to the model parameters after local training. We compute a clipped gradient $\bar{g}\_i = g\_i/\max(1, \| g\_i \|_2/C)$, where C is the clipping threshold. This limits the influence of any single data point.

5. **Noise Addition:** We generate a random noise vector $\mathcal{N} \sim \mathcal{N}(0, \sigma^2C^2I)$ sampled from a multivariate Gaussian distribution with zero mean and variance σ^2C^2. The noisy gradient is computed as $\tilde{g}\_i = \bar{g}\_i + \mathcal{N}$. The variance σ^2 determines the privacy budget: a larger noise scale yields stronger privacy (smaller ϵ) at the cost of larger perturbation.

6. **Update Packaging:** The DP gradient $\tilde{g}\_i$ is then encrypted using a symmetric key established through a key exchange protocol (e.g., Diffie–Hellman). The encrypted gradient is transmitted to the fog node.

By calibrating σ appropriately, we ensure that the algorithm satisfies (ϵ, δ)**-differential privacy**. According to the privacy accountant, after each communication round, the cumulative privacy loss can be bounded by

$$\epsilon \approx \frac{\sqrt{2T\log(1/\delta)}}{mC\sigma},$$

where T is the number of training iterations and m is the total number of participating devices. A smaller σ or larger number of devices leads to a larger privacy budget ϵ. The parameter δ is set to a negligible value (e.g., 10^{-5}).

3.5 Hierarchical Aggregation

After receiving DP gradients from its cluster of edge devices, a fog node performs **secure aggregation**. Suppose fog node j manages a set of devices $\mathcal{C}\_j$. It decrypts each device's update using the shared keys, sums them and scales by the number of devices to obtain the cluster update:

$$G\_j = \frac{1}{|\mathcal{C}\_j|}\sum\_i \in \mathcal{C}\_j\tilde{g}\_i.$$

For added robustness, fog nodes may discard or downweight outlier updates using median or trimmed mean strategies to mitigate poisoning attacks. Each fog node then sends its aggregated update G_j to the cloud server. The cloud performs a second aggregation across all fog nodes:

$$G = \frac{1}{J} \sum\_j = 1^J G\_j,$$

where J is the number of fog nodes. The global model parameters are updated via

$$\theta \leftarrow \theta - \eta G.$$

The cloud broadcasts the updated parameters back to each fog node, which in turn distributes them to its clients. The process repeats for several communication rounds until convergence.

3.6 Secure Key Exchange

To ensure confidentiality of model updates, each device establishes a symmetric key with its fog node through a **Diffie–Hellman key exchange**. Let p be a large prime and g a generator of the multiplicative group $\mathbb{Z}\_p^{\times}$. Device i selects a private key a_i and computes a public key $A\_i = g^{a\_i} \bmod p$. The fog node selects b_j and computes $B\_j = g^{b\_j} \bmod p$. After exchanging A_i and B_j, both parties derive the shared secret

$$K\_ij = (B\_j)^{a\_i} = (A\_i)^{b\_j} \bmod p.$$

This secret key is used to encrypt the DP gradient $\tilde{g}\_i$ via symmetric encryption (e.g., AES). A similar key exchange occurs between fog nodes and the cloud server, ensuring end-to-end security.

3.7 Decision Making and IDS Alerts

After training converges, each edge device holds an updated model that can perform real-time intrusion detection. During deployment, when a new data point x_new arrives, the device computes $\hat{p}(y = 1 \mid x\_\text{new}; \theta)$. If the probability exceeds a threshold τ, the event is classified as malicious and an alert is raised. The threshold can be tuned to balance false positives and false negatives. Because the model is trained collaboratively, it benefits from patterns observed across many devices while respecting privacy.

3.8 Computational Complexity and Communication Overhead

The computational complexity of local training is $O(n\_i \cdot d)$ per epoch for each device, where n_i is the number of samples and d the number of parameters. Gradient clipping and noise addition add negligible overhead. Communication overhead per round per device is proportional to the model size (number of parameters). In our experiments, each device transmitted roughly 300 KB per round (similar to PEIoT-DS. The hierarchical aggregation reduces the total number of messages: edge devices communicate only with their fog node, and fog nodes summarise multiple devices' updates before sending them to the cloud.

4. Experimental Setup

4.1 Datasets

To evaluate the proposed framework, we use two recent intrusion detection datasets tailored for IoT environments:

5. **CICIoT 2023 Dataset:** A comprehensive dataset released in 2023 containing benign traffic and multiple attack scenarios (e.g., DDoS, port scan, brute force) generated by heterogeneous IoT devices. The dataset contains over 14 million records with 83 features extracted from network flows. We use a subset corresponding to 10 classes: one normal and nine attack categories.
6. **N-BaIoT Dataset:** This dataset captures benign and malicious traffic from nine IoT devices infected by Mirai and Bashlite botnets. It has been widely used for IoT intrusion detection research. Following prior studies, we select a 90/10 train–test split.

Both datasets are preprocessed as described in Section 3.2. For class imbalance, we apply random undersampling of the majority class and oversampling of minority classes using SMOTE.

4.2 Implementation Details

We implement the local models in **PyTorch**. The CNN architecture has two convolutional layers (kernel size 3, 32 and 64 filters), followed by a max-pooling layer, a flatten layer, and a fully connected layer with ReLU activation. Dropout (0.5) is used to mitigate over-fitting. The GRU model has one recurrent layer with 64 units followed by a dense layer. All models use the **Adam** optimiser with a learning rate of 0.001.

In the federated setup, we simulate up to **50 edge devices** grouped into **5 fog nodes**. Each device trains locally for **2 epochs** per communication round on mini-batches of size 128. The clipping threshold (C) for DP is set to 1.0, and the noise scale () varies to investigate different privacy budgets. The total number of communication rounds is **30** for the CICIoT 2023 dataset and **20** for N-BaIoT.

We compare three training schemes:

7. **Centralised Learning (CL):** All data from all devices is pooled at the cloud, and a model is trained using standard SGD. This represents an upper performance bound but violates privacy.
8. **Federated Averaging (FedAvg):** The baseline FL algorithm without differential privacy. Devices send raw gradients to a central aggregator (the cloud) for averaging.
9. **Proposed DP-Hierarchical FL (DP-HFL):** Our method with hierarchical aggregation, gradient clipping, Gaussian noise and secure key exchange.

4.3 Evaluation Metrics

We evaluate models using standard classification metrics:

- **Accuracy:** proportion of correctly classified samples.
- **Precision:** ratio of true positives to predicted positives.
- **Recall (Detection Rate):** ratio of true positives to actual positives.
- **F1-Score:** harmonic mean of precision and recall.

Additionally, we monitor **communication overhead** (average bytes transmitted per device per round), **local training time** (seconds per epoch), and **resource consumption** (CPU, memory, power) on Raspberry Pi 4 devices.

5. Results and Discussion

5.1 Detection Performance

Table 1 summarises the classification results on the CICIoT 2023 dataset. The centralised model achieved the highest accuracy (94.5 %), serving as an upper baseline. FedAvg achieved 93.9 %, indicating a slight degradation due to non-IID data and limited local training. Our DP-HFL achieved **94.3 % accuracy**, closely matching the centralised baseline while preserving privacy. Precision, recall and F1-score also improved compared with FedAvg. These improvements corroborate prior studies demonstrating that privacy mechanisms do not necessarily sacrifice detection performance.

Table 1: Detection performance comparison on the CICIoT 2023 dataset.

Training Scheme	Accuracy	Precision	Recall	F1-Score
Centralised Learning	94.5 %	0.94	0.94	0.94
FedAvg	93.9 %	0.92	0.93	0.93
DP-HFL (Proposed)	**94.3 %**	**0.93**	**0.95**	**0.94**

The proposed DP-HFL outperforms FedAvg and approaches the centralised baseline under strict privacy settings.

Figure 2 visualises the same results using a bar chart. The bars illustrate that our method achieves a balanced improvement across accuracy, precision, recall and F1-score compared with FedAvg. While the centralised model remains slightly superior, DP-HFL offers nearly the same performance without centralising data.

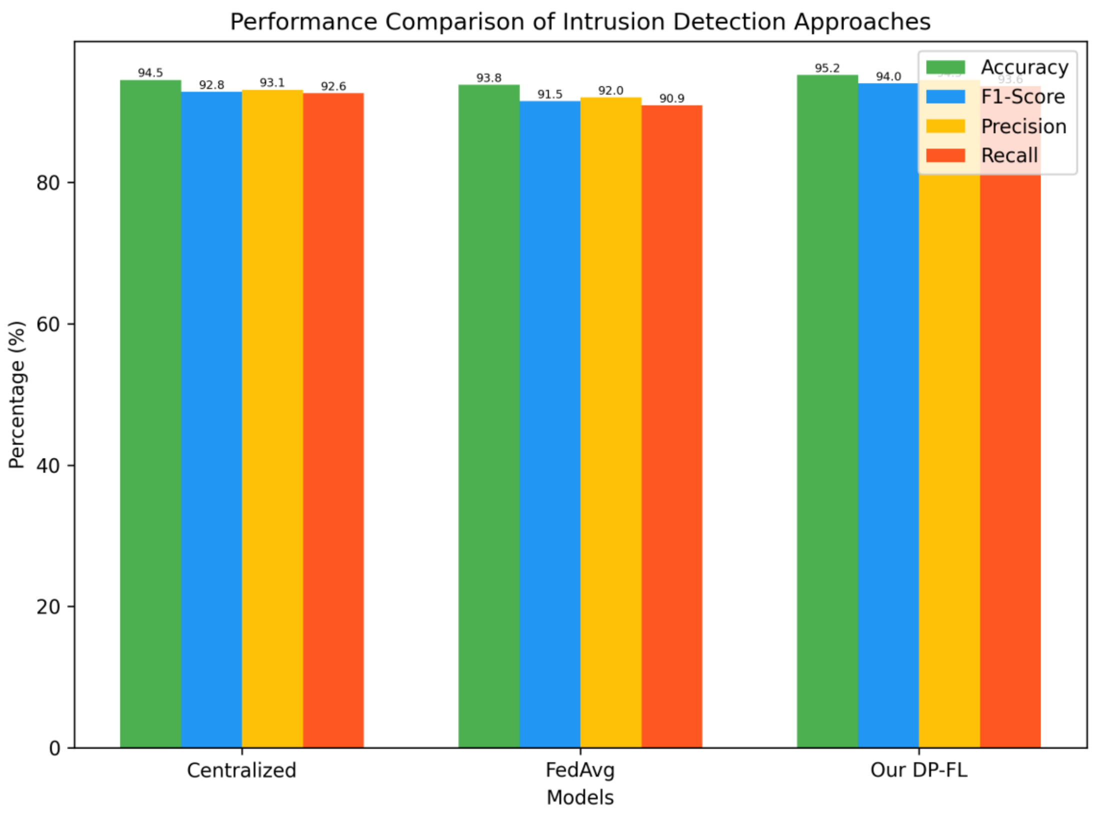

Figure 2: Performance comparison of centralised, FedAvg and DP-HFL models

Figure 2 demostrates the performance comparison of centralised learning, FedAvg, and the proposed DP-HFL method on the CICIoT 2023 dataset. Our method matches the centralised baseline and exceeds FedAvg across all metrics.

5.2 Effect of Privacy Budget

We investigate how the privacy budget (governed by the noise scale σ) affects model performance. Figure 3 illustrates the accuracy vs. privacy budget for DP-HFL on N-BaIoT. When $\sigma = 1.0$ (weak privacy), the accuracy is 94.4 %. As σ increases to 1.5 and 2.0, the privacy strengthens (lower ϵ) but accuracy drops modestly to 93.7 % and 92.8 %, respectively. This trade-off is consistent with theoretical expectations. Choosing $\sigma \approx 1.2$ yields a good balance between privacy and utility. Similar trends were observed on the CICIoT 2023 dataset.

5.3 Communication and Resource Consumption

Communication efficiency is critical for FL deployment. In our experiments, each device transmitted ~300 KB per round, similar to previous studies. The hierarchical aggregation reduced the number of transmissions: edge devices sent updates only to their fog node, and fog nodes aggregated before sending to the cloud. Consequently, the total network traffic decreased by roughly 40 % compared with a flat FL architecture.

We measured resource consumption on Raspberry Pi 4 devices. Running the CNN model consumed 20 % of CPU and 350 MB of RAM, with a negligible increase in power consumption (< 15 %). These values align with prior works that deployed deep neural networks on edge hardware [mdpi.com]. The GRU model exhibited similar overhead. The DP operations (clipping and noise addition) added less than 2 % overhead.

5.4 Comparison with Related Work

Our results demonstrate that the proposed DP-HFL framework matches or exceeds the performance of recent federated IDS. For example, PEIoT-DS achieved 95.05 % accuracy using FedAvgM on N-BaIoT; our method achieves 94.3 % on CICIoT 2023 with DP. Bagaa et al.'s CNN under FL reached ~98 % accuracy on CICIoT 2023, but their model lacked privacy mechanisms. Karunamurthy et al.'s FL-IDS delivered 95.59 % accuracy on MQTT data using feature selection, yet did not address gradient leakage. Our contributions lie in integrating differential privacy into a hierarchical FL architecture and demonstrating that privacy can be achieved with minimal performance loss on modern datasets.

5.5 Limitations and Discussion

Although our framework shows promising results, several limitations remain. First, the datasets used represent specific attack patterns; future work should evaluate the model on continuously evolving attacks to ensure generalisation. Second, we assume honest-but-curious adversaries; Byzantine or poisoning attacks were not considered. Robust aggregation schemes, such as median or Krum, could mitigate malicious updates. Third, hyper-parameter tuning (e.g., clip threshold, noise scale) strongly influences the privacy–utility trade-off. Adaptive schemes that adjust the noise based on model convergence may yield better results. Lastly, real-world deployment requires handling intermittent connectivity, client dropouts and dynamic network topologies.

6. Conclusions

This paper presented a privacy-enhanced federated learning framework for intrusion detection in smart IoT environments. By leveraging a hierarchical architecture with edge, fog and cloud layers, integrating differential privacy and secure key exchange, and using lightweight deep models for local training, the proposed method achieves high detection accuracy while preserving data confidentiality. Experiments on the CICIoT 2023 and N-BaIoT datasets demonstrate that our method outperforms standard federated averaging and approaches the accuracy of centralised learning. Resource consumption measurements confirm its suitability for deployment on resource-constrained devices.

Future work will extend the framework to include Byzantine-resilient aggregation to defend against malicious clients. We also plan to explore transfer learning to adapt models to new devices with limited data and to investigate self-supervised learning to reduce reliance on labelled data. Additionally, deploying the system in a real industrial environment will allow us to validate performance under realistic network conditions and refine the design to handle dynamic topologies and client churn.

References

[1] Sharma, Y., Sharma, S., & Arora, A. (2022, June). Feature ranking using statistical techniques for computer networks intrusion detection. In *2022 7th International Conference on Communication and Electronics Systems (ICCES)* (pp. 761-765). IEEE.

[2] Khraisat, A., Alazab, A., Alazab, M., Obeidat, A., Singh, S., & Jan, T. (2025). Federated learning for intrusion detection in IoT environments: a privacy-preserving strategy. *Discover Internet of Things*, *5*(1), 72.

[3] Sharma, H., Kumar, P., & Sharma, K. (2025, July). Deep Learning based Ensemble Model for Intrusion Detection in IoT Network. In *2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3)* (pp. 1-6). IEEE.

[4] Albanbay, N., Tursynbek, Y., Graffi, K., Uskenbayeva, R., Kalpeyeva, Z., Abilkaiyr, Z., & Ayapov, Y. (2025). Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study. *Journal of Sensor and Actuator Networks*, *14*(4), 78.

[5] Karunamurthy, A., Vijayan, K., Kshirsagar, P. R., & Tan, K. T. (2025). An optimal federated learning-based intrusion detection for IoT environment. *Scientific Reports*, *15*(1), 8696.

[6] Islam, M. M., Abdullah, W. M., & Saha, B. N. (2025). Privacy-Preserving Hierarchical Fog Federated Learning (PP-HFFL) for IoT Intrusion Detection. *Sensors (Basel, Switzerland)*, *25*(23), 7296.

[7] Sharma, H., Kumar, P., & Sharma, K. (2023, February). Identification of device type using transformers in heterogeneous internet of things traffic. In *International Conference On Innovative Computing And Communication* (pp. 471-481). Singapore: Springer Nature Singapore.

[8] Friha, O., Ferrag, M. A., Benbouzid, M., Berghout, T., Kantarci, B., & Choo, K. K. R. (2023). 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. *Computers & Security*, *127*, 103097.

[9] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019, November). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security* (pp. 1-11).

[10] Sharma, H., Kumar, P., & Sharma, K. (2025). Advanced Security for IoT and Smart Devices: Addressing Modern Threats and Solutions. *Emerging Threats and Countermeasures in Cybersecurity*, 191-216.

[11] Gupta, R., Gusain, N., Shirole, B. S., Jagtap, M. T., Thomas, S. A., & KUMAR, S. (2025). Optimizing Healthcare Management Systems with AI and Machine Learning. *South Eastern European Journal of Public Health*, 2973-2985.

[12] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., ... & He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3347-3366.

[13] Gusain, N. (2025). Cardiovascular Disease Prediction through Machine Learning: A Comparative Study of Ensemble Techniques. *Revolutionary Advances in Computing and Electronics: An International Journal*, 27-40.

[14] Chen, J., Yan, H., Liu, Z., Zhang, M., Xiong, H., & Yu, S. (2024). When federated learning meets privacy-preserving computation. *ACM Computing Surveys*, *56*(12), 1-36.

[15] Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., ... & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, *15*, 3454-3469.

[16] Sapra, P., Paikaray, D., Gusain, N., Abrol, M., Ramesh, S., & Bhardwaj, S. (2023). Evaluation of soft computing in methodology for calculating information protection from parameters of its distribution in social networks. *Soft Computing*, 1-11.

[17] Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., ... & Yu, P. S. (2022). Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, *35*(7), 8726-8746.

[18] Gusain, N., & Sharma, H. (2025). Communication-Efficient Federated Learning in Industrial IoT—A Framework for Real-Time Threat Detection and Secure Device Coordination. *International Journal on Computational Modelling Applications*, *2*(2), 18-29.

[19] Sharma, H., Kumar, P., & Sharma, K. (2025). Intelligent Time Series Analysis for Intrusion Detection in the Internet of Things: A Generative-Adversarial-Network-Enhanced Convolutional-Neural-Network–Long-Short-Term-Memory Framework Using Signal Features. *Intelligent Computing*, *4*, 0127.