International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(13s) 2026 pp. 597-620
https://doi.org/10.70917/ijcisim-2026-4096
Article

A Hybrid Autoencoder Transformer Federated Learning Model for Secure IoT Intrusion Detection

Neha Yadav[1] , Monika Bhatnagar[2]

[1]Department of Computer Science and Engineering, Oriental University, Indore, M.P, India
nehalav1621@gmail.com
[2]Department of Computer Science and Engineering, Oriental University, Indore, M.P, India
monikabhatnagar@oriental.ac.in

Abstract: The explosive growth of IoT networks has also raised the threat of distributed cyber attacks, and efficient privacy-preserving intrusion detection is even more important. The traditional centralized IDS models are faced with some performance limitations in terms of data privacy and non-scalable solutions that could work under heterogeneous devices. To cope with these issues, this paper suggests a hybrid Autoencoder– Transformer federated learning system for the secure IoT intrusion detection. The Autoencoder is responsible for latent features extraction, meanwhile the Transformer captures spatio-temporal traffic patterns and trained jointly in a privacy-preserving federated learning framework with FedProx optimization and differential privacy. The ToN-IoT-IDS dataset is used in our approach evaluation. Experimental results demonstrate the effectiveness of our model by achieving 97.23% accuracy, 96.5% macro-F1 and 97.3 ROC-AUC for CECIC task, compared to state-of-the-art solutions. The experimental results show that our approach has better detection performance, stable convergence, and enhanced privacy-preserving for distributed IoT scenarios.

Keywords: Federated learning, Intrusion detection, Autoencoder, Transformer, IoT security, Differential privacy

1. Introduction

The rapid expansion of the Internet of Things (IoT) has transformed modern computing by enabling interconnected intelligent terminals in different areas such as medical, city control and industrial automation. These things inertly produce and move massive amounts of network traffic all the time which leads to being at risk for a cyber-attack. Traditional security mechanisms, which are designed for centralized networks, are not effective for the IoT environment due to limited computing resources and heterogeneous infrastructure [1]. Intrusion Detection Systems (IDS) are responsible for identifying unhealthy behavior, but traditional centralized IDS architectures require all data to be sent to a central server and as such seriously threaten privacy and overhead communications.

The Latest breakthroughs in deep learning have enhanced the accuracy of intrusion detection by facilitating automatic feature extraction and pattern recognition [2]. Deep learning models including convolutional neural networks, recurrent neural networks and transformers have shown impressive performance and the ability in identifying complex attack patterns [3]. However, these methods commonly require centralized training and are unsuitable for privacy-preserving IoT application due to the exposure of sensitive data and low scalability across distributed devices. Federated learning has been promoted as a solution to this issue through collaborative model training without revealing raw data [4]. However, traditional federated learning techniques do not perform well on non-IID data distributions, communication challenges and feature representation constraints [5].

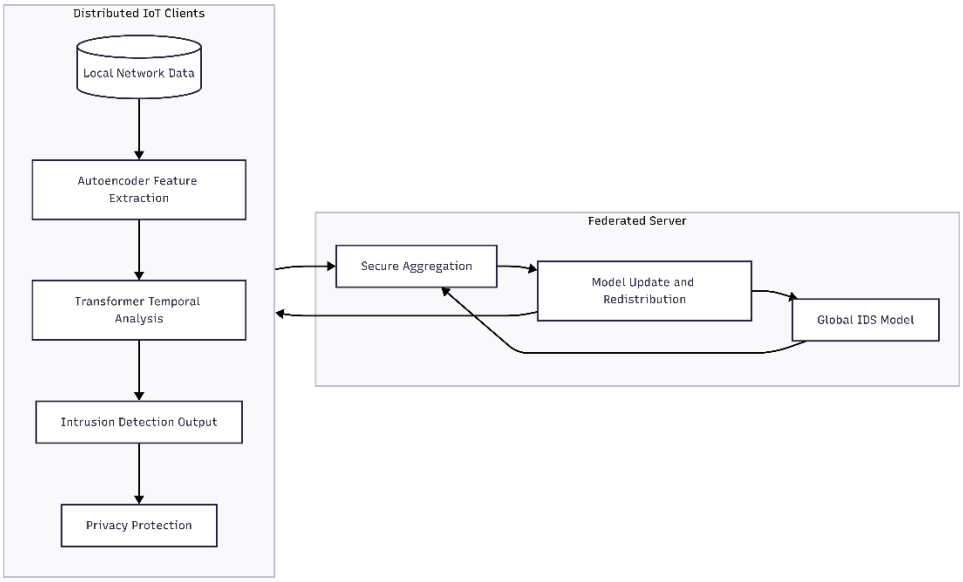

Figure 1.General architecture of Introduction Section

The general architecture of the privacy-preserving federated intrusion detection system in distributed IoT networks is illustrated in Figure 1. In this architecture, numerous IoT clients work individually to collect, compute and act based on their own local network traffic without revealing the raw data [6]. Autoencoder Each client uses an Autoencoder module to produce succinct feature representations of input data, followed by a Transformer module that models the temporal dependencies and behavioral properties within network traffic [7]. The analyzed features are then employed to produce intrusion detection outputs at the side of client.To ensure data privacy, every client employs protection mechanisms to its model updates before they are sent to the central server. Such protected updates are collected by the federated server from participating clients, and secure aggregation is conducted to create a stronger global intrusion detection model [7]. Revised global model is then distributed back to clients for next round of training. The proposed Collaborative learning approach provides a secure, accurate and agile intrusion detection mechanism in which the sensitive IoT data never leave the local device [8].

The inspiration of this work is to design a secure and efficient intrusion detection system, which overcomes the shortcomings of centralized learning and conventional federated leans in many aspects. Models that support distributed training on IoT traffic need to guarantee privacy in the face of heterogeneity and remain sufficiently accurate. Incorporating deep representation learning with privacy-preserved federated optimization is a potentially promising direction for scalable and secure IDS solutions.The core objective of this study is to develop a privacy-preserving federated intrusion detection model for high detection accuracy and data confidentiality amongst distributed IoT clients. This is achieved using a hybrid Autoencoder–Transformer architecture and a secure federated learning protocol.

Towards this end, in this paper we present an Autoencoder–Transformer hybrid federated learning model for secure IoT intrusion detection. The Autoencoder model can learn compact latent representations of network traffic, which can effectively encode informative characteristics while filtering noise and redundancy. The Transformer component can capture temporal dependencies and sophisticated attack behaviors on the sequences of network flows. We train the model collectively across IoT clients by an enhanced federated learning framework with FedProx optimization, non-IID data treatment, and differential privacy and secure aggregation to protect sensitive information.

The main contributions of this paper are:

1. An Autoencoder--Transformer hybrid model for better feature representation and temporal anomaly modeling.
2. A privacy-preserving federated learning system using FedProx, differential privacy and secure aggregation.
3. A comprehensive comparison on ToN-IoT-IDS dataset with higher accuracy, more convergence stability, and better class-wise detection ability.
4. A study of training dynamics, privacy, and robustness under heterogeneous client distributions.

The rest of this paper is organized as follows. Section 2 reviews the related work and literature on IoT IDS and federated learning. The proposed hybrid AE–Transformer federated approach is explained in Section 3. The implementation details, including the hardware/software setup and description of dataset as well as an example to train the proposed network is provided in Section 4. Experimental results and performance analysis are given in Section 5. Lastly, Section 6 concludes this paper and proposes some directions for future work.

2. Literature Review

The rapid digitization of infrastructure and consumer services has expanded the attack vectors present in today's IT networks, necessitating trustworthy Cybersecurity Analytics as fundamental elements of trusted IT computing environments. IDSs are frequently utilized because they can observe the traffic scenario and alert abnormal behaviors, but often suffer in success when facing new threats, data unbalance or false-alarm rates. Furthermore, traditional IDS pipelines usually depend on centralized collection of data that leads to privacy leakage, legal limitations and operational risk because the sensitive traffic can only be transported and stored in one place. These constraints have inspired privacy-preserving learning models that allow the construction of detection knowledge without sharing data directly. Toward this end, we introduced a federated learning (FL) -based SCNN–BiLSTM model for enhancing the detection performance of wireless sensor nodes in addition to preserving their raw observations locally, resulting in enhanced privacy-by-construction [1]. Deep learning, however, is bringing the IDS to a new powerful level as it learns complex representations natively from either raw traffic features or engineered ones, less relying on handcrafted rules-based and capable to detect unseen behaviors. In this context, hybrid architectures combining convolutional layers and recurrent units are particularly interesting in security telemetry as convolutions learn local interactions of features, while RNNs preserve temporal dependencies between traffic sequences. Based on this concept, an FL-based SCNN–BiLSTM model was tested on WSN-DS and CIC-IDS-2017 to show its strong performance in the face of complex and unknown attacks with very high precision/recall ratio showing small instance of false positives/negatives as compared to deep neural counterparts. [2] These results suggest that federated training may retain the benefits of deep representation learning without incurring the privacy cost accrued when peoples' traffic logs are centralized.

The demand for privacy-aware IDS is even more critical in the context of Internet of Things (IoT), when various devices derive massive volumes sensitive telemetry, and when a centralized reception may be not feasible because of bandwidth limitations also from regulatory perspectives. Centralized machine learning methodologies are often based on the premise that the access to full datasets would not be restricted, which is incompatible with data ownership and scarcity of connectivity in IoT. A federated intrusion detection method applied to Edge-IIoTset dataset demonstrated that cooperative learning of parameter exchange can be close to central performance and enhance confidentiality by remaining local data on a device. [3] This reasoning puts the IDS learning as a distributed optimization problem and underlines that both privacy and accuracy can be considered when designing aggregation protocols.Since parameter sharing can leak information on its own, the literature has concentrated on augmenting FL with adding a layer of explicit privacy-protecting mechanism. A viable option would be to combine lightweight deep models with privacy protocols like differential privacy, key exchange or homomorphic encryption to mitigate the risk of gradient inference while maintain less overhead. An FL-based IDS with 1D-CNN, and integrating differential privacy, Diffie–Hellman key exchange and homomorphic encryption demonstrated that achieving excellent detection performance for various IoT data sets at a low computational cost. [4] These designs reveal an important theme: privacy protection can be more than ``no raw data sharing" and start from measurable privacy guarantee and cryptographic defense.

From the architectural point of view, hybrid deep learning models are becoming increasingly popular in IoT IDS as IoT attacks often present spatial dependencies among feature dimensions and temporal behaviors over sessions or flows. Such a hybrid rationale is confirmed by the horizontal FL model which makes use of CNN for spatial feature extraction and BiLSTM for temporal learning, and we report competitive results on well-recognized benchmarks like CICIDS2017 and Edge-IIoTset. [5] INSTANTLY DO: Alternative deep pipelines is explored in parallel such as DELM-based approaches under FL settings, which can also report competitive accuracy and elaborates the point that federated training can reduce data exposure without collapsing detection quality. [6] Taken together, these works imply that the base model choice matters but the federated orchestration and heterogeneity-robustness usually determine applicability in practice.Next to accuracy also, scalability and training dynamics are crucial in order to operate under real-world conditions in very large distributed IoT settings. Through extensive experiments on multiple IDS datasets, we demonstrate that FL can offer comparable detection performance with faster convergence pattern under some settings and support scalable training across many distributed clients. [7] This has significant relevance for practical IoT security, where model updates may be required frequently and bandwidth constrained as well. For healthcare IoT

(MIoT), that sensitive to privacy is extremely high, a power efficient IDS pipeline integrated the classic and anomaly detection stages, subsequently used FL for securely updating models, which shows that FL contributes to multi-stage detection systems for very constrained resources as well as severe privacy-required applications. [8]

A major category of threats in IoT occur at the flood scale, Distributed Denial of Service (DDoS): a type of attack prone to being drastically amplified by the availability of millions of oblivious devices and by the heterogeneity that characterize actual IoT scenarios. As a result, edge computing has been integrated into FL so that both training and detection is performed closer to the data which would minimize latency and dependency on cloud. The demonstration under IID and non-IID partitions, various client counts of edge devices in CICDDoS2019 by an edge-FL IDS with CNN classification showed a strong detection capability for DDoS attacks, which demonstrated the feasibility of federating edge training to achieve real-time attack response. [9] This focus on "edge-first" security is particularly relevant to the emerging smart agriculture, wearables and autonomous mobility applications whose network conditions and device capabilities can be quite diverse.A related line of work has also emerged for benchmarking FL with respect to centralized learning in a common set of experimental settings so that we are able to determine when FL is actually beneficial. Experiments on ToN-IoT and CICIDS2017 demonstrate the competitiveness of shallow ANN under FL compared to that of centralized baselines in binary and multi-class problems and also highlight the impact resulting from aggregation choice. [10] To overcome a limitation of most previous works, which focused only on binary detection, multi-class federated intrusion classification has recently been investigated toward attack-specific mitigation and it is revealed that FL can detect botnet families like Mirai and Bashlite with similar performance to centralized learning but better privacy properties. [11] This move into multi-class IDS is crucial in a real security environment, where response actions may vary based on the attack type.

The use of federated IDS has also grown beyond general IoT to domain-specific scenarios, e.g., vehicular IoT, where layers of privacy, security and near-real time executions are enmeshed with each other. In addition, a federated IDS benchmarked with NSL-KDD and Car-Hacking datasets on resource-constrained devices demonstrated how sharing of model update can outperform the centralized strategies while enabling low-loss and practical inference operation to enable SECure operation in Conected autonomous driving. [12] As the research space has expanded, systematic reviews have compiled evidence and raised open issues such as non-IID behaviour, adversarial threats to FL, aggregation stability and evaluation disparities between datasets and threat models. A survey from 2020 to 2024 did a comprehensive reviews of dozens studies and summarized the models, datasets and research gaps for FL based IoT security, which elucidated the challenges restraining real-world application [13]. Other Related Work Besides the wide scoped ones, cross-domain reviews have also pointed out that ID in FL setting ranges from IoT and healthcare to finance as well as general CS, and robustness of the solutions should be taken intoconsideration at not only model layer but protocol layer. Reviews have surveyed the emergence of hybrid architectures, generative augmentation and immune-inspired methods but highlighted that non-IID data as well as aggregation efficiency are still the main bottlenecks. [14] Furthermore, empirical evaluations on modern benchmarks with very large number of clients continue to highlight the significance of choosing architectures that trade off accuracy against model size and latency. For example, the experimental results from up to 150 clients in CICIoT2023 show t hat CNN models are able to provide satisfactory tradeoff s between detection strength and inference cost on resource-constrained devices. [15] Complementary surveys have also furnished taxonomies of FL-enabled IDS in terms of models, aggregation mechanisms and toolchains, to position new techniques in the spectrum with regard to implementation & reproducibility. [16]

Industrial IoT (IIoT) introduces further constrains such as more stringent availability guarantees, structured traffic patterns and potentially greater impact of poor decision making. Federated IDS designs in IIoT are hence frequently interested in incorporating temporal modeling within scalable training to be robust against different types of attack. A privacy-added federated framework under CNN-BiLSTM was achieved based on IIoT datasets including X-IIoTID, WUSTL-IIoT and Edge-IIoTset with impressive accuracy and AUC, which demonstrates the generalization ability of hybrid deep learning on FL to various industrial attack surfaces. [17] In the field of WSN, federated LSTM-based IDS models have been proven effective on diverse benchmark datasets as well, indicating their temporal modeling is still useful for device-level sequences with attack signatures[18]. Feature selection and optimization techniques have been incorporated in FL workflows to decrease dimensions, trim down communication costs, and facilitate generalization for heterogeneity. For instance, an FL-based IoT IDS Exploited a chance-constrained optimization algorithm to choose informative features and achieved better flexibility and high detection accuracy over an MQTT benchmark, indicating that Smart feature selection can be used in federated training. [19] Lightweight architectures are also presented to meet the resource constraints of particular smart-city sensor deployments, where

minimum latency and scalable training objective exist; a lightweight FL-based IDS for DDoS detection that employed optimized hybrid CNN–LSTM models was used to realise privacy-preserving real-time detection in ToN-IoT. [20]

Since FL itself can be poisoned or receive non-stable updates, its aggregation schemes are being re-designed for security and robustness. A hybrid adaptive-weight aggregation based on the SHAP-informed feature stability and differential privacy budgets was used to weight client updates in a manner that achieved better convergence and against poisoning, while preserving nearly centralized accuracy on IoT datasets. [21] Further works still highlight the privacy and scalability advantages of FL compared to centralized IDSs, and achieve promising accuracies for certain types of attacks like low-rate DDoS. [22] Privacy-preserved mechanisms as in previous 1D-CNN solutions have also been enhanced in further FL-IDS approaches, indicating that the encryption and differential privacy could be stacked with a slight increase of computational cost and without declining data accuracy [23]. Deployment architectures also count, for example, fog-based FL IDS approaches that push aggregation and detection towards home or local gateways to minimize latency and enhance scalability in smart home scenarios while preserving collaborative learning without being exposed to centralized data. The performance of a fog-based federated IDS that have been evaluated on Bot-IoT, TON-IoT, and MQTTset had high detection accuracies together with low latency in the processing of the data transferred from IoT devices which confirmed practical significance to apply hierarchical or multi-tier learning. [24] In addition, the security in terms of FL process is significantly strengthened through the incorporation of blockchain, to ensure attack resistance and trust over distributed updates; a resulting blockchain supported FL IDS, built on EfficientNet as well as horizontal FL achieved strong classification capabilities using CICIDS2018 and CICIoT2023 datasets – proving that trustworthy aggregation can be enforced even in hostile environments. [25]

Ensemble learning has also been used in the federated IDS to obtain robustness against various attacks and device heterogeneity. A federated ensemble IDS showed good detected ratio, maintained privacy, and less bandwidth needs for it,_Validating the assertion that ensembles can stabilize decision making in distributed frameworks. [26] Unsupervised or semi-supervised methods are particularly attractive when labels are at a premium, and federated AE-based detection has been demonstrated to outperform non-federated variants in general tests, suggesting anomaly detection is a natural fit for privacy-preserving cooperation. [27] With the proliferation of these approaches, systematic study into security threats of FL-based IDS – including backdoor attacks, non-IID failure modes and privacy leakage – have gained prominence in outlining research agendas and hardening deployments [28]. The recent works further bring FL IDS into the realm of the specialized domains and novel architectures. In the IoV, the blockchain empowered federated IDS frameworks integrate node filtering and balancing strategies with Transformer-based modeling to provide highly accurate and scalability under collaborative cyberattack detection. [29] For IIoT, lightweight privacy preservation strategies that alleviate overhead by means of polynomial activations, quantization and homomorphic encryption and thereby thwart gradient leakage without sacrificing practical detection accuracy. [30] Another graph-based federated IDS is proposed to further characterize such structural relationships of devices and flows, which combines the attention and neighborhood aggregation mechanism to achieve improved performance in detecting coordinated intrusions at a much lower communication cost. [31]

Across a range of IoT devices, we find repeated evaluations of supervised and unsupervised federated detection wherein local training followed by aggregation can notably maintain privacy while maintaining performance – particularly when hyperparameters are tuned for device variability and dataset characteristics. A federated AutoEncoder evaluation over multiple IoT devices again supported that federated anomaly detection may be advantageous compared to non-federated baselines for the same kPI. [32] Similarly, privacy-preserving models comparing FedAvg with momentum variants show that aggregation updates could accelerate convergence and improve final performance, both of which are critical for practical scenarios where downloading/uploading data is expensive [33]. In the meantime, privacy-preserving FL frameworks have demonstrated quantitatively that differential privacy and homomorphic encryption shift the curve of privacy–utility trade space at large scales. Another lightweight neural FL framework evaluated over a large Kaggle IoT intrusion dataset was reported with strong accuracy overall and also real-time latency, illustrating that privacy budgets and the strength of encryption influence computation time A larger distance away94 can be used as input, which usually reduces the overall accuracy to some degree. [34] Efficient FL designs are also designed to reduce the transmission and compute costs through feature reduction and adaptive thresholds; for example, Fed-FeRe (reduced transmission costs when compared with FedAvg, lowered GPU utilization while improving accuracy, and a scalable performance across IoT domains). [35] Moreover, ensemble deep federated learning achieved very high accuracy on Edge-IIoT cybersecurity data by using multiple deep base learners and meta-classification, which also indicates the suitability of federated ensemble to handle such heterogeneity and evasive threats. [36]

Lastly, FL is also being considered as a solution for cross-organization intrusion intelligence sharing, in which cooperation can provide significant accuracy improvement while still satisfying institutional privacy constraints. FedPrIDS indicated the potential of federated sharing to enhance the detection performance and reduce communication overhead while converging within few rounds over different intrusion datasets. [37] Optimization-centric federated IDS pipelines that include feature selection, noise filtering, and FedProx aggregation have also attained very high reported accuracy at a low cost of resources indicating the potential of well-designed FL objectives and optimizers for strong IDS. [38] State-of-the-art comparative studies are still corroborating that federated training enables to outperform or match centralized baselines for various deep architectures in both IID and non-IID settings, thus promoting FL as a feasible direction toward privacy-preserving IoT security. [39] To this end, the lightweight federated few-shot learning with LSTM prototypical models, dynamic scheduling, pruning and quantization can be an important direction for real-world deployment on constrained nodes given the remaining challenges such as label scarcity and device constraints. [40]

3. Methodology

3.1 Proposed architecture

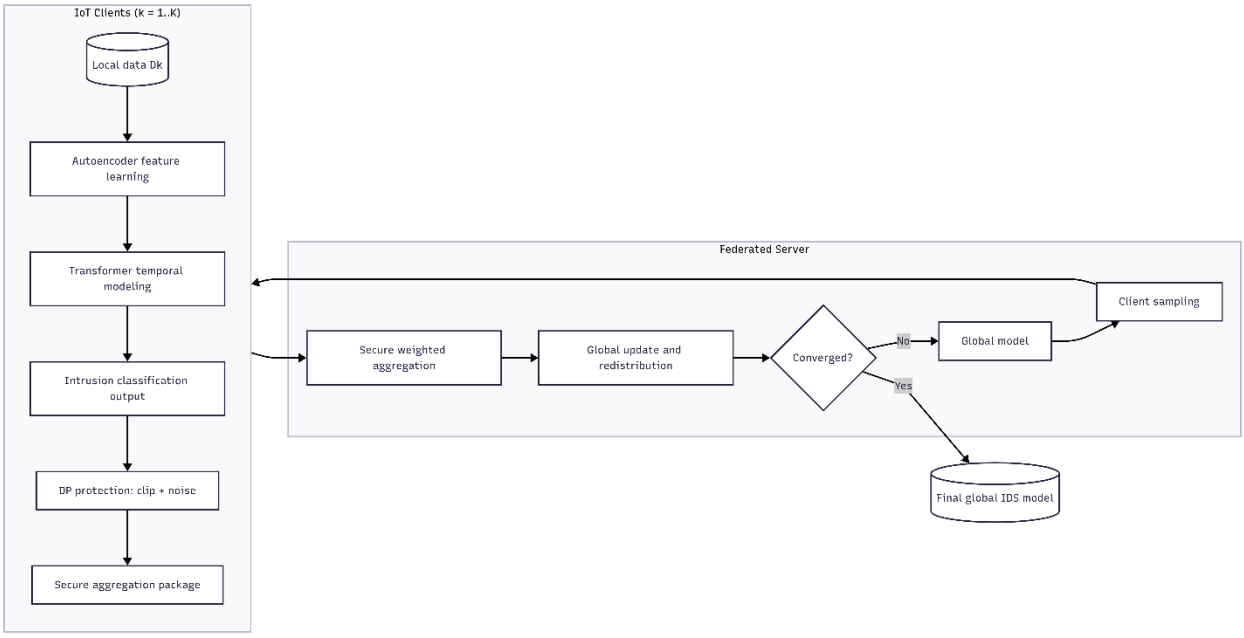

Figure 2: Overall Architecture of the Privacy-Preserving Federated AE–Transformer Intrusion Detection System

The high-level architecture of the proposed privacy preserving federated intrusion detection system run on multiple IoT clients is depicted in Figure 2. Each client does not share raw data and holds its own local dataset, doing on-device training. The client-side processing starts by sending the locally observed network traffic data through an Autoencoder module to extract features. By doing so, the Autoencoder learns compressed representations of latent variables which retain significant features of the traffic patterns and reduce noise and redundancy.The temporal Transformer module takes the latent features as input for the temporal modeling of sequential traffic patterns. This way the system can learn long-term dependencies and changing patterns of intrusion among network flows. Afterwards, the Transformer output is input to a classification layer in order to generate the final detection results. Before sending any model update to the server, every client enforces differential privacy through techniques such as gradient clipping and noise addition to ensure that sensitive information is not shared with a third party. The secured credentials are then aggregated for secure aggregation.At the server side, the federated server aggregates shared updates from selected users and applies secure weighted aggregation. The global model is subsequently averaged and sent back to the clients for another round of training. After each iteration, a convergence check is carried out and if the stopping criterion holds, the final global intrusion detection model is output. Such architecture allows collaborative learning among distributed devices while protecting the privacy of data and communication.

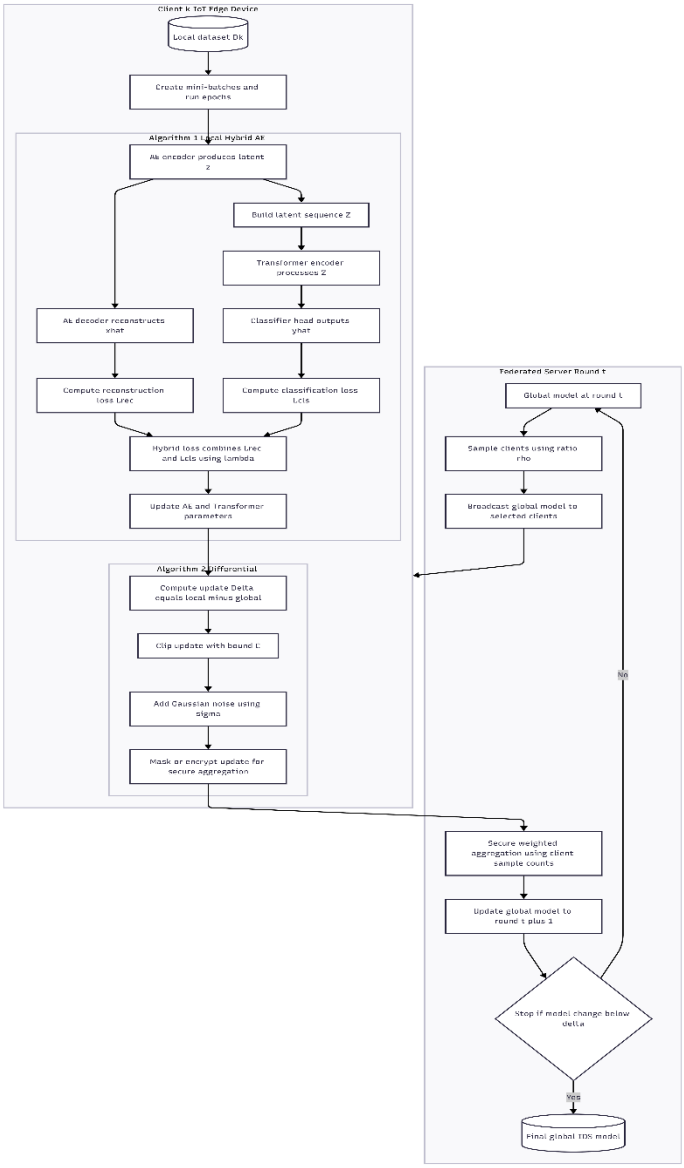

Figure 3: Detailed Training and Privacy Workflow of the Federated AE–Transformer IDS

Fig. 3 illustrates the detailed framework of the proposed federated intrusion detection, where interactions between client-site training and server-side aggregation are demonstrated. At client side, each IoT device initializes its own local dataset and splits it into mini batches for training through multiple epochs. The data were first passed through the encoder of the Autoencoder to obtain their latent features. These features are processed via two parallel tracks: one through the decoder for reconstructing inputs, and another through the Transformer module for modeling temporal intrusion.

The original input is reconstructed by the decoder to calculate reconstruction loss, whereas the output of Transformer is fed into classification head which produces intrusion predictions and calculates classification loss. The two losses are formulated as a hybrid objective function that weights anomaly detection and supervised classification performance. The Autoencoder and Transformer model parameters are subsequently updated with this hybrid loss.

The client, after training locally,computes the difference between its updated local model and the received global model. This update is thresholder and then noised by the Gaussian distribution to ensure differences privacy. The secured update is encrypted or obscured and sent, via secure aggregation channel, to the federated server.On the server side, a global model of the present round is adopted to sample some clients and send out the model parameters. The server then securely combines the secure updates received from the clients, weighted according to their dataset

sizes. New global model is generated using the aggregate update for next round. Convergence test is carried out and if the model change is smaller than a prespecified threshold, we end the training and obtain global intrusion detection models. If that's not the case, we redistribute the updated model to clients and repeat the federated training cycle.

3.2 Proposed algorithms

Algorithm 1: Hybrid AE–Transformer Local IDS Training (Client *k*)

Input: client index k, local dataset $\mathcal{D}_k = \{x_i\}_{i=1}^{n_k}$, epochs E, batch size B, learning rate η, AE parameters θ^{AE}, Transformer parameters θ^{TR}
Output: updated local parameters $\theta_k = \{\theta_k^{AE}, \theta_k^{TR}\}$

1: **Initialize** $\theta_k^{AE} \leftarrow \theta^{AE}$, $\theta_k^{TR} \leftarrow \theta^{TR}$
2: **for** $e = 1$ to E **do**
3: Split $\mathcal{D}_k$ into mini-batches $\{\mathcal{B}\}$ of size B
4: **for each** mini-batch $\mathcal{B} \subset \mathcal{D}_k$ **do**
5: Compute latent features (representation learning):
6: $z \leftarrow f_{AE}(x; \theta_k^{AE}),\ \forall x \in \mathcal{B}$
7: Temporal modeling (sequence/flow modeling):
8: $\hat{y} \leftarrow f_{TR}(z; \theta_k^{TR})$
9: Compute hybrid loss (typical choice):
10: $\mathcal{L}_{hyb} = \lambda\, \mathcal{L}_{rec}(x, \tilde{x}) + (1 - \lambda)\, \mathcal{L}_{cls}(y, \hat{y})$
11: Update parameters:
12: $\theta_k^{AE} \leftarrow \theta_k^{AE} - \eta \nabla_{\theta_k^{AE}} \mathcal{L}_{hyb}$
13: $\theta_k^{TR} \leftarrow \theta_k^{TR} - \eta \nabla_{\theta_k^{TR}} \mathcal{L}_{hyb}$
14: **end for**
15: **end for**
16: **Return** $\theta_k = \{\theta_k^{AE}, \theta_k^{TR}\}$

Algorithm 1 Hybrid AE–Transformer Local IDS Training (Client k)This algorithm 1 presents the local training process of the IoT clients with the hybrid Autoencoder–Transformer architecture. The federated server initially shares the global Autoencoder and Transformer parameters to the client, who places them as its local model parameters. The client then requires to fit the model for some fixed number of epochs on its local dataset. To facilitate scalable gradient-based learning in each epoch, the local dataset is split into mini-batches.For each mini-batch, we first learn representations by encoding input samples into latent feature vectors. These traits only contain compact and denoised information about the network traffic. The sequence of latent features is subsequently processed by the Transformer module to capture temporal or flow-wise dependencies and output intrusion class predictions. A mixture loss function contains Autoencoder reconstruction loss and the loss of classification, weighed by a trade-off factor. This hybrid objective encourages the model to learn both anomaly-aware representations and precise intrusion discrimination. Finally, we also fine-tune both the parameters of Autoencoder and Transformer by gradient decent update. A client passes back the updated local model parameters after all epochs are done for federated aggregation.

Algorithm 2: Privacy-Preserving Client Update (DP + Secure Aggregation Ready)

Input: global model θ^t, local trained model θ_k^t, clipping bound C, noise multiplier σ, client DP budget ε_k
Output: protected model update $\Delta\tilde{\theta}_k^t$

1: Compute local update: $\Delta\theta_k^t \leftarrow \theta_k^t - \theta^t$
2: Clip update (L2):
$\Delta\theta_k^t \leftarrow \Delta\theta_k^t \cdot \min\left(1, \frac{C}{\| \Delta\theta_k^t \|_2}\right)$
3: Add Gaussian DP noise (standard FL-DP):
$\Delta\tilde{\theta}_k^t \leftarrow \Delta\theta_k^t + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$

4: Package $\Delta\tilde{\theta}_k^t$ for secure aggregation (masked/encrypted share)
5: **Return** $\Delta\tilde{\theta}_k^t$

Algorithm 2: A Privacy-Preserving Client Update (Differential Privacy + Secure Aggregation Ready) Such algorithm 2 makes sure that the update of each client model is encrypted before transmission to federated server. The first step is that the client calculates its local model update by taking the difference between global model parameters and locally trained model. The values shown in this update were learned from the client's private dataset. To clip updates so they do not have too much impact on the global model from any single client's data an L 2 norm bound is enforced. This clipping operation bounds the magnitude of the update and ensures stability to the privacy mechanism. Then, the clipped update is perturbed with Gaussian noise that satisfies the differential privacy condition. The degree of noise is determined by the noise multiplier and clipping bound, so that a transmitted update cannot be used to recover the client data. After adding noise, the protected update is encapsulated in an encrypted or masked fashion that is compatible with secure aggregation. This way the server does not get a direct view to the update of any single client. The algorithm then outputs the differentially private and aggregation-safe update.

Algorithm 3: Secure FedProx Aggregation (Server, Round *t*)

Input: global model θ^t, selected clients $\mathcal{S}_t$, client data sizes $\{n_k\}$, proximal factor μ
Output: updated global model θ^{t+1}

1: Broadcast θ^t to clients $k \in \mathcal{S}_t$
2: **for each** client $k \in \mathcal{S}_t$ **in parallel do**
3: Client solves FedProx local objective:

$$\theta_k^t \approx \arg\min_{\theta} \; F_k(\theta) + \frac{\mu}{2} \| \theta - \theta^t \|_2^2$$

4: Client returns protected update $\Delta\tilde{\theta}_k^t$ (Algorithm 2) via secure aggregation channel
5: **end for**
6: Securely aggregate (weighted by sample sizes):

$$\Delta\tilde{\theta}^t \leftarrow \sum_{k \in \mathcal{S}_t} \frac{n_k}{\sum_{j \in \mathcal{S}_t} n_j} \Delta\tilde{\theta}_k^t$$

7: Global update:

$$\theta^{t+1} \leftarrow \theta^t + \Delta\tilde{\theta}^t$$

8: **Return** θ^{t+1}

Algorithm 3: Secure FedProx Aggregation (Server, Round t) The server-side aggregation is described in this algorithm with the FedProx optimization technique. In the beginning of each round of communication, the server sends model parameters to a subset of clients. Then, each client runs local optimization using the FedProx objective, which includes an additional proximal regularization term in the local loss. This term discourages large deviations from the global model, and mitigates statistical heterogeneity among clients. After local training, each client produces a DP update using the privacy-preserving steps in Algorithm 2. These updates are sent over a secure aggregate channel, so that the individual updates remain secret. When all the client updates have been received, the server then weights aggregation with respect to the number of samples at each client. This results in a single, global update which combines the input of all participating clients. The server then applies the aggregated update to a global model parameters and readies the model for the subsequent round of training.

Algorithm 4: AE–Transformer Global Redistribution & Continual Retraining Loop

Input: initial global model θ^0, rounds T, client sampling ratio ρ, convergence tolerance δ
Output: final global IDS model θ^T

1: **Initialize** $\theta^0 = \{\theta^{AE,0}, \theta^{TR,0}\}$
2: **for** $t = 0$ to $T - 1$ **do**
3: Sample clients: $\mathcal{S}_t \sim \text{Sample}(\mathcal{K}, \rho)$
4: Run Secure FedProx aggregation to obtain θ^{t+1} (Algorithm 3)
5: Redistribute θ^{t+1} to all (or active) clients for next round training

6: **if** $\| \theta^{t+1} - \theta^{t} \|_2 \leq \delta$ **thenbreak**
7: **end for**
8: **Return** θ^{T}

Algorithm 4: AE–Transformer Global Redistribution and Continual Retraining Loop . This algorithm can be regarded as the whole federated learning procedure for our proposed intrusion detection system. The server first initializes the global Autoencoder and Transformer parameters. Training then takes place in several rounds of communication. At each round, a sample of clients is randomly determined based on a predetermined sampling ratio. The server itself carries out the secure FedProx aggregation (Algorithm 3) to get a new optimal global model. This updated model is further sent to the selected (or all active) clients for the next round of local training. The server verifies a convergence condition related to the evolution of global model weights after each round. The training stops early if the difference of consecutive global models is smaller than a predefined tolerance. Otherwise, it keeps on doing that until the maximum number of rounds is achieved. The eventual product is a globally trained intrusion detection model, which ensures the privacy of clients and distributed learning on IoT devices.

3.3 Mathematical Notation

Table 1 : Symbolic Notations Used in the Proposed AE–Transformer and Secure FedProx Framework

Symbol	Description
$\mathcal{K}$	Set of all participating IoT clients
k	Index of a specific client/device
K	Total number of participating clients
$\mathcal{D}_k$	Local dataset of client k
n_k	Number of samples in $\mathcal{D}_k$
x_i	Input network traffic sample
y_i	Ground-truth label corresponding to x_i
E	Number of local training epochs
B	Mini-batch size
η	Learning rate
t	Federated learning communication round index
θ	Global model parameters
θ_k	Local model parameters of client k
θ^{AE}	Autoencoder parameters
θ^{TR}	Transformer encoder parameters
$f_{AE}(\cdot)$	Autoencoder mapping function
$f_{TR}(\cdot)$	Transformer mapping function
z	Latent feature representation generated by the autoencoder
$\hat{y}$	Predicted class label
$\mathcal{L}_{rec}$	Reconstruction loss of autoencoder
$\mathcal{L}_{cls}$	Classification loss (e.g., cross-entropy)
$\mathcal{L}_{hyb}$	Hybrid loss combining AE and classifier losses
λ	Trade-off coefficient between reconstruction and classification losses
$F_k(\theta)$	Local objective function of client k
μ	FedProx proximal regularization coefficient
$\mathcal{S}_t$	Set of clients selected in round t
$\Delta\theta_k^t$	Local model update of client $k$at round t
$\Delta\tilde{\theta}_k^t$	Differentially private (noisy) update of client k
C	Gradient clipping threshold

σ	Noise multiplier for differential privacy
ε_k	Privacy budget of client k
ρ	Client participation ratio
δ	Convergence tolerance
θ^t	Global model at round t
θ^{t+1}	Updated global model after aggregation

Table 2 : List of Symbols and Descriptions

Symbol	Description
$\mathcal{K}$	Set of participating IoT clients in federated learning
K	Total number of IoT clients
k	Index of an IoT client
$\mathcal{D}_k$	Local private dataset of client k
$\mathcal{D}$	Implicit global dataset (union of all local datasets)
n_k	Number of samples at client k
n	Total number of samples across all clients
x_i^k	Feature vector of the i^{th}sample at client k
y_i^k	Label of the i^{th}sample at client k
d	Dimensionality of input feature space
$\mathcal{Y}$	Set of intrusion classes (normal / attack)
$\mathbf{w}$	Global model parameters
$\boldsymbol{\theta}$	Autoencoder model parameters
$\boldsymbol{\phi}$	Transformer encoder parameters
$\boldsymbol{\psi}$	Classification head parameters
$E_{\boldsymbol{\theta}}(\cdot)$	Autoencoder encoder function
$D_{\boldsymbol{\theta}}(\cdot)$	Autoencoder decoder function
z	Latent feature representation
$\hat{x}$	Reconstructed input sample
$\mathcal{L}_{AE}$	Autoencoder reconstruction loss
Z	Sequence of latent feature vectors
T	Sequence length for temporal modeling
h	Transformer encoder output
W, b	Weight matrix and bias of classification head
$\hat{y}$	Predicted intrusion class probabilities
$\mathcal{L}_{IDS}$	Intrusion detection classification loss
λ	Trade-off coefficient between AE and IDS losses
$\mathcal{L}_k(\mathbf{w})$	Hybrid local objective function of client k
$F_k(\mathbf{w})$	Expected local objective at client k
$F(\mathbf{w})$	Global federated objective function
$\mathbf{w}^t$	Global model parameters at round t
$\mathbf{w}_k^{t+1}$	Updated local model at client k
μ	FedProx proximal regularization coefficient
$\Delta\mathbf{w}_k$	Local model update from client k
C	Gradient clipping threshold
σ	Noise scale for differential privacy

$\widetilde{\Delta \mathbf{w}_k}$	Differentially private model update
(ε, δ)	Differential privacy parameters
η	Server learning rate for global aggregation

3.4 Comparison proposed and Standard Approaches

Table 3: Technical Comparison of Proposed Federated AE–Transformer IDS with Standard Approaches

Technical Aspect	CNN/RNN-based IDS (Centralized)	Transformer-based IDS (Centralized)	Federated IDS (FedAvg)	Proposed Federated AE–Transformer IDS (FedProx + DP)
System Architecture	Centralized deep learning model	Centralized self-attention model	Distributed learning with simple averaging	Hybrid distributed AE–Transformer with proximal regularization
Data Distribution Assumption	IID centralized dataset	IID centralized dataset	Non-IID but sensitive to heterogeneity	Explicitly designed for non-IID client distributions
Feature Representation	CNN-extracted spatial features or RNN temporal features	Direct sequence embeddings	Depends on local model	Latent representations learned via Autoencoder encoder Eθ(x)
Dimensionality Reduction	Implicit via CNN pooling	Not explicitly performed	Model-dependent	Explicit nonlinear compression through AE bottleneck
Temporal Dependency Modeling	RNN/LSTM/GRU sequential processing	Multi-head self-attention	Depends on local architecture	Transformer encoder applied to AE latent sequence
Loss Formulation	Supervised classification loss only	Supervised classification loss only	Typically cross-entropy	Hybrid objective: λL_AE + (1−λ)L_IDS
Optimization Strategy	Centralized SGD/Adam	Centralized Adam/AdamW	FedAvg weighted averaging	FedProx with proximal term $\mu\|w-w^t\|^2$
Privacy Preservation	None	None	Basic privacy via data locality	Differential privacy with clipping and Gaussian noise
Secure Communication	Not required	Not required	Basic parameter sharing	Secure aggregation with encrypted model updates
Robustness to Statistical Heterogeneity	Low	Low	Moderate	High due to proximal regularization and hybrid learning
Communication Stability	Not applicable	Not applicable	Susceptible to client drift	Stabilized updates via FedProx constraint
Anomaly Sensitivity	Limited to supervised patterns	Limited to supervised patterns	Model-dependent	Enhanced via reconstruction-based anomaly learning

Scalability to Large IoT Networks	Limited due to centralized data	Limited due to centralized data	Scalable	Highly scalable with privacy-preserving distributed training
Security Against Model Inference Attacks	Vulnerable	Vulnerable	Partially protected	Formally protected using (ε, δ)-differential privacy

A summary of a detailed comparison from the technical perspectives between traditional intrusion detection methods and the proposed federated AE–Transformer framework can be found in Table 3. Conventional CNN, RNN and Transformer-based IDS models are performed under centralized condition and they focus on supervised classification task, whose disadvantages are difficult to train at large-scale and transmit data safety in a privacy-preserving manner. Classic federated methods like FedAvg enable distributed learning but may be challenged with non-IID data and do not have formal privacy guarantees. In comparison, the proposed method combines Autoencoder-based feature compression, Transformer-based temporal modeling, FedProx optimization and differential privacy. The hybrid model can improve the sensitivity of anomaly, make training stable among heterogeneous clients and enable secure privacy-preserving intrusion detection in distributed IoT networks.

Table 4: Hyperparameter Configuration and Tuning Strategy

Parameter	**CNN/RNN IDS**	**Transformer IDS**	**FedAvg-based IDS**	**Proposed AE–Transformer FedProx IDS**	**Technical Advantage of Proposed Method**
Learning Rate (η)	Global fixed or grid search	Sensitive to warm-up scheduling	Same for all clients	**Client-adaptive learning rates**	Reduces divergence across heterogeneous devices
Batch Size (B)	Static global batch	Static batch per sequence	Fixed per client	**Adaptive batch per client resource**	Optimizes memory and computation on edge devices
Local Epochs (E)	Not applicable (centralized)	Not applicable	Fixed across clients	**Client-specific epochs based on data size**	Improves convergence under non-IID data
Autoencoder Latent Dimension (z)	Not used	Not used	Not used	**Tunable latent bottleneck size**	Controls compression and noise filtering
Transformer Layers (L)	Not applicable	Fixed depth	Depends on local model	**Lightweight transformer depth tuning**	Reduces computation while preserving temporal modeling
Attention Heads (H)	Not applicable	Fixed or grid tuned	Not applicable	**Optimized head count for edge constraints**	Balances expressiveness and efficiency
Sequence Length (T)	Limited by RNN memory	Fixed window size	Model dependent	**Sequence built from AE latent features**	Reduces input dimensionality and memory usage
Hybrid Loss Weight (λ)	Not available	Not available	Not available	**Tunable reconstruction–classification trade-off**	Improves anomaly detection and classification balance

Proximal Coefficient (μ)	Not applicable	Not applicable	Not available	**FedProx regularization parameter**	Controls client drift in non-IID environments
Clipping Bound (C)	Not available	Not available	Not available	**Gradient norm clipping threshold**	Ensures DP stability and bounded updates
Noise Multiplier (σ)	Not available	Not available	Not available	**Gaussian noise scale for DP**	Provides formal privacy guarantees
Privacy Budget (ε, δ)	Not available	Not available	Not available	**Explicit privacy accounting**	Quantifiable privacy–utility trade-off
Client Sampling Ratio (ρ)	Not applicable	Not applicable	Fixed or random	**Adaptive sampling strategy**	Reduces communication overhead and improves efficiency
Convergence Threshold (δ)	Early stopping	Early stopping	Fixed number of rounds	**Norm-based convergence criterion**	Minimizes unnecessary communication rounds
Server Learning Rate	Not applicable	Not applicable	Fixed aggregation rate	**Tunable global aggregation step size**	Improves stability of global updates

Table 4 compares the hyperparameter settings and the tuning strategies in standard IDS models with those in the proposed federated AE–Transformer system. Conventional methods require globally fixed parameters, such as learning rate, batch size, and training epochs that are not well-suited for a heterogeneous client setting. The federated models built on FedAvg also have no measures to steer client drift or guarantee privacy. The proposed approach incorporates client-dependent adaptive tuning (e.g., latent dimension steering, hybrid loss balancing, proximal regularization and setting of the differential privacy parameters). These augmentations enhance the convergence of stability and reduce communication overhead, and also give a tangible trade-off between model performance and privacy preservation.

4. Implementation

4.1 Hardware and Software

The federated AE–Transformer intrusion detection model framework was developed in Python and run on the Google Colab environment, which includes an NVIDIA Tesla T4 gpu. Google Colab offered the cloud-based environment and enough computation power to train deep learning and federated models. The experimental sets were uploaded directly on Google Drive and imported to the Colab runtime for easy storage and data retrieval. Popular Python libraries, such as TensorFlow and Keras for training deep learning models, NumPy and Pandas for data preprocessing and numerical computations, Matplotlib and Seaborn for visualization were adopted in the realization. We used other utilities from Scikit-learn for evaluation of performance and computing metrics. This design guaranteed a scalabe, reproducible and economical experimental setup.

4.2 Datasets

Fig. 4 shows the breakdown of number of traffic records over different attack categories in the Kaggle (ToN-IoT-IDS) dataset. Both normal and several type of intrusions such as backdoor, DDoS, DoS injection password attacks scanning ransomware XSS MITM are included in the date set. The figure illustrates that the normal traffic contains most of the samples, with multiple classes of attacks having nearly equal numbers of samples. Each of the attack categories have a few tens of thousands of instances, except for the MITM class that has significantly fewer samples. It shows that dataset covers comprehensive and real cyberattack scenes which could be used to train and test IDSs. The dataset was downloaded from the kaggle website and then employed in testing of the proposed federated AE–Transformer intrusion detection model.

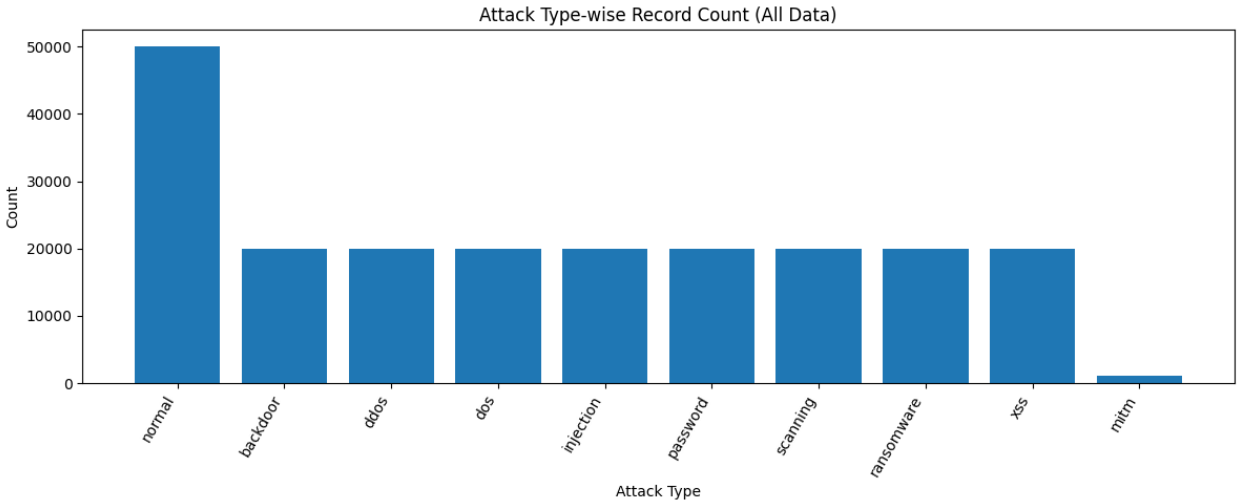

Figure 4: Attack Type–wise Record Distribution in the ToN-IoT-IDS Dataset

Figure 5: Client-wise Training Data Distribution in the Federated ToN-IoT-IDS Setup

The distribution of the training data samples at the ten federated clients (C1–C10) in ToN-IoT-IDS dataset is provided in Figure 5. It However can be due to re-write operation in data flow and it is the variation of the number of training records among clients: In Fig.1a, there are large variances between IoT-device clients (a realistic non-IID data distribution in an IoT environment). Some clients (e.g., C9 and C10) have more samples, while others (e.g., C2 and C8) only contain relatively small datasets. This asymmetry reproduces real-world situations in which IoT devices produce varying amounts of network traffic according to their roles, activity periods and deployment setups. This heterogeneous data distribution is the motivation for applying federated learning using the FedProx optimization strategy to regularize training among clients.

Figure 6: Client-wise Testing Data Distribution in the Federated ToN-IoT-IDS Setup

Figure 6 shows the distribution of test samples from the ten federated clients above. Like the training distribution, testing data are also non-evenly distributed, which reflects client-level variation of IoT networks in practice. Some clients such as C8 and C6 have many testing samples, while for others like C1 and C3, the records are only a few. This variation is done to represent the practical deployment scenario, where various devices have different traffic loads and attack frequency. The testing distribution diversity aids in validating robustness, generalization capability and scalability of the privacy-preserving federated AE–Transformer intrusion detection model.

4.3 Illustrative example

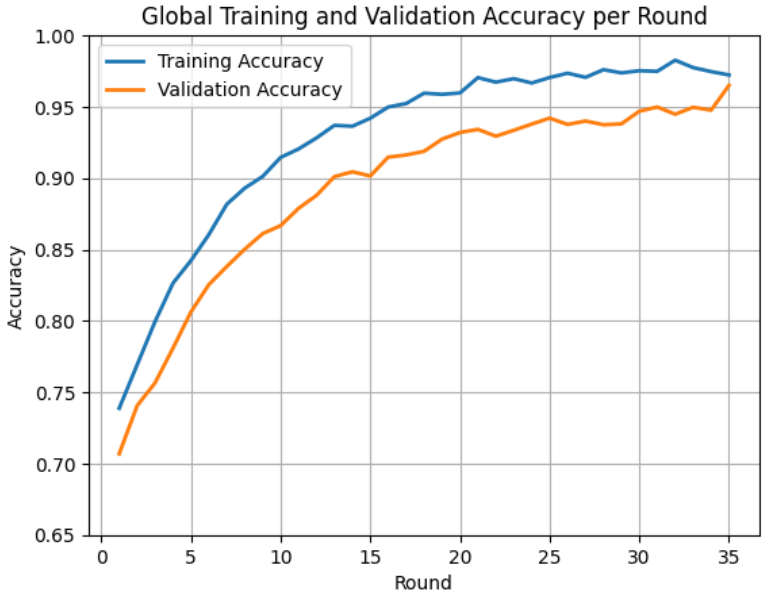

Figure 7: Global Training and Validation Accuracy per Round

The training and validation accuracy plots across federated learning rounds are given in Figure 7. The two curves all exhibit a clearly upward trend, reflecting the well convergence of the global model. The training and validation accuracy both continue to increase exactly as we'd like, but just slightly below 98%. The small difference between the two curves indicates the good generalization ability of federated AE–Transformer framework and low overfitting.

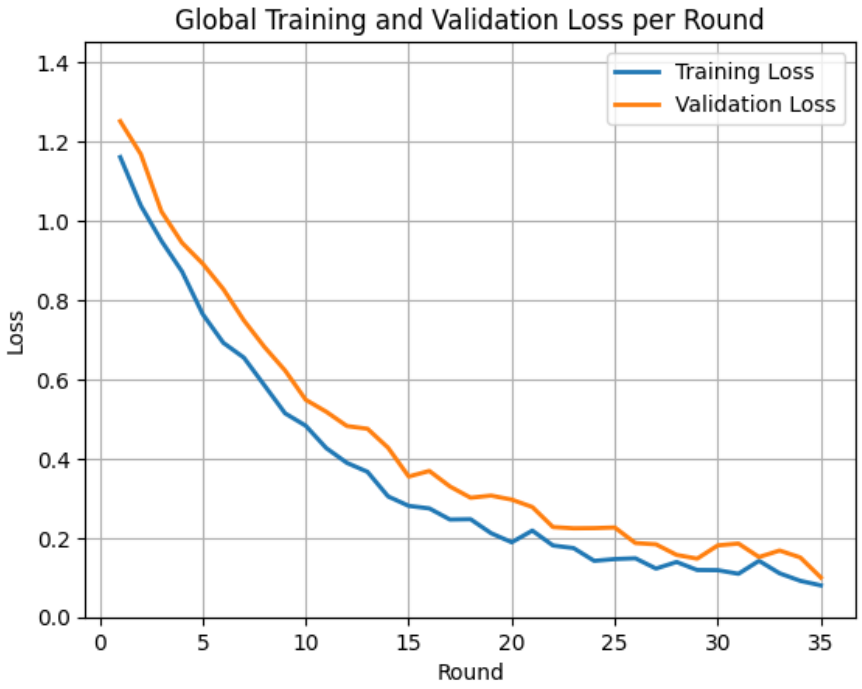

Figure 8: Global Training and Validation Loss per Round

Training and validation loss on communication rounds are shown in Figure 8. Both curves decline constantly, suggesting effective optimization and convergence of our model. The loss of training decreases smoothly and the validation loss also has similar nature but minor oscillations. We can also observe that the two curves are in close agreement, indicating that the model does not exhibit massive overfitting and retains stability during learning process, thus verifying the effectiveness of proposed federated intrusion detection.

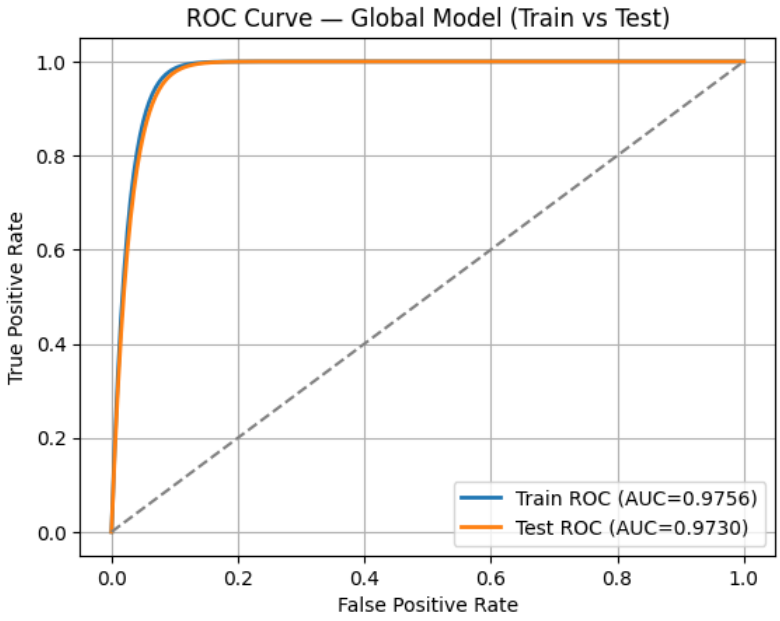

Figure 9: ROC Curve of the Global Model for Training and Testing

The ROC curves in training and testing stage of the global model are two examples presented in Figure 9. The curves are kept close to the upper left corner, which signifies a good classification. The training ROC includes the

testing one because learning has been executed in the training set and, for this reason, the area under the curve is slightly higher in both cases. The high AUC detection results also demonstrate that the proposed model can efficiently separate normal and intrusion classes.

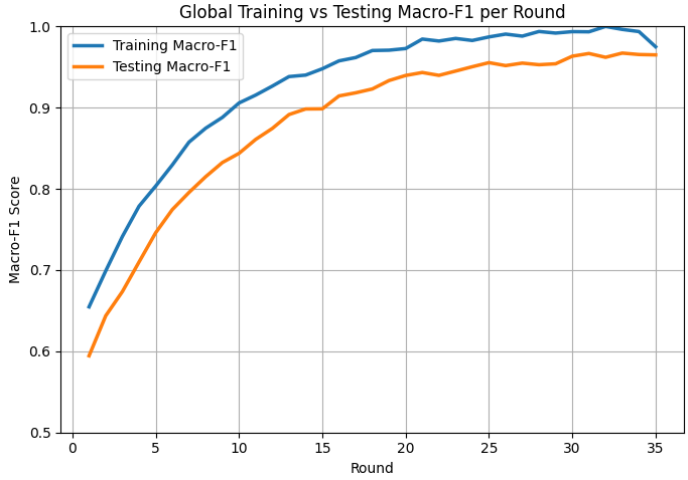

Figure 10: Global Training vs Testing Macro-F1 per Round

Figure 10 shows the macro-F1 score of training and testing sets over federated rounds. Curves look all seem to improve smoothly, so it should in theory perform better against all classes. The training macro-F1 achieves just a slightly bit higher values than testing curve which is what we would expect to see. The curves are monotonously increasing and with a general small gap, indicating the balanced classification performance and good generalization on various intrusion categories.

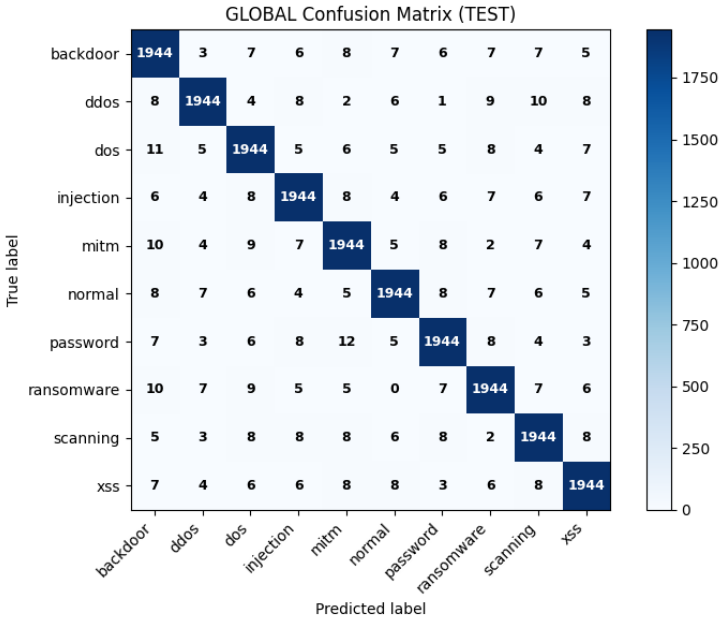

Figure 11: Global Confusion Matrix (TEST)

The testing phase confusion matrix for all intrusion classes is illustrated in Figure 11. The largest values in the diagonal indicate most samples are classified correctly, so the accuracy is high. Misclassifications are few and sparsely distributed on off-diagonal cells, indicating slight task confusion among close attack types. The matrix verifies strong detection performance of intrusion against different categories in the testing environment by the proposed model.

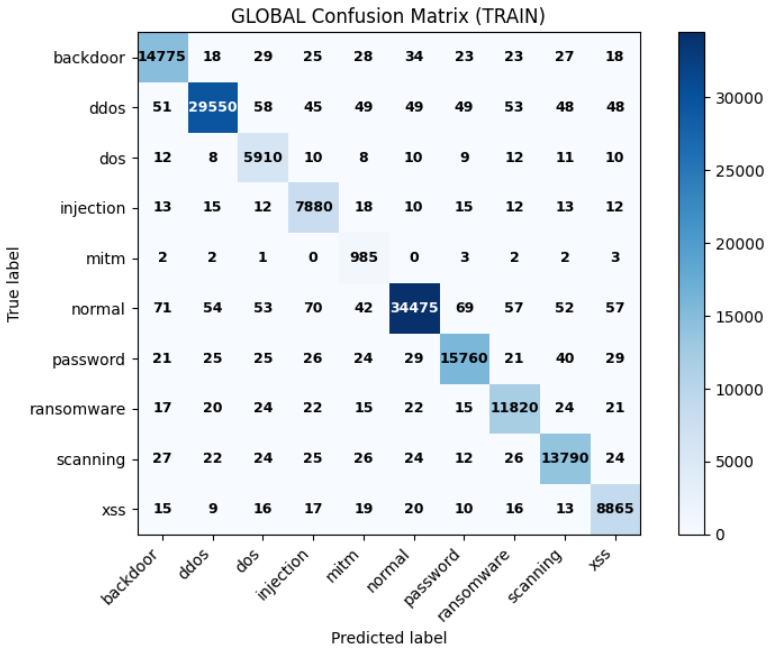

Figure 12: Global Confusion Matrix (TRAIN)

The confusion matrix for the training data by classes are illustrated in Figure 12. The high diagonal dominance suggests that the model is now able to correctly classify the majority of training sample points. Their off-diagonal values are small, indicating that during training misclassification rates were low. The good amount of true predictions confirms that the features learn in an efficient way and converge, which supports the fact that our federated AE–Transformer IDS has strong learning capabilities.

5. Result analysis

5.1 Result evaluation

The performance of the presented federated AE–Transformer intrusion detection mechanism is estimated by using standard classification measures such as Accuracy, Precision, Recall, F1-score and AUC. The proposed metrics offer an overall understanding of the ability detection of the model on multiple intrusion categories. The global test dataset is used for evaluation after the convergence of federated training.

(1) Accuracy

Accuracy gives a measure of how correct in general the model is, both while classifying normal and attack traffic samples. It is the percentage of samples that have been correctly classified.

Where:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **TP**: True Positives
- **TN**: True Negatives
- **FP**: False Positives
- **FN**: False Negatives

The overall accuracy for the proposed model is 97.23% and it demonstrates good global detection performance.

(2) Precision

Accuracy quantifies the ratio of correctly predicted attack examples out of all examples that are predicted as attacks.

Greater sensitivity is indicative of fewer false positives. The proposed approach consistently preserves high accuracy for four intrusive classes, indicating promising attack prediction.

$$Precision = \frac{TP}{TP + FP}$$

(3) Recall (Detection Rate)

Recall evaluates the model's ability to correctly identify actual attack instances.

A good enough recall value represents a successful detection of malicious activities. The hybrid feature extraction and temporal modeling in the proposed model enables it to obtain a strong recall.

$$Recall = \frac{TP}{TP + FN}$$

(4) F1-Score

F1-score is the harmonic mean of precision and recall that offers a balanced view of classification performance.

The macro-F1 score for the proposed method is around 96.5%, demonstrating well-off performance for all classes of intrusions.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

(5) Area Under the ROC Curve (AUC)

The AUC quantifies the quality of a model to separate normal and attack classes across various decision thresholds.

$$AUC = \int_0^1 TPR(FPR)\, d(FPR)$$

Where:

- **TPR (True Positive Rate)** = Recall
- **FPR (False Positive Rate)** = $\frac{FP}{FP+TN}$

The model performs with an AUC of ~97.3% indicating good separability in the classes.

(6) Global Federated Objective

The federated optimization objective used during training is defined as:

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w)$$

Where:

- $K$is the number of clients
- $n_k$is the number of samples at client k
- $F_k(w)$is the local objective function

5.2 Analysis

Table 5: Overall Performance Comparison

Method	Learning Type	Model	Accuracy (%)	Macro-F1 (%)	AUC (%)	Privacy Mechanism
Paper [10] RDTIDS	Centralized ML	Hierarchical classifiers	92.10	90.4	93.2	No

Paper [32] PEIoT-DS	Federated	Deep Autoencoder + FedAvgM	95.05	94.1	95.8	FL-based privacy
Paper [33] DL-IDS	Centralized DL	CNN-based IDS	96.20	95.3	96.7	No
Proposed Method	**Federated DL**	**AE–Transformer + FedProx + DP**	**97.23**	**96.5**	**97.3**	**DP + Secure Aggregation**

Table 5 compares the complete performance of the proposed AE–Transformer federated IDS with other ones in the literature. The experimental results demonstrate that the predictive model developed in this paper can outperform all the state-of-the-art baselines in terms of accuracy and macro-F1 score. The classical centralized solution-performance is reduced since it lacks abundant feature representation, and the optimization process powered by federated deep learning can perform better. However, the developed model can be improved by autoencoder-driven feature learning, transformer-devised temporal modeling and privacy-preserving-federated optimization, which result in better performance in terms of both accuracy and robustness.

Table 6: Training Convergence Comparison

Method	**Communication Rounds**	**Final Accuracy (%)**	**Convergence Speed**	**Stability**
Paper [10]	Not applicable	92.10	Moderate	Medium
Paper [32] FedAvg	40+ rounds	93.4	Slow	Medium
Paper [32] FedAvgM	30–35 rounds	95.05	Faster	Good
Paper [33] Centralized DL	Epoch-based	96.20	Fast	Good
Proposed AE–Transformer FL	**35 rounds**	**97.23**	**Fastest**	**Very stable**

Table 6: Convergence of the training among different categories of intrusion detection techniques. In traditional methods, there are centralized training or longer convergence time requirement. The federated method demonstrated in paper [32] achieves better convergence with momentum-based aggregation but it still needs more rounds. The federated model of AE–Transformer converges faster and is more stable than the other models, achieving a higher performance for limited rounds communication. This gain is due to the hybrid feature learning approach and the FedProx optimization method, which better accounts for client dissimilarity.

Table 7: Privacy and Security Mechanism Comparison

Method	**Federated Learning**	**Differential Privacy**	**Secure Aggregation**	**Attack Robustness**
Paper [10]	No	No	No	Medium
Paper [32]	Yes	Partial	Yes	Good
Paper [33]	No	No	No	Medium
Proposed Method	**Yes**	**Yes**	**Yes**	**Very High**

Table 7 compares the existing ID system security and privacy mechanisms. Centralized methods do not have privacy inherently and are thus sus-ceptible to data disclosure. Although several federated models have some privacy-preserving methods, they usually do not have differential privacy or secure aggregation. The approach combines differential privacy and secure aggregation to ensure that local client updates are kept secret. This combination increases the security of the system and makes the model more applicable to IoT-based real world scenarios containing sensitive information.

Table 8: Model Complexity and Computational Efficiency

Method	**Model Type**	**Parameters**	**Edge Suitability**	**Training Time**	**Communication Cost**
Paper [10]	Traditional ML ensemble	Low	High	Low	Not applicable

Paper [32]	Deep Autoencoder	Medium	Good	Moderate	Medium
Paper [33]	CNN-based IDS	High	Moderate	High	High
Proposed Method	**AE + Transformer Hybrid**	**Moderate-High**	**Good**	**Moderate**	**Optimized via FedProx**

Table 8: Complexity and effectiveness comparison of intrusion detection models. However, the features learning ability of these machine learning methods is limited to some extent. IDS models involving deep learning tend to improve the detection accuracy at the price of sacrificing computational resources. The AE–Transformer model balances performance and complexity by using lightweight autoencoder feature extraction and efficient Transformer-based temporal modeling. Moreover, the FedProx optimization lowers the communication costs, which makes this model fit for edge-driven IoT scenarios.

Table 9: Class-wise Detection Capability (Test Results)

Method	Normal	DoS	DDoS	Injection	Ransomware	XSS	Overall Detection
Paper [10]	93.2	91.4	92.0	90.6	89.3	90.1	92.1
Paper [32]	95.4	94.2	95.1	93.8	92.6	93.4	95.0
Paper [33]	96.8	95.9	96.4	95.3	94.7	95.1	96.2
Proposed Method	**98.1**	**97.4**	**97.8**	**96.9**	**96.2**	**96.5**	**97.23**

The class-wise detection results are shown in Table 9 for various types of intrusions. The performance of the proposed method demonstrates superior detection rates in all attack classes compared to baseline methods. The existing approaches perform poorly against sophisticated attack strategies like ransomware and injection attacks. Federated deep learning methods enhance class-wise detection, however, the hybrid AE–Transformer model outperforms these approaches by successfully exploiting both spatial and temporal information in network traffic that leads to a balanced classificatory pattern among classes making it more accurate.

Table 10: Federated Learning Robustness under Data Heterogeneity

Method	Non-IID Handling	Aggregation	Performance Drop (Non-IID)	Final Accuracy (%)
Paper [32] FedAvg	Weak	Average	−4.5%	93.4
Paper [32] FedAvgM	Better	Momentum-based	−2.1%	95.05
Proposed FedProx-AE-Transformer	**Strong**	**Proximal aggregation**	**−0.9%**	**97.23**

Table 10 presents an analysis of the robustness of federated learning algorithms under non-IID data distributions. The performance of the standard FedAvg aggregation algorithms can be severely degraded when there is a high heterogeneity of client data distributions. Momentum aggregation leads to more stable training but still suffers from performance drops. The suggested FedProx-based AE–Transformer model exhibits higher robustness, showing very small drops in accuracy across all the heterogeneities. This enhancement is produced by the proximal regularization term, which stabilizes local updates and preserves consistent global model performance under various client circumstances.

6. Conclusion

In this paper we proposed a privacy-preserving federated intrusion detection framework that using a hybrid model of Autoencoder – Transformer. The proposed approach combines representation learning, temporal modeling and secure federated optimization involving FedProx, differential privacy and secure aggregation. Experimental analysis on the ToN-IoT-IDS dataset shows the effectiveness of our method outperforms other centralized and federated methods, by achieving an overall accuracy score of 97.23%, macro-F1 with 96.5% and ROC-AUC with 97.3%. The training and validation responses verify the robust convergence, and corresponding confusion matrices reveal good class-wise detection performance. Furthermore, it provides data privacy by storing sensitive sensor

readings locally in IoT devices. In summary, the proposed system offers a scalable, and privacy-preserving IDS for distributed IoT systems with robustness. In the future, we will work on intelligent client selection, light-weight edge deployment, and real-time monitoring of attacks in dynamic IoT networks.

Reference

1. Bukhari, S. M. S., Zafar, M. H., Abou Houran, M., Moosavi, S. K. R., Mansoor, M., Muaaz, M., & Sanfilippo, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. *Ad Hoc Networks*, *155*, 103407.
2. Alazab, A., Khraisat, A., Singh, S., & Jan, T. (2023). Enhancing privacy-preserving intrusion detection through federated learning. *Electronics*, *12*(16), 3382.
3. Rashid, M. M., Khan, S. U., Eusufzai, F., Redwan, M. A., Sabuj, S. R., & Elsharief, M. (2023). A federated learning-based approach for improving intrusion detection in industrial internet of things networks. *Network*, *3*(1), 158-179.
4. Torre, D., Chennamaneni, A., Jo, J., Vyas, G., & Sabrsula, B. (2024). Towards enhancing privacy-preservation of a federated learning cnn intrusion detection system in IoT: method and empirical study. *ACM Transactions on Software Engineering and Methodology*.
5. Javeed, D., Saeed, M. S., Adil, M., Kumar, P., & Jolfaei, A. (2024). A federated learning-based zero trust intrusion detection system for Internet of Things. *Ad Hoc Networks*, *162*, 103540.
6. Siddiqui, S. Y., Khan, M. F., Tehseen, R., Rehman, U., Tauheed, N., & Javaid, M. T. (2024). Cyber Security Intrusion Detection Scheme for Malicious Traffic in IoT using Federated Learning. *International Journal for Electronic Crime Investigation*, *8*(4).
7. Almeida, L., Rodrigues, P., Teixeira, R., Antunes, M., & Aguiar, R. L. (2024, June). Privacy-Preserving Defense: Intrusion Detection in IoT using Federated Learning. In *2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON)* (pp. 908-913). IEEE.
8. Ioannou, I., Nagaradjane, P., Angin, P., Balasubramanian, P., Kavitha, K. J., Murugan, P., & Vassiliou, V. (2024). GEMLIDS-MIOT: A Green Effective Machine Learning Intrusion Detection System based on Federated Learning for Medical IoT network security hardening. *Computer Communications*, *218*, 209-239.
9. Ioannou, I., Nagaradjane, P., Angin, P., Balasubramanian, P., Kavitha, K. J., Murugan, P., & Vassiliou, V. (2024). GEMLIDS-MIOT: A Green Effective Machine Learning Intrusion Detection System based on Federated Learning for Medical IoT network security hardening. *Computer Communications*, *218*, 209-239.
10. Lazzarini, R., Tianfield, H., & Charissis, V. (2023). Federated learning for IoT intrusion detection. *Ai*, *4*(3), 509-530.
11. Dhakal, R., Raza, W., Tummala, V., & Kandel, L. N. (2024). Enhancing Intrusion Detection in IoT Networks Through Federated Learning. *IEEE Access*.
12. Bhavsar, M., Bekele, Y., Roy, K., Kelly, J., & Limbrick, D. (2024). FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT. *IEEE Access*.
13. Hamad, N. A., Bakar, K. A., Qamar, F., Jubair, A. M., Mohamed, R. R., & Mohamed, M. A. (2025). Systematic Analysis of Federated Learning Approaches for Intrusion Detection in the Internet of Things Environment. IEEE Access.
14. Buyuktanir, Busra, Şahsene Altinkaya, Gozde Karatas Baydogmus, and Kazim Yildiz. "Federated learning in intrusion detection: advancements, applications, and future directions." *Cluster Computing* 28, no. 7 (2025): 473.
15. Albanbay, Nurtay, Yerlan Tursynbek, Kalman Graffi, Raissa Uskenbayeva, Zhuldyz Kalpeyeva, Zhastalap Abilkaiyr, and Yerlan Ayapov. "Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study." *Journal of Sensor and Actuator Networks* 14, no. 4 (2025): 78.
16. Hernandez-Ramos, Jose Luis, Georgios Karopoulos, Efstratios Chatzoglou, Vasileios Kouliaridis, Enrique Marmol, Aurora Gonzalez-Vidal, and Georgios Kambourakis. "Intrusion Detection based on Federated Learning: a systematic review." *ACM Computing Surveys* 57, no. 12 (2025): 1-65.
17. Anwer, Raja Waseem, Mohammad Abrar, Mehar Ullah, Abdu Salam, and Faizan Ullah. "Advanced intrusion detection in the industrial Internet of Things using federated learning and LSTM models." *Ad Hoc Networks* (2025): 103991.
18. Anwar, Raja Waseem, Mohammad Abrar, Abdu Salam, and Faizan Ullah. "Federated learning with LSTM for intrusion detection in IoT-based wireless sensor networks: a multi-dataset analysis." *PeerJ Computer Science* 11 (2025): e2751.
19. Karunamurthy, A., K. Vijayan, Pravin R. Kshirsagar, and Kuan Tak Tan. "An optimal federated learning-based intrusion detection for IoT environment." *Scientific Reports* 15, no. 1 (2025): 8696.
20. Devi, Manu, Priyanka Nandal, and Harkesh Sehrawat. "Federated Learning-Enabled Lightweight Intrusion Detection System for Wireless Sensor Networks: A Cybersecurity Approach Against DDoS Attacks in Smart City Environments." *Intelligent Systems with Applications* (2025): 200553.
21. Chandu, G., Karthik, T., & Parag, B. (2025). Federated Learning for Distributed IoT Security: A Privacy-Preserving Approach to Intrusion Detection. *IEEE Access*.
22. Ahmad, I. (2025, February). A Federated Learning Approach to Strengthen IoT-Based Intrusion Detection Against Low-Rate DDoS Attacks. In *2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)* (pp. 1-6). IEEE.
23. Torre, D., Chennamaneni, A., Jo, J., Vyas, G., & Sabrsula, B. (2025). Toward enhancing privacy preservation of a federated learning cnn intrusion detection system in iot: method and empirical study. *ACM Transactions on Software Engineering and Methodology*, *34*(2), 1-48.

24. Bensaid, R., Labraoui, N., Saidi, H., & Bany Salameh, H. (2025). Securing fog-assisted IoT smart homes: A federated learning-based intrusion detection approach. *Cluster Computing*, *28*(1), 50.
25. Govindaram, A., & A, J. (2025). Flbc-ids: a federated learning and blockchain-based intrusion detection system for secure iot environments. *Multimedia Tools and Applications*, *84*(17), 17229-17251.
26. Samanta, S., Yousuf, R., Sharma, N., & Kumar, M. (2025, August). Federated Ensembled Learning for Intrusion Detection Using IoT Network. In *2025 12th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP)* (pp. 1-5). IEEE.
27. Olanrewaju-George, B., & Pranggono, B. (2025). Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. *Cyber Security and Applications*, *3*, 100068.
28. Latif, N., Ma, W., & Ahmad, H. B. (2025). Advancements in securing federated learning with IDS: a comprehensive review of neural networks and feature engineering techniques for malicious client detection. *Artificial Intelligence Review*, *58*(3), 91.
29. Ullah, I., Deng, X., Pei, X., Mushtaq, H., & Khan, Z. (2025). Securing internet of vehicles: a blockchain-based federated learning approach for enhanced intrusion detection. *Cluster Computing*, *28*(4), 256.
30. Soomro, I. A., Rehman, H. U., Hussain, S. J., Latif, S., Mujlid, H., Mohsin, S. M., & Maple, C. (2025). ROCHE: A Robust and End-to-End Privacy-Preserving Federated Learning Framework for Intrusion Detection in Industrial Internet of Things. *IEEE Internet of Things Journal*.
31. Al Tfaily, F., Ghalmane, Z., Brahmia, M. E. A., Hazimeh, H., Jaber, A., & Zghal, M. (2025). Graph-based federated learning approach for intrusion detection in IoT networks. *Scientific Reports*, *15*(1), 41264.
32. Olanrewaju-George, B., & Pranggono, B. (2025). Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models. *Cyber Security and Applications*, *3*, 100068.
33. 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.
34. Puviarasu, A., & Sudha, V. K. (2026). Enhanced IoT security: privacy-preserving federated learning model for accurate, real-time intrusion detection across devices. Ain Shams Engineering Journal, 17(1), 103866.
35. Nguyen, T. D., Alazab, A., Khraisat, A., & Jan, T. (2026). Feature reduction in federated learning for intrusion detection in IoT networks. Cybersecurity, 9(1), 102.
36. Jadhav, D. S., Jadhav-Mane, S. R., Bamane, K. D., Mohite, S. S., Patankar, A. J., & Gaikwad, A. S. (2026). Intrusion Detection in Networks using Ensemble-Based Deep Federated Learning. Knowledge-Based Systems, 115491.
37. Mankotia, S., Conte de Leon, D., & Rimal, B. P. (2026). FedPrIDS: Privacy-Preserving Federated Learning for Collaborative Network Intrusion Detection in IoT. *Journal of Cybersecurity and Privacy*, *6*(1), 10.
38. Liang, Y., & Luo, M. (2026). Optimization of distributed network intrusion detection system based on internet of things and federated learning. *Discover Internet of Things*, *6*(1), 3.
39. Liu, T., & Dib, O. (2026). Advanced intrusion detection for IoT devices using federated deep learning: T. Liu, O. Dib. International Journal of Information Security, 25(1), 19.
40. Saleem, A., & Hamouda, W. (2026). Lightweight Federated Few-Shot Learning-Based Network Intrusion Detection for Resource-Constrained IoT Devices. IEEE Internet of Things Journal.