Wasit Journal of Engineering Sciences
Journal homepage: https://ejuow.uowasit.edu.iq
Vol. 14 No. 3 September 2026, pp.82-105
https://doi.org/10.31185/ejuow.v14i3.997

ISSN 2305-6932
Original Research

A Privacy-Preserving Intrusion Detection System for IoT Networks Using Federated Learning

Ali Abd Alraheem Abd[1], Ali Obeid[1], Bassam Noori Shaker[1]

[1] Faculty of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq

Corresponding Author Email: **master.student241@qu.edu.iq**

Received Jul. 21, 2026 | Revised Aug. 19, 2026 | Accepted Aug. 22, 2026 | Online Sep. 1, 2026

ABSTRACT

With the increasing presence of IoT devices in the real world, this widespread presence leads to serious security challenges related to the privacy of these devices' data. Despite the important role of intrusion detection system (IDS) and its ability to identify malicious security activities in traditional centralized learning solutions that rely on collecting raw data from devices and sharing it directly to a central server, these solutions may raise concerns regarding data privacy and an increase in communication overhead. To address these challenges, this study proposes a privacy-preserving intrusion detection system using federated learning (FL) that enables distributed IoT devices to engage in collaborative learning, without share the raw data, only updates, with taking into account the preservation of data privacy, all existing IoT clients independently train the Multilayer Perceptron (MLP) models on their own data only, and then share only the models updates with the central server, the Federated Averaging (FedAvg) algorithm is used within the central server to aggregate the updates and create a global model. The proposed framework was evaluated using the NF-BoT-IoT dataset. The experimental results demonstrate that the proposed lightweight IDS achieves an accuracy of 83.21% and an F1-score of 81.78 %, with performance comparable to the centralized learning approach while preserving client data privacy. In addition, the proposed hybrid feature selection approach reduces the feature space from eight to five features, resulting in measurable computational benefits. In particular, leading to a 15.21% reduction in inference time, a 4.59% reduction in training time, and a 2.78% reduction in memory usage. These results demonstrate that the proposed framework not only maintains competitive detection performance and data privacy but also reduces computational and memory requirements, supporting its suitability as an effective and lightweight IDS for resource-constrained IoT environments.

Keywords: Internet of Things (IoT), Intrusion Detection System (IDS), Federated Learning (FL), Multilayer Perceptron (MLP)

1. Introduction

The Internet of Things (IoT) has led to a significant increase in the use of connected devices, including smartwatches, smart homes, healthcare systems, industrial automation, transportation, and other IoT-enabled applications. IoT is a network of physical devices that can be monitored and managed through the Internet [1]. As recently reported, billions of IoT devices are connected worldwide, and this number is growing annually, it is estimated that by the year 2030, there will be a global count of active IoT devices close to 25.44 billion [2]. Despite the numerous advantages of the IoT, the limited computing power, memory, and battery. The growing

interconnectivity of IoT devices exposes them to malicious activities, such as Denial-of-Service (DoS) attacks, Distributed Denial-of- Service (DDoS) attacks, reconnaissance attacks, and malware attacks. Therefore, maintaining the security and privacy of IoT networks has become a critical research challenge.

An Intrusion Detection System (IDS) functions as a security mechanism or software developed to oversees the activities within an IoT network. By recognizing unusual patterns or irregular activities among IoT devices, IDS can identify potential threats and intrusion attempts [3]. An IDS can be categorized into two main types: signature-based and anomaly-based methods. The signature-based approach employs established rules or signatures to detect recognized attacks, whereas the anomaly-based method leverages statistical or machine learning techniques to learn the characteristics of both legitimate and malicious data during a training or offline phase, subsequently identifying attacks in real-time incoming traffic during the prediction or online phase [4]. Most IDS's currently existing depend on centralized learning [5], where raw data from the network traffic that is gathered from distributed IoT devices is sent to a central server for model training. While this method usually gives high detection accuracy, it also has significant issues with user data privacy, communication overhead, and a single point of failure.

Federated Learning (FL) is a decentralized machine learning approach that allows multiple clients to collaboratively learn to create a global model without sharing the raw data. Rather, local updates to the model are carried out by each client, based on its own data, and then sent to a central server for aggregation. This distributed learning process improves data privacy and still achieves competitive model performance, which makes FL an ideal choice for intrusion detection in distributed IoT systems. A significant advantage of Federated Learning (FL) lies in its ability to maintain a balance between accuracy and privacy. This balance is accomplished using aggregation methods like Federated Averaging (FedAvg), which enables various clients to independently train their local models and transmit averaged model updates to a central server, all while keeping raw data privacy [6].

This paper presents a privacy-preserving intrusion detection using FL framework for IoT networks. The model in the proposed framework is the Multilayer Perceptron (MLP) model, which is trained IoT clients based on their private network traffic. The raw data is not sent to a central server, but rather only model parameters are sent and aggregated with the FedAvg algorithm to create a global model. The proposed framework is evaluated on multi-class classification detection on NF-BoT-IoT dataset. The experimental results show the feasibility of proposed federated learning framework for achieving competitive detection performance while maintaining the privacy of the user data, demonstrating its potential for secure and distributed IoT networks.

2. Related works

This section reviews recent studies on intrusion detection systems that utilize federated learning, highlighting the proposed models, datasets, methodologies used, experimental results, and recognized limitations.

Nguyen et al. [7] proposed DIoT, a federated self-learning anomaly detection system that detects malicious activities in IoT networks while maintaining user privacy. Unlike existing centralized intrusion detection systems, DioT allows IoT gateways to train local models of anomaly detection using local network traffic and only send updates about the models to a central server via a federated learning approach. The proposed system utilizes Gated Recurrent Units (GRUs) to analyze traffic sequences and detect deviations, and the Federated Averaging (FedAvg) algorithm for aggregating local models into a global model. This model has been assessed across 30 devices and classified into 23 types, attaining a detection rate of 95.6% and identifying compromised devices within 257 milliseconds, all without generating any false alarms during evaluations conducted in actual smart home environments. The experimental results showed that the federated approach could provide similar detection performance to the centralized learning while reducing privacy risks by keeping raw network traffic on local devices. Furthermore, the robustness of the model is heavily reliant on the security of the gateway; any breach could lead to a failure of the entire system. Additionally, there is a risk of model poisoning attacks occurring within a federated learning framework.

Zhao et al. [8] proposed an intelligent intrusion detection framework based on FL coupled with a long short-term memory (LSTM) framework to achieve accurate intrusion detection while protecting the privacy of users. The proposed framework uses a decentralized learning architecture in which multiple user servers independently train local LSTM models on their own data. A weighted parameter aggregation strategy based on FedAvg is used to generate a global model. The MD5 check is used when uploading to ensure the integrity of the model weight parameter information. The proposed FL-LSTM model was compared with centralized LSTM (CL-LSTM), centralized CNN (CL-CNN), and independently trained local LSTM models. The experiments were conducted using an enhanced version of the SEA (UNIX Shell Commands) dataset. Experimental results demonstrated that the proposed FL-LSTM model achieved an accuracy of 99.21% and an F1-score of 99.21%, while the centralized LSTM achieved a slightly higher accuracy of 99.51%. The results indicate that the federated model maintained detection performance very close to centralized learning while significantly outperforming independently trained local models. However, the research utilized a tailored command-sequence dataset instead of a contemporary network traffic dataset specific to IoT environments, which restricts the applicability of the findings to actual IoT intrusion detection scenarios.

Rahman et al. [9] suggested an intrusion detection system for the IoT environment that utilizes federated learning (FL), To address the privacy and scalability challenges of centralized IDS. The suggested architecture employs the Federated Averaging (FedAvg) algorithm to consolidate model updates from the devices involved. To assess the efficacy of this approach, the authors analyzed three distinct learning paradigms: centralized learning, on-device (self-learning), and federated learning, across various data distribution scenarios, such as attack-based, balanced, and random data distributions. The experiments utilized the NSL-KDD dataset. Experimental findings indicated that the centralized model attained an accuracy of 83.09%, while the federated model demonstrated similar effectiveness, achieving accuracy rates between approximately 80.45% and 81.48% after five communication rounds. Additionally, the federated model consistently surpassed the self-learning method in all assessed scenarios. The research found that federated learning can achieve intrusion detection results comparable to those of centralized learning, all while greatly enhancing data privacy. However, the suggested framework depends on the NSL-KDD dataset, which is relatively outdated and may not adequately represent modern IoT attack scenarios.

Novikova et al. [10] proposed architecture designed for cyber-physical systems, particularly for industrial applications like water treatment systems, which allow for collaborative model training without exchanging raw data. The researcher used a Gradient Boosting Decision Tree (GBDT) classifier, which was trained using the SecureBoost federated learning algorithm. To make the model aggregation more private, homomorphic encryption was used to communicate the model parameters safely and to ensure that the information is not transmitted to the attacker. The model is evaluated on the Secure Water Treatment (SWaT) dataset. To simulate vertical federated learning, the dataset was partitioned into six technological process facilities and our class labels were stored by the aggregation server. The experimental results showed that the proposed federated model had an accuracy of 99% with 30 decision trees. While the proposed framework effectively demonstrates, the cost of symmetric encryption for training and inference time is quite high, which is a problem in real-time scenarios. Moreover, the study relies on the SWaT industrial control system dataset instead of an IoT penetration dataset.

Campos et al. [11] introduced a decentralized approach called Federated Learning (FL), which shares just the partial updates to the central entity, and evaluated the use of Federated Learning in IoT with consideration to the impact of non-IID data. This paper used the CIC-ToN-IoT Dataset, a version of the ToN-IoT database whose features were extracted using CICFlowMeter to generate 83 features. For the ToN-IoT dataset the paper proposes three scenarios with different partitions and processing of the data: in the first one network flows are partitioned based on their destination IP address; the second is based on the balance of attack types among the clients; and the third is a hybrid approach, where a compromise between the two is found using the Shannon entropy. For all scenarios, the FedAvg and Fed+ aggregation methods were used. For mixed scenario, using Fed+ for these rounds, accuracy is as low as 0.8498 and eventually rises to 0.8876 at 50 rounds and stays stable

until the end at 0.8869. After about 40 rounds, Fed+ overtakes the accuracy for the distributed case 0.877. The behavior of FedAvg is worse than the distributed case though. The study identifies several research challenges that require further investigation, including deploying FL on resource-constrained IoT devices.

Khraisat et al. [12] proposed the Privacy-Enhanced IoT Defense System (PEIoT-DS), a federated learning-based IDS designed to address privacy and normal performance issues in centralized learning in IDS. This model aims to improve the privacy of data and provide high-performance IDS. In this paper, the authors compared and evaluated both Federated Averaging (FedAvg) and Federated Averaging with Momentum (FedAvgM) as aggregation algorithms. Model updates during communication will be secured by Secure Multiparty Computation (SMPC) and Differential Privacy (DP). The detection accuracy of FedAvg was 94.05% on the N-BaIoT dataset, while FedAvgM achieved 95.05% detection accuracy, which reduced the false positive rate by 3.9% and converged faster under the non-IID data distribution. The authors found that in heterogeneous IoT environments, both the stability of the learning process and the effectiveness of communication are enhanced by momentum-based aggregation. However, communication overhead and local training costs may still be high, especially for resource-limited IoT devices.

Although previous studies have shown promising performance in intrusion detection using federated learning. Several limitations persist. Many studies have concentrated mainly on improving accuracy, neglecting feature dimensionality reduction. Additionally, some current datasets may also not accurately reflect modern IoT network traffic or new cyber threats. Therefore, this study aims to conduct a lightweight and privacy-preserving intrusion detection system (IDS) utilizing federated learning, which is suitable for IoT devices' resource constraints and can provide high detection performance and minimal communication overhead, all while maintaining the security of network traffic data.

Table 1. Reviews recent works of federated learning privacy IDS.

Table 2: Summary of Related Work.

Authors (year)	Technique	Dataset	Results	Limitation
Nguyen et al. (2019)	DIoT (GRU + FedAvg)	Custom IoT dataset (30 devices, 23 types)	95.6% detection rate; 257 ms	Gateway compromise and model poisoning risks.
Zhao et al. (2020)	FL-LSTM + FedAvg + MD5	Enhanced SEA dataset	99.21% Accuracy; 99.21% F1-score.	Non network traffic IoT dataset used
Rahman et al. (2020)	FL + FedAvg	NSL-KDD	FL: 81.48%, Centralized: 83.09%.	Uses outdated NSL-KDD dataset.
Novikova et al (2022)	GBDT + SecureBoost + Homomorphic Encryption	SWaT	99% Accuracy.	High computation and encryption overhead; not suitable for real-time IDS.
Campos et al. (2022)	Multinomial Logistic Regression + FedAvg/Fed+	CIC-ToN-IoT	Fed+ achieved 88.76% accuracy and outperformed FedAvg under non-IID.	Resource constraints deployment challenges and the impact of non-IID data

Khraisat et al. (2025)	PEIoT-DS (FedAvg/FedAvgM + SMPC + DP)	N-BaIoT	FedAvgM: 95.05%, FedAvg: 94.05%. .	Communication overhead and local training cost.

3. Theoretical Background

This section provides a detailed overview of the concept of IoT networks and security challenges, followed by a review of the concept of IDS and their types. Furthermore, it discusses an overview for DL and MLP. The chapter covers the feature selection, introduces the concept of decentralized learning, federated learning, from centralized to distributed learning. In addition, it describes the FL architecture, privacy-preserving mechanisms, and the major challenges of FL in IoT networks, including communication overhead and non-IID data distribution. Lastly, the chapter introduces the performance measures employed to evaluate the performance of the proposed method.

3.1 IoT Networks and Security Challenges

IoT is a platform that makes things smart, processing smart and communication smart. IoT devices come with limited memory, low processing capability, and battery constraints, making the implementation of conventional security mechanisms difficult. these constraints, the effectiveness of security solutions for IoT infrastructures is heavily affected by these constraints in terms of scalability and distributed nature. and the performance efficiency of the solution [13]. In addition, IoT environments are composed of a vast number of diverse devices connected to each other via different protocols and possibly placed in unattended or public areas, which renders them susceptible to various cybersecurity threats. such as DoS, DDoS, man-in-the-middle attacks, malware and botnets, spoofing, and reconnaissance attacks [14], and data theft. These attacks may affect service confidentiality, integrity, and availability, causing service disruptions, access to services, and data leakage.

Security attacks can be classified into two main types: First, active attacks occur during the runtime of a system and have the potential to disrupt or damage the physical device. These types of attacks are generally more challenging to execute and detect than passive ones. A well-known example of an active attack is Denial of Service. Other instances include packet replay, spoofing, and message modification. Second, passive attacks Focus on observing and monitoring information related to a specific target without altering the data. Attackers remain undetected while maintaining an open communication channel to gather information. Common threats in this category include eavesdropping, network mapping, and traffic analysis [15]

3.2 Intrusion Detection Systems (IDS)

An Intrusion Detection System (IDS) is a security mechanism that mainly works at the network layer of an IoT system. To get the maximum performance from an IoT system, the IDS needs to analyze data packets at different layers of the IoT network with various protocol stacks. Additionally, it should be capable of generating real-time responses and adjusting to diverse technologies pertinent to the specific IoT environment [16].

IDS can be categorized based on the identified locations of detection [17], into two main types; Host intrusion detection system (HIDS) and network intrusion detection system (NIDS), Based on the identified locations of detection HIDS are those that operate on an individual workstation, whereas NIDS refer to systems that operate independently as devices within a network [18].

IDSs can be effectively tailored to address the expanding range of potential attacks. Notable advancements have been made in identifying harmful traffic through machine learning techniques. Nonetheless, current IDS implementations depend on network devices with substantial computational power for the ongoing training and updating of intricate network models. This reliance can hinder the overall efficiency and defensive strength of intrusion detection systems [19], primarily due to IoT devices resource constraints such as limited memory, low

processing capability, and battery constraints, making it challenging to deploy IDS models. The IDS can be classified broadly into three types based on detect malicious activities, Signature , Anomaly, and Hybrid.

3.3 Deep Learning for IDS

The rapid growth of IoT devices has led to significant security challenges due to the large volume of traffic they generate, their limited resources, diverse nature, and constant connectivity. These factors render IoT environments particularly susceptible to various cyber threats, highlighting the need for effective and intelligent IDS. To address these issues, researchers have increasingly turned to artificial intelligence, with a specific focus on Deep Learning (DL), to create robust IDS that can protect IoT devices by automatically learning intricate traffic patterns and effectively differentiating between legitimate and harmful activities.

DL offers a compelling approach for developing IDS in the IoT network. DL models utilized for traffic classification can be grouped into three main types: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN)[20].

A DNN is designed with a multi-layer framework comprising three specific layers: input layer, hidden layers, and an output layer (see Fig 1).

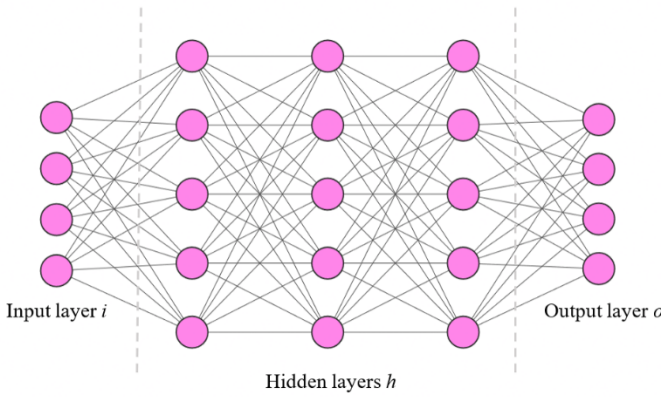

Figure 1. Specified DNN Layers

3.4 Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is one of the most frequently used types of feedforwards DNNs. It has an input layer, one or more hidden layers, and an output layer, with full connection between neurons in the adjacent layers. The hidden layers are used to learn the features in the input layer, and the nodes in the output layer are output classes of the database [21] . The hidden layer employs the Rectified Linear Unit (ReLU) activation function, as outlined in Equation 3.1. In contrast, SoftMax activation functions are applied within the output layer to suit multi-class classification tasks, as indicated in Equation 3.2. The training process for this network utilizes the Adam optimizer (Adaptive Moment Estimation), with Categorical Cross-entropy functioning as the loss metric according to Equation 3.3.

$$\boldsymbol{a_{relu}} = \mathbf{max}(\mathbf{0}, \boldsymbol{z}) \tag{3.1}$$

Where $\boldsymbol{z}$: Functions as the input for the neuron.

$$\boldsymbol{a_{soft}} = \frac{\boldsymbol{e^{z_i}}}{\sum_{\boldsymbol{i}}^{\boldsymbol{J}} \mathbf{1} \boldsymbol{e^{z_i}}} \tag{3.2}$$

Where **J**: Represents the class number, $\mathbf{z_i}$: Indicating the output value.

$$Loss = -\sum_{i=1}^{output\ size} y_i . log\ \hat{y}_i \qquad (3.3)$$

Where y_i: Denotes the actual true label, while $\hat{y}_i$ signifies the predicted probability for class i

3.5 Feature selection

IDS have difficulties identifying attacks, because of the huge amount of data created in the internet. Furthermore, irrelevant and redundant features can hinder the effectiveness of IDS by impacting both detection rates and processing times [22]. Feature selection is a critical approach that aims to enhance detection capabilities while also improving the clarity of results. Typically, Feature selection methods that are typically used in classification problems can be summarized into three categories: filter methods, wrapper methods, embedded methods [23].

3.6 Centralized learning

The centralized learning (CL) method is a costly traditional approach in which data is gathered and preprocessed before being transmitted to a centralized server that possesses substantial computational resources as shown in fig 2. This server aggregates data from different IoT devices and subsequently utilizes it to train a single model once [24]. The server aggregates the data into an extensive dataset to train machine learning (ML) and DL models.

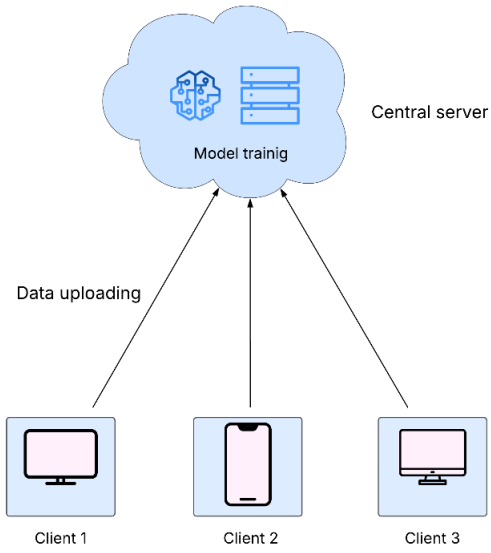

Figure 2. Centralized Learning Architecture

3.7 Distributed learning

Distributed learning as an alternative to CL, seeking scalability by enhancing computational efficiency through the reduction of total training duration and leveraging the processing power of distributed devices instead of relying solely on a central server. FL originated as a privacy-preserving mechanism to enhance privacy by allowing the learning process to take place across distributed mobile clients while only sharing the model

parameters with a central server. This approach helps safeguard the clients' data from unauthorized access by third parties, as the computational burden is shifted to them. The central server subsequently functions as a controller, responsible for selecting clients, aggregating model parameters, and managing schedules. Consequently, in FL, it is typically the ML model that is transmitted over the network rather than the data itself, distinguishing it from CL [25]. For environments where privacy is a key concern, data is never sent to a central server, the FL framework is made specifically for such scenarios. Federated learning is appropriate for distributed IoT applications where collaborative learning and data privacy are needed, as only local updates to model parameters are sent in the update process.

3.8 Federated Learning

FL is an enhancement of distributed learning, allowing multiple models to be trained collaboratively without sharing the raw data. In FL each client independently trains on its own local data by using the DL model. Each client sends its updated parameters to the server for the aggregation process [26] , updated parameters from all the clients participating in the process are aggregated with an aggregation algorithm such as FedAvg to form an updated global model on the server.as shown in Equation 2.5.

$$\boldsymbol{w_{t+1}} = \sum_{\boldsymbol{k=1}}^{\boldsymbol{k}} \frac{\boldsymbol{nk}}{\boldsymbol{n}} \boldsymbol{w_{t+1^k}} \tag{3.4}$$

Where:

$\boldsymbol{nk}$: number of training samples on client k.

$\boldsymbol{w_{t+1}}$: updated global model after aggregating all local models.

$\boldsymbol{n}$: total number of training samples across all participating clients.

$\boldsymbol{w_{t+1^k}}$: local model weights of client *k* after training during communication round.

This global model is then communicated to all clients and the process repeated through several rounds of communication until either the predetermined number of rounds is completed or the global model achieves convergence.

In contrast to the Distributed learning, FL assumes that the data are permanently located on the local devices. This characteristic makes FL especially relevant for IoT environments, where sensitive network traffic data continuously generated and privacy concerns can prohibit a centralized approach to data collection. FL guarantees the security of data privacy, minimizes latency, decreases power usage, and facilitates training directly on devices [27]. The architecture of a FL system generally consists of three main components: the central aggregation server, participating clients, and communication rounds, as shown in fig 3.

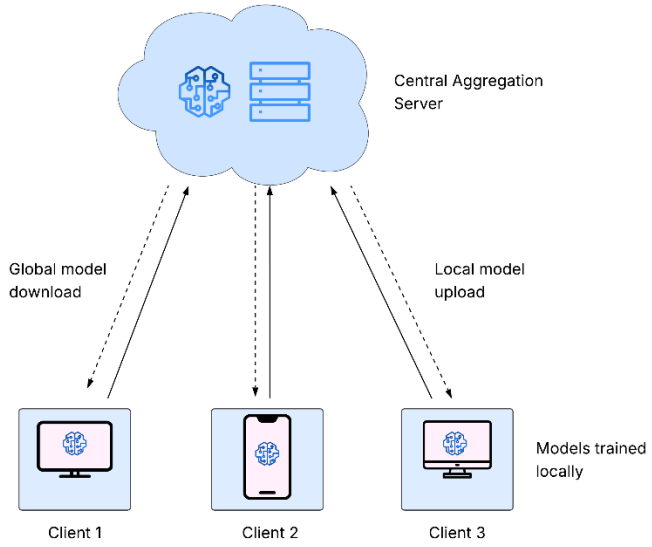

Figure 3. Federated Learning Architecture

3.8.1 Privacy Preservation in Federated Learning

Preserving privacy is amongst the main purposes behind the implementation of FL in IoT networks. FL offers a security mechanism designed to safeguard data privacy, ensuring the secure transfer of models while maintaining the confidentiality of local sensitive data. Nonetheless, despite the safeguards provided by FL's security mechanism, numerous attack vectors exist that can target the FL system. These attacks can undermine the system's reliability and pose risks to the data privacy of its participants [28].

In IoT based IDSs, preserving privacy is a critical concern due to the fact that network traffic can consist of sensitive information related to user actions, device activities, communication patterns, and organizational infrastructure. It could also breach privacy laws and introduce the danger of data leakage or unauthorized access if the data is shared with a centralized server. FL allows for collaborative model development while reducing such privacy concerns by keeping training data locally.

FL offers greater security and privacy protections than centralized learning, it is not free from security and privacy risks. The adversarial participants can then try to extract sensitive information from the exchanged model parameters by applying attacks like gradient leakage or try to control the learning process by using model poisoning attacks.

3.8.2 Challenges of Federated Learning in IoT Networks

Although FL presents significant advantages in terms of data privacy and collaborative model training, it also poses several challenges for deploying FL in an IoT network; on the one hand, the cost of communicating per training round is high. The amount of traffic on the server, loss of packet transmission and the time taken to communicate the parameters can vary extensively depending on the available network bandwidth. Also, different kinds of devices in a network means that each device has different computation power. Based on these factors, it is fair to say that in reality the overall throughput of any federated network is probably going to be low. When extended to IDSs these requirements for efficiency and speed become even more important [29].

Another important challenge is the non-independent and identically distributed (non-IID) nature of IoT data, where various devices produce network traffic with distinct characteristics and patterns of attacks. The properties and characteristics of the packets intercepted may be widely varying depending on various entities like usage, time, purpose and many others when it comes to IDS. It is therefore not possible to assume that the data are uniform, identical or independent of each other. These properties can cause a client model to be inconsistent and problematical in the training and aggregation process [29]. The heterogeneity of this data may

hinder model convergence and diminish the accuracy of the global model. Therefore, effectively managing non-IID data is a crucial aspect of FL system to ensure adequate training and resilience against failures.
Moreover, the computational power, memory, and battery capacity of IoT devices are limited, posing challenges for ML model training locally. Furthermore, the dynamic nature of IoT networks, where devices can join or leave the training process often, can impact the stability and efficiency of the FL system.

3.9 Performance Evaluation Metrics

Four fundamental terms are vital for evaluating the effectiveness of intrusion detection systems (IDS). True Positive (TP) signifies the number of attack events that have been correctly identified as threats. True Negative (TN) represents the count of legitimate cases accurately classified as normal. In contrast, False Positive (FP) refers to the number of legitimate instances that have been erroneously flagged as attacks, while False Negative (FN) describes situations where actual attack events are incorrectly classified as normal. These classifications are essential for assessing how proficiently an IDS can identify intrusions and reduce misclassifications. They form the basis for calculating various metrics that measure IDS performance, including:
Accuracy: determines the proportion of instances that are correctly classified in relation to the total instances. It can be computed using Equation (3.5).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (3.5)$$

Recall: calculates the ratio of correctly identified positive cases to the total instances that were meant to be positive. Equation (3.6)

$$Recall = \frac{TP}{(TP+FN)} \quad (3.6)$$

Precision: calculates the ratio of true positive classifications to the total number of instances predicted as positive, as illustrated in Equation (3.7).

$$Precision = \frac{TP}{(TP+FP)} \quad (3.7)$$

Weighted average F1-score: is utilized to assess the overall effectiveness of an IDS model, especially in situations where there is an imbalance among classes, particularly within multiclass classification contexts, as outlined in Equation (3.9).

$$F1 - score = 2 * \frac{(Precision*recall)}{(Precision*recall)} \quad (3.8)$$

All the metrics listed are utilized to evaluate the performance of models in multi-class classification

4. Method

This section presents two main steps: feature selection and FL. The feature selection step provides a hybrid approach that combines MI and forward selection for CL for IDS. The FL step is the second, which presents six steps to develop an efficient and lightweight IDS preserving data privacy and suitable for resource-constrained IoT networks.
After the initial reading, the dataset was preprocessed, the preprocessing stage pipeline comprised various steps, including label encoding, elimination of uninformative features, categorical feature encoding, normalization of features, and splitting the data into training and testing sets. First, label encoding was used to transform the target attack labels into numerical values. Then, the non-informative attributes, such as the identifier-based features, were deleted to avoid potential bias during the training process. Then, the rest of the categorical features were converted to numerical values by Label Encoding. The features that were used were then normalized using the min-max scaler, which will scale all feature values between 0 and 1 so that the features that have different numerical ranges all contribute equally while training the model. Last, the target labels were

converted into one-hot encoded vectors for the multi-class classification task undertaken by the proposed MLP model.

4.1 Proposed Feature Selection Method

This section presents the sequential stages of the proposed hybrid feature selection framework designed to create a lightweight and effective IDS for IoT environments. The primary aim is to determine the most pertinent features, thereby improving detection performance.

As illustrated in algorithm 1, the framework consists of two sequential stages. First, Mutual Information (MI) is employed as a filter method to measure the statistical dependency between each feature and the target class. The features are then ranked in descending order according to their MI scores.

Second, Forward Selection is applied as a wrapper method guided by a MLP classifier. Starting with an empty feature subset, features are iteratively added according to their MI ranking. At each iteration, the MLP is trained using the candidate subset and evaluated using the F1-score. A feature is retained if its inclusion improves or maintains the predefined F1-score threshold (F1 ≥ Th); otherwise, it is discarded. The process continues until the stopping criterion is satisfied. Consequently, the proposed method reduces the initial feature set from eight to five features, providing a more compact feature representation suitable for lightweight IoT IDS deployment.

Algorithm 1: Proposed Hybrid Feature Selection Framework

Input:
Dataset $\boldsymbol{D}$ = *{X, Y}*
Performance threshold $\boldsymbol{Th}$.
Output:
Optimal feature subset **Fs**.
Begin
Step 1: Read the dataset $\boldsymbol{D}$.
Step 2: Perform the preprocessing stage.
Step 3: Compute the Mutual Information (*MI*) score between each feature $\boldsymbol{Xi}$ and the target class $\boldsymbol{Y}$.
Step 4: Sort all features in descending order according to their *MI* scores to obtain the ranked feature list $\boldsymbol{L_{mi}}$
Step 5: Initialize

$$\boldsymbol{Fs} = \emptyset$$

Select the highest-ranked feature.
Add the selected feature to **Fs**.
Train an MLP classifier using **Fs**.
Compute the current weighted F1-score.
Step 6: While unselected features remain do
    Select the next highest-ranked feature.
    Temporarily add the feature to **Fs**
    Retrain the MLP classifier.
    Compute the updated F1-score.
    if Current weighted F1-score ≥ **Th** then
        Permanently keep the feature.
    else
        Remove the feature from **Fs**
    end if
end while
Step 7: Return the selected feature subset **Fs.**
End

4.2 Proposed Federated learning Method

This section presents the main steps of the proposed FL framework to develop and evaluate a privacy-preserving IDS for IoT networks.

As illustrated in Fig 4. The proposed method consists of six steps:

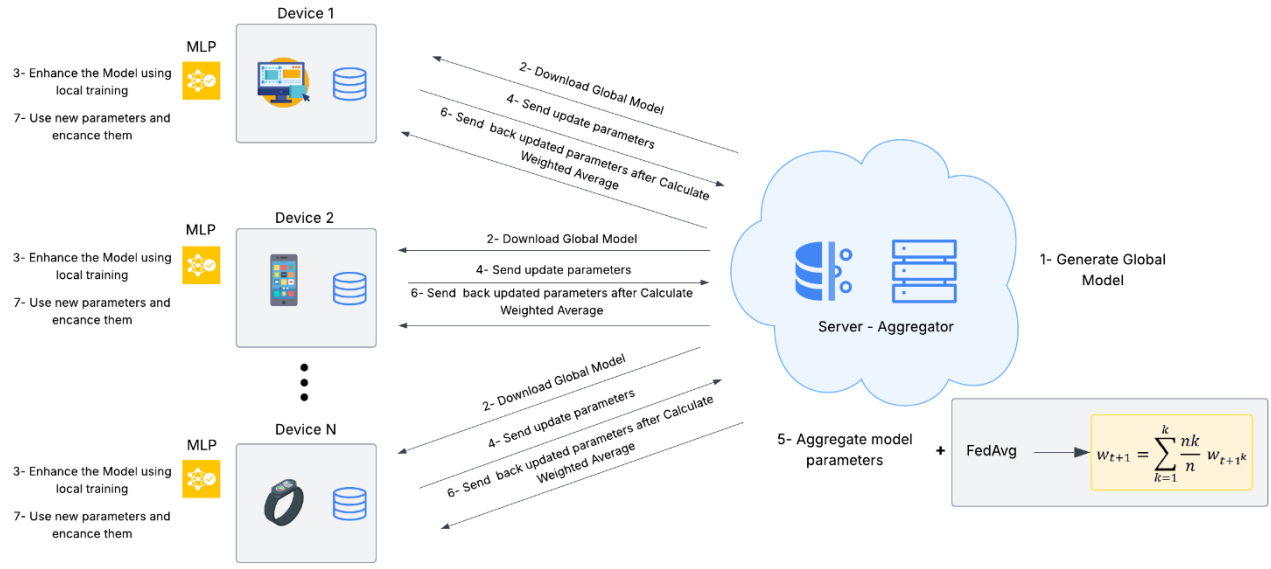

Figure 4. Proposed Federated Learning-based Intrusion Detection System (IDS) for IoT Devices.

The proposed approach consists of main steps:

Step 1: A global intrusion detection model is initially generated at the central aggregation server; the server generates the initial MLP model with randomly initialized parameters and selects the participating IoT clients for the current communication round. Then, the initialized global model is distributed to all IoT clients. In this step, a common starting point is created for the different participating clients to learn from as a global model, and all local models are trained from the same identical initial parameters before collaborative learning begins.

Step 2: The global model is downloaded by the IoT devices from the aggregation server, and each one independently trains on its own local data by using the MLP algorithm. At this stage every client learns the characteristics of its local network traffic data without exposing any private information to other participants.

Step 3: To protect the privacy of the clients, after completing local training, each IoT client updates the model parameters according to its local dataset. Each client sends only updated global model parameters to the central aggregator instead of sending raw data.

Step 4: The central aggregator collects these parameters of all client weights and combines them to generate an improved global model. In this work, the FedAvg algorithm is used to aggregate the received parameters by taking the weighted average of all clients. Our global model is more general with knowledge learned from various distributed IoT devices while maintaining data privacy.

Step 5: Following aggregation, the updated global model is redistributed to all participating IoT devices. Each client enhances its local model parameters with the updated global weights and further refines the model based on any new local data generated, preparing for the next round of federated training. Thus, the intrusion detection model is continuously improving and is not dependent on centralized access to the original data.

Step 6: The process of federated learning is iterative. Consequently, the steps involving local training, sharing of parameters, aggregation of models, and redistribution of the global model are carried out multiple times over several communication rounds. This continues until either the predetermined number of rounds is completed or the global model achieves convergence.
The global model with each round increasingly acquires specific intrusion patterns gathered from various IoT devices, leading to enhanced detection abilities and improved generalization.

5. Experimental results and setup

This section presents the Experimental Setup and Experimental result of the hybrid feature selection and federated learning method.

5.1 Experimental Setup

This section presents the results and analysis of experiments conducted on the proposed IDS framework. It begins by providing an overview of the NF-BoT-IoT dataset employed to evaluate the effectiveness of the recommended FL model, as well as detailing the experimental requirements, including the implementation environment. Finally, the section wraps up with a discussion of model configuration, presentation and analysis of the experimental outcomes, demonstrating the system's effectiveness.

5.1.1 Dataset Description

To assess the proposed intrusion detection model, the NF-BoT-IoT dataset [30] was employed. The Cyber Range Lab at the Australian Centre for Cyber Security (ACCS) developed a realistic network environment that included both standard and botnet traffic. The tools Ostinato and Node-Red were used to generate non-IoT and IoT traffic, respectively. A total of 69.3GB of pcap data was gathered, with the Argus tool utilized to extract 42 initial features from the dataset. The dataset consists of 477 benign flows (0.01%) alongside 3,668,045 attack flows (99.99%), culminating in an overall total of 3,668,522 flows. Furthermore, an IoT NetFlow-based dataset has been established utilizing the BoT-IoT dataset, which is designated as NF-BoT-IoT. Features were sourced from publicly available pcap files, and flows were classified according to their respective attack types. This dataset encompasses a total of 600,100 data flows: 586,241 (97.69%) are categorized as attack samples while 13,859 (2.31%) are benign instances. The NF-BoT-IoT dataset includes 12 features and categorizes attacks into five classes: Benign, Reconnaissance, DDoS, DoS, and Theft, as depicted in Fig 5. During preprocessing, flow identifiers such as IDs, source and destination IP addresses, ports, timestamps, and start/end times were omitted to eliminate bias. As a result, the NetFlow datasets containing only eight features used in classification experiments exhibited similar attack detection performance to that of the original BoT-IoT features. The remaining eight features (l7_proto, in bytes, out bytes, tcp_flags, flow_duration_milliseconds, in_pkts, out_pkts, protocol) are presented in Table 2.

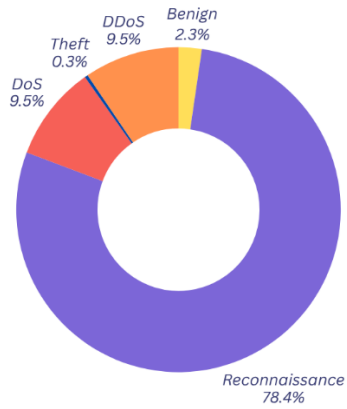

Figure 5. Multi Classes Distribution for NF-BoT-IoT Dataset.
Table 2. Description of NF-BoT-IoT Features

No	Feature	Description
1	IN_BYTES	Incoming byte count
2	OUT_BYTES	Outgoing byte count
3	L7_PROTO	Layer 7 protocol (numeric)
4	TCP_FLAGS	Cumulative of all TCP flags
5	FLOW_DURATION_MILLISECONDS	Flow duration in milliseconds
6	IN_PKTS	Incoming packet count
7	OUT_PKTS	Outgoing packet count
8	PROTOCOL	IP protocol identifier

5.1.2 Hardware and Software Requirements

All the experiments, data preprocessing and model training were carried out using the cloud-computing platform named Google Colaboratory (Colab). The cloud hardware provided had 12.6 GB of System RAM and was accelerated using an NVIDIA Tesla T4 GPU. The framework was implemented using Python programming language (Version 3.12.13). The pipeline used in the experiment was based on a several software libraries: pre-processing with Pandas and NumPy. The following computation of classification evaluation metrics were implemented on Scikit-learn. Finally, the MLP classifier was built and trained using the TensorFlow Library.

5.1.3 Model Configuration

To test the proposed privacy-preserving intrusion detection framework, a FL architecture was put in place where each participating IoT client has a local classifier which is a MLP. Considering that the same neural network architecture was used for all the clients, the aggregation of the parameters was also performed consistently throughout the federated learning process.

The local MLP model (as shown in table 3) comprises an input layer with the selected features by the feature selection method, three fully connected hidden layers with 64, 32 and 32 neurons respectively, and an output layer with a single neuron. In order to add non-linearity and increase the learning capacity of the model, a ReLU activation function (see equation 3.1) was used in each of the hidden layers. The output layer is followed by an activation function named SoftMax (as described in equation 3.2), which is very suitable for multi-class intrusion detection with the number of output neurons being the number of attack classes.

Table 3. MLP Model Architecture

Layer	Number of layers	Neurons	Activation Function
Input	1	Num_features	
Hidden	3	64,32,32	ReLU
Output	1	Num classes	SoftMax

Both the local models were optimized using the Adam optimizer with the Categorical Cross-Entropy loss function (as given in equation 3.3). An Early Stopping method was adopted using the validation loss to increase generalization and reduce overfitting.
The basic parameters and training parameters used with federated Learning model is shown in table 4.

Table 4. The Configuration Parameters

Parameter	**Configuration**
Train/Test split	70-20-10
Optimizer	Adam
Activation Function	ReLU / SoftMax
Loss Function	Categorical Cross-Entropy
Early Stopping	Enabled (patience = 1, monitored validation loss)
Number of Clients	2-5
Number of Samples	96.016
Local Epochs	3
Batch Size	128
Communication Rounds	5
Aggregation Method	FedAvg
Random seeds	42
Learning rate	0.001
Runs	5

The proposed framework follows a Horizontal Federated Learning strategy, where all clients share an identical feature set while holding distinct subsets of samples. The training dataset was partitioned among five IoT clients using an Independent and Identically Distributed (IID) scheme. Consequently, all clients utilize the same feature representation and maintain approximately similar data distributions, without partitioning based on attack type or device identity. Although non-IID distribution is a recognized challenge in federated learning, this study investigates performance under an IID baseline setting.
Five IoT clients were selected for the federated learning framework. Each local model was trained on their individual data for three local epochs with a batch size of 128 on their own. In local training, each client stored its own network traffic locally and only sent the new parameters of the model to the central aggregation server, so that the data were kept private. Once all the local clients were trained, the model parameters were sent to the central server where they were aggregated together using Federated Averaging (FedAvg) algorithm.
The aggregated model was then sent to all clients, and all clients received the aggregated model, replacing to all the local models, before the next communication round begins

The collaborative training process was repeated five times, with the global model learning progressively learn knowledge while keeping to the principle of data privacy. This iterative approach allows the proposed IDS to use distributed training without the need of exchanging raw network traffic data. Besides, the reduced feature

subset based on the feature selection reduces the number of features, which significantly reduces the computational complexity, memory consumption, and communication overhead, making the proposed FL framework appropriate for resource-limited IoT environments.

5.2 Experimental results

This section presents the experimental result of the hybrid Feature Selection and proposed federated learning framework for privacy-preserving intrusion detection in IoT networks**.**

5.2.1 Results of the Feature Selection

To enhance the feature space and identify the most informative network features, the proposed hybrid feature selection method was employed by combining MI with Forward Selection. First, the MI evaluated the statistical dependency between each network feature and the target attack classes and ranked the features according to their MI scores. Subsequently, Forward Selection was applied using the MI-ranked feature list to iteratively select the most relevant features. At each iteration, the selected feature subset was evaluated using an MLP model to determine its contribution to the classification performance.

The proposed hybrid approach identified an optimal subset of five features: IN_BYTES, OUT_BYTES, FLOW_DURATION_MILLISECONDS, L7_PROTO, and TCP_FLAGS. In contrast, three features—IN_PKTS, OUT_PKTS, and PROTOCOL—were excluded because their inclusion did not provide further improvement in the model performance, thereby satisfying the predefined stopping criterion, as presented in Table 5.

Table 5. Selected features by hybrid feature selection

No.	Feature	Mi Score	F1 Score	Accuracy	Precision	Recall
1	IN_BYTES	0.426	0.6940	0.7863	0.6395	0.7863
2	OUT_BYTES	0.360	0.6940	0.7862	0.6425	0.7862
3	FLOW_D_MS	0.359	0.7542	0.8105	0.7707	0.8105
4	L7_PROTO	0.355	0.7971	0.8170	0.7436	0.8170
5	TCP_FLAGS	0.328	0.8121	0.8331	0.8096	0.8331

Following the feature selection process, the MLP model was retrained using the five selected features. The experimental results demonstrate that the reduced feature set maintained the model's predictive capability, achieving an F1-score of 81.21% and an accuracy of 83.31% compared with the model trained using the original feature set. Reducing the feature space from eight to five features demonstrates the effectiveness of the proposed approach in eliminating less informative and redundant features can improve the computational efficiency of the proposed IDS by reducing memory consumption, shortening training time, and decreasing inference overhead. Therefore, the proposed hybrid feature selection method is particularly suitable for developing lightweight IDS solutions for resource-constrained IoT environments.

5.2.2 Results of the Proposed Federated learning

This section summarizes the experimental assessment of the proposed federated learning approach to privacy-preserving ID in an IoT network. Once the features chosen are used to judge the effectiveness of the proposed privacy-preserving IDS, the training data is divided into multiple clients and each client trains a local MLP model using their private data. A global model was created by aggregating the local ones with FedAvg without sharing raw data. The results of the experiments indicate that the accuracy, precision, recall and F1 of the proposed federated learning model obtained 83.216 %, 82.072 %, 83.216 % and 81.784 % respectively. The federated approach resulted in comparable intrusion detection accuracy and F1-score to the centralized approach, with improved data privacy as all sensitive network traffic stayed on local clients during training.
The results obtained prove that the hybrid feature selection (as mention in section 4.1) and federated learning provide an effective trade-off between detection performance and privacy protection, respectively, which makes the proposed IDS more appropriate for the resource constrained IoT environments.

Table 6 compares the performance of the proposed FL framework with the CL approach using the NF-BoT-IoT dataset. Both approaches were evaluated over five independent runs under the same experimental settings, and the reported results are presented as the mean ± sample standard deviation (SD), reflecting both the average performance and variability across the repeated runs. The results show that the FL framework achieves comparable overall performance to the CL approach, with Accuracy of 83.216 ± 0.056% and F1-score of 81.784 ± 1.768%, compared with 83.166 ± 0.03% and 81.568 ± 0.77% for CL, respectively. Although CL achieves higher Precision (84.006 ± 2.33%), FL maintains competitive Precision (82.072 ± 3.509%) while providing the additional advantage of preserving data privacy. In the FL setting, raw network traffic is not shared with the central server; instead, only model updates are exchanged for aggregation. These results demonstrate that the proposed FL framework can achieve comparable IDS classification performance while improving privacy by avoiding the sharing of raw client data.

FL provides a structural privacy advantage by keeping raw data locally and sharing only model updates, so does not incorporate additional mechanisms such as differential privacy or secure aggregation.

Table 6: Comparison between CL and FL with NF-BoT-IoT dataset with Mean and Standard Deviation.

Learning Approach	privacy	Raw data sharing	Runs	Accuracy (Mean ±SD)	Precision (Mean ±SD)	Recall (Mean ±SD)	F1-score (Mean ±SD)
Centralized Learning (CL)	low	Yes	5	83.166 ± 0.03	84.006± 2.33	83.166 ± 0.03	81.568 ± 0.77
Federated Learning (FL)	high	No	5	83.216 ± 0.056	82.072 ± 3.509	83.216 ± 0.056	81.784 ± 1.768

Table 7 and Figures 6–7 illustrate the evolution of the global model during federated training. Accuracy increases from 0.830 in the first communication round to 0.832 in the second round and remains stable thereafter. Similarly, Recall stabilizes at 0.832 from the second round. Precision and F1-score, however, exhibit gradual changes across subsequent rounds, with Precision increasing to 0.815 and F1-score to 0.811 in the fifth round. The convergence plots therefore demonstrate that the global model reaches a stable overall performance within the five-round configuration.

Table 7. Impact of Communication Rounds on Global Model Performance and Classification Accuracy.

No. of round	Accuracy	Recall	Precision	F1-score
1	0.830	0.830	0.795	0.802
2	0.832	0.832	0.795	0.795
3	0.832	0.832	0.810	0.794
4	0.832	0.832	0.801	0.807
5	0.832	0.832	0.815	0.811

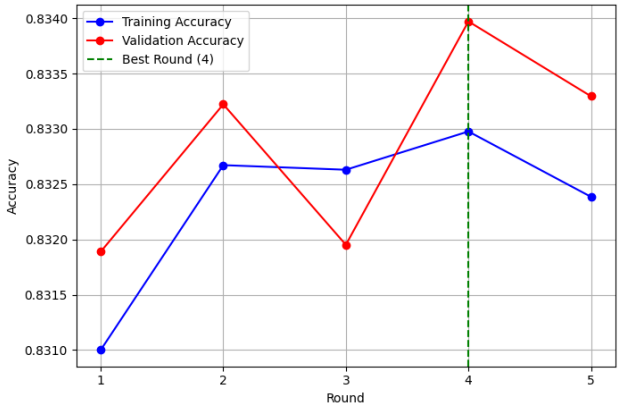

Figure 6. Accuracy of the global model across communication rounds

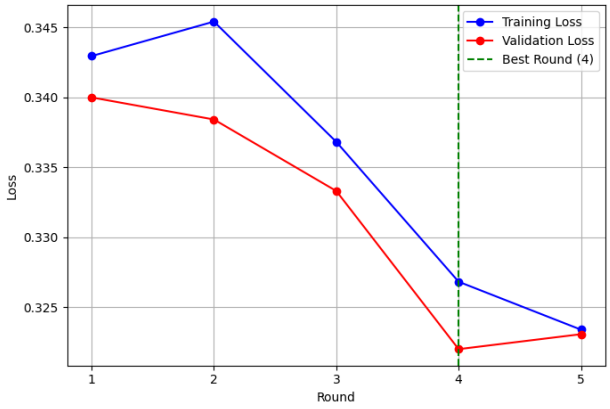

Figure 7. Loss of the global model across communication rounds

Table 8 presents the ablation study evaluating the effect of the proposed hybrid feature selection method on the IDS performance and computational efficiency. The results compare the model using the original eight features with the model using the five selected features. After applying feature selection, the number of input features was reduced from eight to five, Accuracy and F1-score remained comparable, more importantly, feature selection reduced the training time from 8.83 to 8.43 minutes, inference time from 7.23 to 6.13 seconds, and memory usage from 332.06 to 322.82 MB. These results demonstrate that the proposed feature selection method effectively reduces the computational and memory requirements while maintaining comparable detection performance, supporting its role in developing a lightweight IDS for resource-constrained IoT environments.

Table 8. Ablation comparing performance with and without the feature selection

Configuration	Feature	Accuracy	F1-score	Training Time	Inference Time	Memory Usage
Without Feature Selection	8	83.39	81.07	8.83 min	7.23 s	332.06 MB
With Feature Selection	5	83.21	81.78	8.43 min	6.13 s	322.82 MB

Table 9 and Table 10 present the per-class classification performance of the Centralized Learning (CL) and Federated Learning (FL) models across the five traffic classes: Benign, DDoS, DoS, Reconnaissance, and Theft. The tables report Precision, Recall, and F1-score for each class.

Table 9. Per-class performance of Federated Learning.

Class	Precision	Recall	F1 score
Benign	65.07	41.74	50.86
DDoS	38.71	93.89	54.82
DoS	44.40	4.08	7.48
Reconnaissance	97.75	93.06	95.35
Theft	100.00	0.52	1.04

Table 10. Per-class performance of Centralized Learning

Class	Precision	Recall	F1 score
Benign	65.15	41.34	50.58
DDoS	24.39	0.09	0.18
DoS	38.96	96.24	55.46
Reconnaissance	97.38	93.25	95.27
Theft	0.00	0.00	0.00

Table 11 summarizes the macro and weighted-average performance of the CL and FL models. Macro-averaged metrics give equal weight to each class, whereas weighted-average metrics account for the number of samples in each class.

Table 11. Macro- and Weighted-Average Performance Comparison of the Centralized and Federated Learning Models**.**

Average	CL	FL
Macro Precision	45.18	69.19
Macro Recall	46.18	46.66
Macro F1	40.30	41.91
Weighted Precision	83.88	86.36
Weighted Recall	83.21	83.23
Weighted F1	81.16	81.86

The confusion matrices provide a detailed visualization of the classification outcomes for the Centralized Learning (CL) and Federated Learning (FL) models across the five traffic classes. The rows represent the actual classes, while the columns represent the predicted classes. Correct classifications are indicated by the diagonal elements, whereas the off-diagonal elements represent misclassifications between classes.

Figure 8 presents the confusion matrix of the Centralized Learning model. It illustrates the distribution of correctly and incorrectly classified samples across the five classes and provides a detailed view of the model's class-specific prediction behavior.

Figure 9 presents the confusion matrix of the Federated Learning model. The matrix illustrates the correct predictions and misclassification patterns across the five classes, complementing the per-class Precision, Recall, and F1-score results reported in **Table 9 and Table 10**.

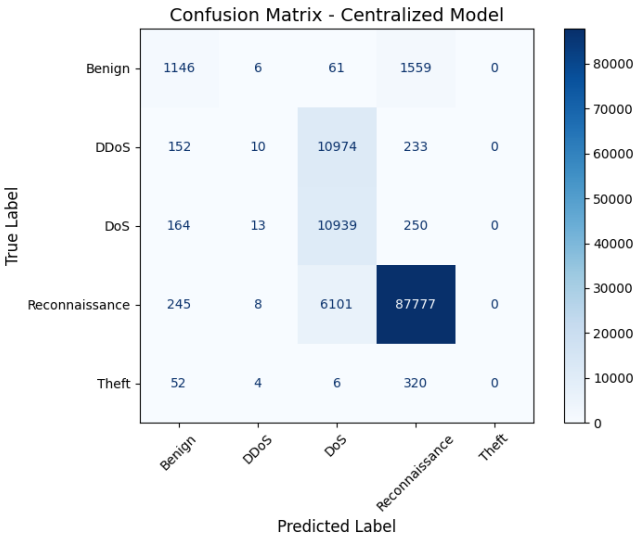

Figure 8. Confusion Matrix of the Centralized Learning Model

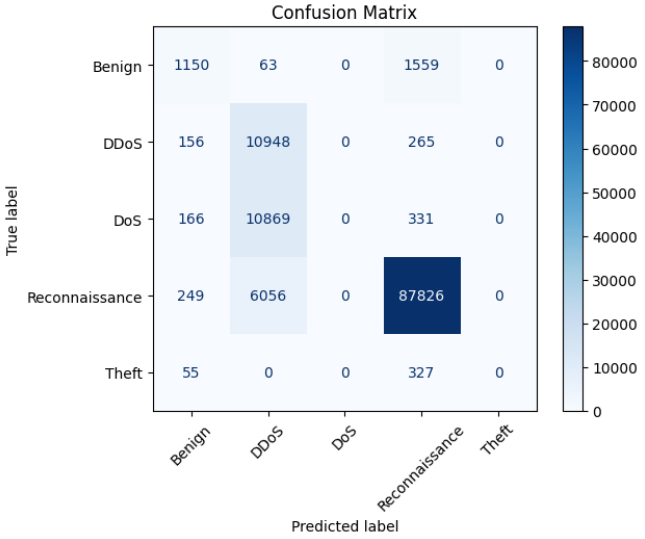

Figure 9. Confusion Matrix of the Federated Learning Model

5.2.2.1 Computational Efficiency Analysis

The proposed model uses five selected features and contains 3,685 trainable parameters, corresponding to a model size of 14.39 KB when using 32-bit floating-point parameters. For comparison, the same MLP architecture using the original eight features contains 3,877 trainable parameters, corresponding to 15.14 KB. Thus, the proposed feature selection reduces the number of input features by 37.5% and reduces the number of trainable parameters and model size by approximately 4.95%.

Furthermore, considering the final federated configuration of five clients and five communication rounds, and assuming that the complete model parameters are uploaded from each client to the aggregation server and the aggregated model is subsequently distributed back to all clients, the estimated total communication cost is 719.73 KB for the proposed five-feature model, compared with 757.23 KB for the eight-feature model. This represents an estimated 4.95% reduction in communication cost as shown in Table 12.

Table 12. Model Complexity and Estimated Communication Cost

Metric	8 Features	5 Features	Reduction
Number of Features	8	5	37.50%
Trainable Parameters	3,877	3,685	4.95%
Model Size	15.14 KB	14.39 KB	4.95%
Number of Clients	5	5	—
Communication Rounds	5	5	—
Total Communication Cost	757.23 KB	719.73 KB	4.95%

Table 13 Compares the proposed framework with some IDS based FL studies in a qualitative manner, and summarizes the data sets, learning models, aggregation techniques, and contributions.

Table 13. Comparison between the Proposed Framework and some IDS based FL studies

Study (year)	Dataset	Model	Aggregation	Main Contribution	Limitation
Campos et al. (2022)	CIC-ToN-IoT +	Logistic Regression	Fed+ / FedAvg	The impact of non-IID data in FL.	Resource constraints deployment challenges.
Zhao et al. (2020)	SEA	LSTM	FedAvg	FL-based IDS	Non-IoT dataset
Rahman et al. (2020)	NSL-KDD	FL + FedAvg	FedAvg	IDS based FL	Uses outdated NSL-KDD dataset
Khraisat et al. (2025)	N-BaIoT	PEIoT-DS	FedAvgM/ FedAvg .	Proposed a privacy-enhanced federated IDS IoT environments.	Communication overhead and local training cost.
Proposed method	NF-BoT-IoT	MLP	FedAvg	Lightweight privacy-preserving IDS using FL	Evaluated on one dataset

6. Conclusions

IoT devices encounter constraints such as memory capacity, processing capabilities, and battery longevity, which hinder the implementation of IDS models. Additionally, the growing number of sensitive network traffic produced by IoT devices poses privacy issues when centralized learning approaches are utilized. To address these challenges.

This study proposed a federated learning framework to develop a privacy-preserving intrusion detection system for IoT networks. The proposed framework allows the training of the model on distributed IoT clients without passing the network traffic to the main central server and only exchanging the model updates with the local clients using the FedAvg algorithm, which ensures privacy of data and competitive intrusion detection

performance. In the experiments, the proposed framework achieves an accuracy of 83.21% and an F1-score of 81.78 %.
FL provides a structural privacy advantage by keeping raw data locally and sharing only model updates so does not incorporate additional mechanisms such as differential privacy or secure aggregation.

The FL framework was able to offer similar detection performance as CL and also preserve data privacy, which makes it promising for the distributed IoT environment. The hybrid feature selection approach reduced the input features from eight to five, leading to a 15.21% reduction in inference time, a 4.59% reduction in training time, and a 2.78% reduction in memory usage. These findings demonstrate that the proposed framework provides a competitive, privacy-preserving, and computationally efficient IDS solution suitable for resource-constrained IoT environments.
The proposed future works, evaluate the proposed framework with different IoT datasets, and apply the proposed system in real IoT networks, training IDS with more advanced federated aggregation algorithms while still keeping privacy-preserving, and also consider different deep learning architectures to improve the performance of detection capabilities.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding Information

No funding was received from any financial organization to conduct this research

Author Contributions

All authors proposed the research problem.

Acknowledgments

The authors express their gratitude to Wasit University/ College of Engineering/Electrical Engineering department in Alkut-Wasit-Iraq for supporting this study.

References

[1] A. Heidari and M. A. Jabraeil Jamali, “Internet of Things intrusion detection systems: a comprehensive review and future directions,” *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, Dec. 2023, doi: 10.1007/s10586-022-03776-z.
[2] M. Aziz Al Kabir, W. Elmedany, and M. S. Sharif, “Securing IoT Devices Against Emerging Security Threats: Challenges and Mitigation Techniques,” 2023, *Taylor and Francis Ltd.* doi: 10.1080/23742917.2023.2228053.
[3] A. Zohourian, S. Dadkhah, H. Molyneaux, E. C. P. Neto, and A. A. Ghorbani, “IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks,” *Comput. Secur.*, vol. 146, Nov. 2024, doi: 10.1016/j.cose.2024.104034.
[4] M. M. Rahman, S. Al Shakil, and M. R. Mustakim, “A survey on intrusion detection system in IoT networks,” Dec. 01, 2025, *KeAi Communications Co.* doi: 10.1016/j.csa.2024.100082.
[5] M. Ali Khan, R. Naveed Bin Rais, O. Khalid, and F. Gul Khan, “A Comparative Analysis of Federated and Centralized Machine Learning for Intrusion Detection in IoT.”
[6] T. Bunko, M. N. Johnstone, W. Yang, and B. A. Scott, “A survey of privacy-preserving federated learning for intrusion detection systems,” *Artif. Intell. Rev.*, vol. 59, no. 5, May 2026, doi: 10.1007/s10462-026-11519-4.

[7] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," May 2019, [Online]. Available: http://arxiv.org/abs/1804.07474

[8] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Physical Communication*, vol. 42, Oct. 2020, doi: 10.1016/j.phycom.2020.101157.

[9] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things intrusion Detection: Centralized, On-Device, or Federated Learning?," *IEEE Netw.*, vol. 34, no. 6, pp. 310–317, Nov. 2020, doi: 10.1109/MNET.011.2000286.

[10] E. Novikova, E. Doynikova, and S. Golubev, "Federated Learning for Intrusion Detection in the Critical Infrastructures: Vertically Partitioned Data Use Case," *Algorithms*, vol. 15, no. 4, Apr. 2022, doi: 10.3390/a15040104.

[11] E. M. Campos *et al.*, "Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges," *Computer Networks*, vol. 203, Feb. 2022, doi: 10.1016/j.comnet.2021.108661.

[12] A. Khraisat, A. Alazab, M. Alazab, A. Obeidat, S. Singh, and T. Jan, "Federated learning for intrusion detection in IoT environments: a privacy-preserving strategy," *Discover Internet of Things*, vol. 5, no. 1, p. 72, Jun. 2025, doi: 10.1007/s43926-025-00169-7.

[13] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained IoT devices," *Mech. Syst. Signal Process.*, vol. 136, Feb. 2020, doi: 10.1016/j.ymssp.2019.106436.

[14] Z. ElSayed, A. Abdelgawad, and N. Elsayed, "Cybersecurity and Frequent Cyber Attacks on IoT Devices in Healthcare: Issues and Solutions," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.11250

[15] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of Things*, vol. 3, no. 1, Dec. 2023, doi: 10.1007/s43926-023-00034-5.

[16] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, Jan. 2024, doi: 10.3390/s24020713.

[17] N. Islam *et al.*, "Towards Machine Learning Based Intrusion Detection in IoT Networks," *Computers, Materials and Continua*, vol. 69, no. 2, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.

[18] Safana Hyder Abbas, Wedad Abdul Khuder Naser, and Amal Abbas Kadhim, "Subject review: Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)," *Global Journal of Engineering and Technology Advances*, vol. 14, no. 2, pp. 155–158, Feb. 2023, doi: 10.30574/gjeta.2023.14.2.0031.

[19] H. Liu *et al.*, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," *IEEE Trans. Veh. Technol.*, vol. 70, pp. 6073–6084, Apr. 2021, doi: 10.1109/TVT.2021.3076780.

[20] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," Nov. 01, 2021, *Springer*. doi: 10.1007/s42979-021-00815-1.

[21] Bassam *et al.*, "A Comparative Study of IDS-Based Deep Learning Models for IoT Network," *2023 International Conference on Advances in Artificial Intelligence and Applications (AAIA 2023), November 18â•fi20, 2023, Wuhan, China*, vol. 1, 2023, doi: 10.1145/3603273.

[22] S. Maza and M. Touahria, "Feature selection algorithms in intrusion detection system: A survey," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, pp. 5079–5099, 2018, doi: 10.3837/tiis.2018.10.024.

[23] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction," 2022, *Frontiers Media SA*. doi: 10.3389/fbinf.2022.927312.

[24] A. Omotosho, Y. Qendah, and C. Hammer, "IDS-MA: Intrusion Detection System for IoT MQTT Attacks Using Centralized and Federated Learning." [Online]. Available: https://www.kaggle.com/datasets/edotfs/dht11-temperature-and-humidity-

[25] M. Ali Khan, R. N. Bin Rais, O. Khalid, and M. Deriche, "Comparative Analysis of Centralized and Federated Intrusion Detection in IoT-Enabled Cyber-Physical Systems Under Data and Label-Skew," *IEEE Access*, vol. 13, pp. 160767–160785, 2025, doi: 10.1109/ACCESS.2025.3608856.

[26] G. Drainakis, K. V. Katsaros, P. Pantazopoulos, V. Sourlas, and A. Amditis, "Federated vs. Centralized Machine Learning under Privacy-elastic Users: A Comparative Analysis," in *2020 IEEE 19th*

International Symposium on Network Computing and Applications, NCA 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/NCA51143.2020.9306745.

[27] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," Apr. 2021, doi: 10.1109/COMST.2021.3075439.

[28] M. Venkatasubramanian, A. H. Lashkari, and S. Hakak, "IoT Malware Analysis Using Federated Learning: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 5004–5018, 2023, doi: 10.1109/ACCESS.2023.3235389.

[29] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513–535, Feb. 2023, doi: 10.1007/s13042-022-01647-y.

[30] S. Agrawal *et al.*, "Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions," Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.09527

[31] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems," Nov. 2020, doi: 10.1007/978-3-030-72802-1_9.