Exploring Federated Deep Learning for Internet of Things Cyberattacks

Abderahmane Hamdouchi
Vanguard Center, Mohammed VI Polytechnic University
Benguerir, Morocco

Ali Idri
Data and Software Sciences Research Laboratory
ENSIAS, Mohammed V University, Rabat, Morocco
Faculty of Medical Sciences, Mohammed VI Polytechnic University
Benguerir, Morocco

The Internet of Things (IoT) connects billions of devices that operate autonomously, increasing the risk of cyber threats, such as theft and manipulation of personal data. This has increased interest in utilizing deep learning (DL) methods to develop intrusion detection systems (IDS). In general, DL-based IDS rely on centralized approaches, which require IoT devices to transmit data to central servers for analysis. However, these centralized methods raise privacy concerns, prompting the adoption of federated learning (FL) as a promising alternative. This paper evaluates and compares various FL configurations using dense neural networks (DNNs) and convolutional neural networks (CNNs) as base models. The research explores three aggregation methods (FedAVG, FedPROX and FedSGD), three device counts (5, 15 and 30), two data setups (raw and balanced) and two feature selection methods (analysis of variance and chi-squared) with two feature thresholds (50% and 100%). The evaluation was conducted on the NF-ToN-IoT-v2 and NF-BoT-IoT-v2 datasets, using the Scott–Knott test and the Borda count method to analyze 144 FL configurations. The results indicate that FedAVG and FedPROX outperform other aggregation methods, with DNNs identified as the most effective base model for attack detection in FL environments. The top-performing models, using only 17 features, were DNN_R50_PROX_30 (accuracy of 97.80%) and CNN_R50_PROX_5 (accuracy of 99.87%) for NF-BoT-IoT-v2 and NF-ToN-IoT-v2, respectively.

Keywords: intrusion detection system; federated learning; deep learning; NetFlow; IoT; cybersecurity

1. Introduction

In today's digital era, the production and storage of data have skyrocketed, fueled by cheaper storage and the trend of recording every digital interaction. This growth is further amplified by the rise in

https://doi.org/10.25088/ComplexSystems.34.4.509

Internet of Things (IoT) devices, such as smart home gadgets, the development of smart cities and Industry 4.0 advancements. Big data companies highly value the information from these devices for insights and business intelligence, making data a crucial asset that must be protected [1].

As data has become increasingly central to daily operations and critical infrastructures, ensuring cybersecurity has become more vital than ever for preventing disruptions, unauthorized access and breaches. The field of information security continuously evolves to address emerging threats. Historically, cybersecurity efforts were primarily focused on a limited number of expert-operated systems. The proliferation of smartphones equipped with sensors and the generation of massive volumes of sensitive data were once unimaginable. Today, intrusion detection systems (IDS) play a crucial role in monitoring and detecting cyber threats at early stages. The adoption of machine learning (ML) has significantly enhanced the capabilities of IDS, transitioning from static systems reliant on databases of known threats to dynamic, self-adaptive methods. However, early ML-based systems struggled with adaptability and suffered from delayed updates, leaving them vulnerable for extended periods [2].

To address these challenges, modern IDS increasingly leverage ML models capable of autonomously learning and identifying novel threats. However, despite improved detection accuracy, most ML-based IDS solutions are built on centralized architectures, where a single entity aggregates and processes data from multiple devices. This centralization raises significant privacy concerns, particularly in IoT environments such as smart wearables and healthcare devices, where the data is highly sensitive and voluminous. Consequently, there is a growing need for decentralized approaches to data management and learning. Federated learning (FL) has emerged as a promising solution, enabling collaborative model training across distributed devices without exposing raw data, thus improving privacy preservation in sensitive IoT ecosystems [3].

FL was introduced in 2016 as a method where devices (also known as clients or parties) collaborate on learning without sharing their data. Instead, they send updates to a global model on a central entity (known as an aggregator or coordinator) for aggregation. FL aims to enhance user privacy by ensuring device data remains unshared with others [3].

Recently, there has been growing interest in creating FL-based IDS for IoT environments [4–6]. However, several proposed methods depended on unrealistic data distribution across parties and failed to evaluate various FL aggregation methods or the impact of using different numbers of devices [7]. Additionally, the review [8] pointed out the difficulties of implementing FL in the IoT but did not provide guidance on enhancing IDS with FL or critically assess their proposals. This lack of detailed analysis makes it challenging for cybersecurity experts to pinpoint the critical problems of integrating FL into IDS for the IoT.

In an earlier study [9], we evaluated FL for IDS in IoT environments, using dense neural networks (DNNs) as the base learner. The evaluation considered two data configurations: a raw data setup, which preserves the original class distribution across devices, and a balanced data setup, which ensures an equal number of attack and non-attack samples per device to mitigate class imbalance; and two aggregation methods: federated averaging (FedAVG) [10] and federated stochastic gradient descent (FedSGD) [11]. Experiments were conducted with groups of 5, 15 and 30 devices over 100 optimization rounds, utilizing the NF-ToN-IoT-v2 dataset. The previous study [9] identified the most effective configuration as the combination of raw data, the FedAvg method and a five-device setup. To validate or challenge these findings [9], we extended this work by: (1) incorporating convolutional neural networks (CNNs) as an additional deep learning (DL) base learner; (2) including federated proximal (FedPROX) as an additional FL aggregation method; (3) applying feature selection (FS) techniques such as analysis of variance (ANOVA) and chi-squared test (Chi2) with two feature thresholds: 50% (feature reduction) and 100% (no FS, serving as a baseline); and (4) integrating the NF-BoT-IoT-v2 dataset, which was generated in a different context than the NF-ToN-IoT-v2 dataset.

The selection of these FS filters is informed by their minimal computational resource demands and extensive use in the selection of subsets of features in diverse fields [12, 13]. The feature subsets were constructed using two feature thresholds: 50%, which reduces the number of features to half of the total set; and 100%, which retains the entire original feature set without applying any selection, thus serving as a baseline for performance comparison [13–15]. Additionally, two DL learning architectures, DNN and CNN, are employed due to their widespread adoption in intrusion detection [16–18]. FedAVG, FedSGD and FedPROX are selected as FL learning aggregation methods owing to their demonstrated efficiency and prominence within the FL domain [10, 11]. The chosen range of devices (5, 15 and 30) is aligned with established literature recommendations [19–21].

This study evaluates the performance of 144 FL configurations, derived from the combination of two datasets, three aggregation servers, two DL architectures, two data setups, two feature thresholds and three device counts, across 100 optimization rounds. This comprehensive evaluation framework represents a novel contribution, facilitating a robust comparison of model performance under diverse configurations to ensure reliable intrusion detection in varied IoT environments. Additionally, four widely adopted evaluation metrics are employed: accuracy, recall, precision and receiver operating characteristic area under the curve (AUC) [22]. The classifiers are analyzed using the Scott–Knott (SK) test to cluster them and identify the most stable cluster through statistical performance comparison. In addition, the Borda count (BC) ranking system is applied to determine the top-performing models based on multiple performance criteria.

The present paper aims to address the following research questions:

- RQ1: What are the best aggregation methods among FedAvg, FedPROX and FedSGD in the context of FL for attack detection?
- RQ2: What is the best configuration of FL for the detection of attacks across different settings?
- RQ3: Which DL architecture is the most suitable for FL as a base learner?

Here is an outline of the key contributions of this paper:

1. Proposing a framework for evaluating FL in the context of IDS.
2. Constructing 144 FL configurations with various data setups, feature thresholds, aggregation servers, device numbers and IoT datasets.
3. Determining the best aggregation methods for FL in the context of intrusion detection using the NF-ToN-IoT-v2 and NF-BoT-IoT-v2 datasets.
4. Determining the best DL base learner for FL in the context of intrusion detection.
5. Identifying the optimal FL setup across different configurations.

This paper is organized as follows: Section 2 examines relevant literature. Section 3 outlines the datasets, federated aggregators used, performance metrics, statistical tests and research methodology adopted in this paper. Section 4 presents a detailed analysis of the findings and compares our results with those of existing studies. Section 5 addresses the study's limitations and validity considerations. Finally, Section 6 concludes the paper and proposes avenues for future research.

2. Related Work

Numerous relevant studies have focused on anomaly detection in different fields, especially in the IoT, using different FL approaches. This section provides a summary of key research on using FL for intrusion detection in IoT contexts.

Mothukuri et al. [22] suggested a method that applied federated training sessions to gated recurrent units (GRUs) models. This method ensured data remained on local IoT devices, sharing only the model's learned weights with the central server of the FL. Additionally, it used an ensemble method to combine updates from various sources, enhancing the global ML model's accuracy. Their findings showed that this approach surpassed traditional centralized ML methods in protecting data privacy and achieved an overall average accuracy of 90.255% in detecting attacks. Campos et al. [23] introduced a study that examined an FL-powered IDS using a multiclass classifier to identify various attacks in an IoT environment. They used three distinct configurations (basic, balanced and mixed) derived by dividing the ToN_IoT dataset based on the IP addresses of IoT devices and attack types. The study also assessed two aggregation functions, Fed++ and FedAvg, utilizing the IBMFL framework. Their findings suggested that selecting instances based on the Shannon entropy of each local dataset can enhance overall accuracy and achieved the best results (close to 95.6%), comparable to those from a scenario where data is evenly distributed among all participants. Chen et al. [24] introduced FedAGRU, an FL method using an attention GRU, aimed at enhancing FedAVG algorithms. This model is crafted to detect poisoning attacks and remove updates with minimal contribution, leading to an efficient global model with reduced communication costs. Tested on three datasets (WSN-DS, KDD-CUP99 and CICIDS2017), FedAGRU demonstrated effective performance, achieving an accuracy of 99.82% on data that is not independent and identically distributed (non-IID).

Althunayyan et al. [25] proposed robust multi-stage IDS tailored for in-vehicle networks using a hybrid DL approach. Their system combines an artificial neural network (ANN) to detect known attacks and a long short-term memory (LSTM) autoencoder to identify novel threats using the car hacking dataset. By employing hierarchical FL (H-FL), the model enhances privacy by ensuring that sensitive in-vehicle data remains local while aggregating learned patterns at a central server. Experimental results demonstrated exceptional detection performance, with F1-scores exceeding 0.99 for known attacks and 0.95 for unseen ones, alongside a remarkably low false alarm rate of 0.016%. Bukhari et al. [26] introduced a novel intrusion detection model employing a hybrid architecture of stacked CNN (SCNN) and bidirectional LSTM (Bi-LSTM), leveraging FL for privacy preservation in wireless sensor networks (WSNs). This approach allowed distributed sensor nodes to collaboratively train a global model without sharing raw data, ensuring data privacy. The model utilized the WSN-DS and CIC-IDS-2017 datasets, achieving an accuracy of

99.9% across both datasets. Jin et al. [27] proposed an FL-IIDS to address the catastrophic forgetting issue in FL environments. The system employed dynamic example memory and innovative loss functions, such as class gradient balance loss and sample label smoothing loss, to improve local model performance for both old and new classes. Additionally, a relay client mechanism was introduced to select the best old model at a global level, further mitigating catastrophic forgetting. Using the UNSW-NB15 and CICIDS2018 datasets, the framework demonstrated enhanced classification accuracy and memory retention for older classes, achieving final accuracies of 68.76% and 99.62% on the respective datasets.

Table 1 sums up the findings, datasets used, classifiers investigated and the best performance values of some related studies dealing with the use of FL-IDS in the IoT context.

Paper	Dataset	FL Technique	Best Accuracy	Findings
Mothukuri et al. [22]	Modbus network dataset	FLAverage with GRU base learner and RF ensembler	90.25%	Introduced a method that used federated training sessions on GRU models with the Modbus network dataset, keeping data on local IoT devices. It also applied an ensemble technique to merge updates from diverse sources, improving the overall accuracy of the global ML model. Results reached an average accuracy of 90.255% in identifying attacks.
Campos et al. [23]	ToN_IoT	Two aggregation methods (Fed++ and FedAVG) using softmax regression as base learner	95.6%	Investigated an FL-IDS employing a multiclass classifier within an IoT setting, utilizing the ToN_IoT dataset across three data scenarios. It evaluated two aggregation functions, Fed++ and FedAVG, to determine their effectiveness in conjunction with softmax regression using the IBMFL framework. The findings indicated that choosing instances according to the Shannon entropy of each local dataset could improve overall accuracy, achieving results near 90%, which were comparable to other scenarios examined.

Paper	Dataset	FL Technique	Best Accuracy	Findings
Chen et al. [24]	WSN-DS, KDD-CUP99, CICIDS2017	FedAGRU based on (GRU-SVM, GRU-softmax, Improved CNN (ICNN) and VAE)	99.82%	Presented FedAGRU, an FL technique that incorporated a GRU to improve FedAVG algorithms. Designed to identify and filter out attacks and updates with low contribution, FedAGRU aimed to create an effective global model while lowering communication costs. It showed impressive results, achieving 99.82% accuracy on non-IID data.
Althunayyan et al. [25]	Car Hacking	H-FL using ANN and LSTM	0.95 (F1-score)	Proposed a multi-stage IDS combining an ANN for known attacks and an LSTM-autoencoder for unseen attacks, achieving a detection with F1-scores exceeding 0.99 for known attacks and 0.95 for unseen ones, alongside a remarkably low false alarm rate of 0.016%.
Bukhari et al. [26]	WSN-DS, CIC-IDS2017	FL with SCNN-Bi-LSTM	99.9%	Proposed a hybrid FL-based SCNN-Bi-LSTM model, achieving a notable accuracy of 99.9% in detecting intrusions while preserving data privacy and significantly reducing false positives and false negatives.
Jin et al. [27]	UNSW-NB15, CICIDS2018	FL-IIDS with dynamic memory and relay clients	99.62%	Proposed FL-IIDS to address catastrophic forgetting in FL using class gradient balance and label smoothing. Achieved 68.76% accuracy on UNSW-NB15 and 99.62% on CICIDS2018.

Table 1. Summary of the literature review.

3. Experimental Design

This section describes the datasets, performance metrics and methodology employed for the empirical evaluations conducted in the study.

3.1 Dataset Description

Figure 1 depicts the workflow for generating traffic flow data using the nProbe tool, developed by Ntop [28] and based on the NetFlow standard Version 9. This process involves extracting key attributes from network flow data stored in the pcap format and labeling the extracted records, which are then saved in CSV format. These attributes can be utilized for training or evaluating ML models. This study employs three intrusion detection datasets with a shared feature set:

- NF-ToN-IoT-v2: This dataset was created from the ToN-IoT dataset using the nProbe tool, as detailed by Sarhan et al. [29]. The ToN-IoT dataset, generated in an industrial network testbed, includes data from various virtual machines running Windows, Linux and Kali Linux operating systems. It captures both normal and cyber-attack events within IoT networks, covering attack types such as backdoor, DoS, DDoS, injection, MITM, password attacks, ransomware, scanning and XSS attacks. As illustrated in Figure 2(a), the NF-ToN-IoT-v2 dataset consists of 169 440 469 samples, with intrusion events accounting for 63.99% and normal events for 36.01%.

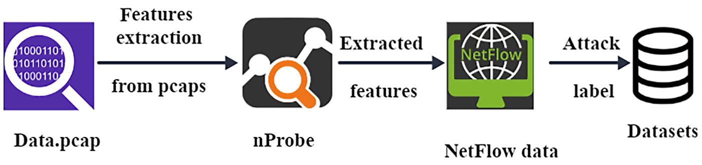


Figure 1. The feature extraction workflow.

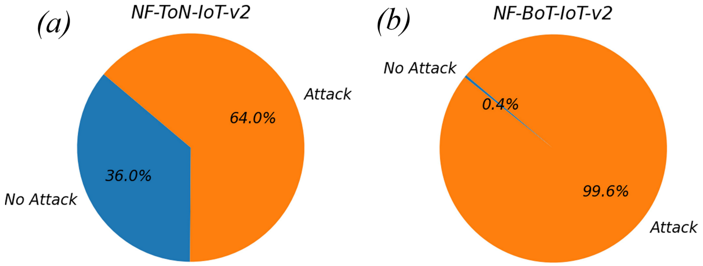


Figure 2. (a) Classes distribution of NF-ToN-IoT-v2; and (b) classes distribution of NF-Bot-IoT-v2.

- NF-BoT-IoT-v2: This dataset was derived from the BoT-IoT dataset [30] using the nProbe tool. The BoT-IoT dataset was created through a combination of network platforms, simulated IoT services and feature extraction integrated with forensic analytics. It was developed at the UNSW Canberra Research Cyber Range Lab using an ESXi-configured cluster of virtual machines managed via the vSphere platform. This setup, connected to both LAN and WAN, enabled IoT service simulation using Node-RED and AWS IoT Hub, with MQTT protocol facilitating machine-to-machine communication. Attack scenarios include reconnaissance, DoS, DDoS and data theft. Figure 2(b) shows that the NF-BoT-IoT-v2 dataset contains 37 763 497 samples, with 99.64% representing intrusion events and 0.36% normal events.

3.2 Federated Learning Aggregation Methods

This section explains the FL aggregation methods used in this paper, specifically FedProx, FedAVG and FedSGD. These optimization algorithms are designed to train ML models across distributed devices while ensuring data privacy is maintained.

- Federated Averaging (FedAvg) is a distributed optimization algorithm designed for FL. It operates by selecting a subset of devices in each communication round, performing local stochastic gradient descent (SGD) for a fixed number of epochs on each device and averaging the resulting model updates on a central server. While effective in reducing communication costs, FedAvg assumes uniform computational capabilities across devices and does not account for statistical heterogeneity, which can lead to divergence or unstable convergence in practical, non-IID settings. Its simplicity and empirical success have made it a baseline method in FL, though it lacks theoretical guarantees for heterogeneous environments [31].
- Federated optimization framework (FedProx) is designed to address statistical heterogeneity in distributed networks. It extends FedAvg by introducing a proximal term to the local subproblems, which restricts the updates to remain close to the global model. This modification enhances convergence robustness and stability, particularly in non-IID data settings. FedProx provides theoretical guarantees under a bounded dissimilarity assumption and demonstrates improved empirical performance across diverse datasets compared to FedAvg. The framework is flexible, allowing any local solver, and maintains the privacy and efficiency benefits of FL [32].
- Federated Stochastic Gradient Descent (FedSGD) is a distributed optimization method used in FL to train models across decentralized devices while preserving data privacy. Unlike standard SGD, it computes gradients locally on each client and aggregates them on a central server to update the global model. Each client performs a single gradient step using local data before transmitting the update. FedSGD promotes consistency across local models by penalizing weight divergence through a regularization term, such as total variation minimization. While communication-efficient, this method requires careful tuning of hyperparameters, including learning rate and regularization strength, to address statistical heterogeneity [33].

3.3 Performance Measures

Four criteria were used to evaluate the federated DL variants of this study: accuracy, recall, precision and receiver operating characteristic

AUC [34]. They are defined by equations (1) to (4), respectively:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{AUC} = \int_0^1 \text{TP}\big(\text{FP}^{-1}(x)\big)dx \approx \sum_{i=1}^{n-1} \frac{\text{TP}_i + \text{TP}_{i+1}}{2} \times (\text{FP}_{i+1} - \text{FP}_i) \tag{4}$$

where TP are true positives, FP are false positives, TN are true negatives, and FN are false negatives.

3.4 Statistical Tests and the Borda Count Ranking Method

This section provides an explanation of the SK test and the BC ranking system. The SK test groups classifiers through statistical comparisons, while the BC ranking system is a voting-based method used to rank models within the best SK test cluster by evaluating multiple performance criteria.

- Scott–Knott (SK) is a clustering algorithm frequently used for comparing multiple groups in the ANOVA studies. It avoids the issue of overlapping groups. Effectively, the SK method begins with all observed mean effects grouped together. It then continuously divides these groups into smaller subgroups, ensuring that no two subgroups share any common members [35].
- Borda count (BC) is a voting method to rank candidates by preference. Each candidate gets points based on their rank, with lower ranks getting fewer points. The points are then added up, and the candidate with the highest total wins. In this paper, we used the BC ranking system to identify the top-performing model, treating all performance measures equally [36].

3.5 Methodology

Figure 3 illustrates the methodology used to evaluate and compare the impact of different aggregation methods and the number of devices on the detection performance of FL-based IDS. The study assessed the performance of three aggregation methods (FedAVG, FedSGD and FedProx) using two NetFlow IoT datasets (NF-ToN-IoT-v2 and NF-BoT-IoT-v2) under balanced and raw data setups with two feature thresholds (50% and 100%). The experiments were conducted with 5, 15 and 30 devices over 100 training rounds. Performance evaluation was conducted using the SK test and the BC voting system.

The experimental procedure comprised the following steps:

- Step 1: The raw data was preprocessed to ensure quality and consistency across both datasets. Initial steps included the removal of missing values, duplicate records and non-informative attributes. Numerical features were examined for multicollinearity, and those with a Pearson correlation above 0.95 and a variance inflation factor (VIF) below 5 were selectively removed. Categorical features were optimized: (1) Features with high cardinality were consolidated to reduce dimensionality. For instance, IP address fields were grouped into five categories: three private address ranges, public addresses and localhost entries. Port numbers were categorized as well-known, registered or dynamic ports; (2) Other categorical features were retained with minimal classes, such as PROTOCOL and DNS_QUERY_TYPE, which contain six and 12 distinct values, respectively. The L7_PROTO attribute was simplified by retaining the five most frequent values and grouping the rest under an “Others” category, which accounted for approximately 3.4% of all entries.

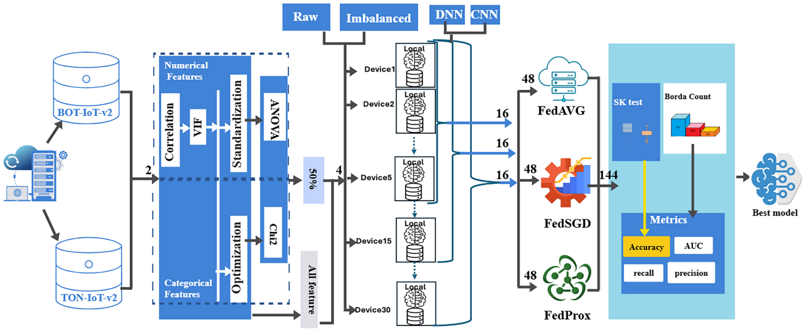


Figure 3. Experimental process.

In addition, standardization was applied to scale numerical features to have zero mean and unit variance, using

$$Z = \frac{x_i - \mu}{\sigma} \tag{5}$$

where x_i is the data point, μ is the mean and σ is the standard deviation.

- Step 2: Set up a simulated IoT network using virtual instances with TensorFlow Federated (TFF). Three sets of end devices (5, 15 and 30) were created, with each device labeled as Device$_i$. Three central server instances were configured to implement FedAVG, FedSGD and FedProx, and two DL base learners: DNN, as shown in Table 2; and CNN, as shown in Table 3, were built. The central servers facilitated the exchange of DL model parameters between the mobile IoT devices and the central FL server. The two datasets, NF-ToN-IoT-v2 and NF-BoT-IoT-v2, were divided into two sections based on different data distribution strategies: (1) The raw data setup preserved the original class imbalance of the dataset by distributing data across devices in accordance with the natural proportion of attack and non-attack samples; and (2) The balanced data setup ensured an equal number of attack and non-attack samples per device, thereby mitigating the effects of class imbalance and facilitating fairer model training across all clients. The hyperparameters used across the different configurations are summarized in Table 4.

Layer Type	Details
Dense Layer 1	Units: 128, Activation: ReLU, Input Shape: (len(Features),)
Dense Layer 2	Units: 64, Activation: ReLU
Dense Layer 3	Units: 1, Activation: Sigmoid (for binary classification)

Table 2. DNN architecture.

Layer Type	Details
Conv1D Layer 1	Filters: 32, Kernel size: 3, Activation: ReLU
MaxPooling1D Layer 1	Pool size: 2
Conv1D Layer 2	Filters: 64, Kernel size: 3, Activation: ReLU
MaxPooling1D Layer 2	Pool size: 2
Flatten Layer	Converts 1D feature maps into a vector.
Dense Layer 1	Units: 128, Activation: ReLU
Dense Layer 2	Units: 1, Activation: Sigmoid (for binary classification)

Table 3. CNN architecture.

Parameter	Description and Group
Evaluation Optimizer	Adam optimizer used for model evaluation across all configurations
Number of Rounds	100 rounds of federated training used in all setups
Training Optimizer	• FedProx: SGD (lr=0.1, clipvalue=1.0) • FedSGD: Implicit optimizer, no learning rate specified • FedAvg: SGD (lr=0.1)
Learning Rate	0.1 (FedProx and FedAvg); unspecified in FedSGD
Gradient Clipping	Applied only in FedProx (clipvalue=1.0)
Proximal Term Coefficient	0.001 (only for FedProx setup)
Model Loss Function	BinaryCrossentropy used in all configurations

Table 4. FL and DL hyperparameters.

This non-IID configuration reflects realistic FL deployment scenarios in heterogeneous IoT environments, where devices naturally observe locally distinct data distributions. In our paper, the local datasets assigned to each client were generated using a split-and-shuffle strategy that maintained non-identical class proportions across devices, thereby ensuring a non-IID data distribution throughout all experiments (raw and balanced).

Each configuration (raw and balanced) was further analyzed under two feature thresholds: 50%, where only half of the most relevant features were selected; and 100%, where all original features were retained without applying any FS, serving as a baseline for performance comparison. Each local dataset i was assigned to its respective virtual Device$_i$.

- Step 3: Construct and evaluate the performance of 144 FL configurations (2 datasets $*$ 2 data setups $*$ 2 feature thresholds $*$ 2 DL base learners $*$ 3 number of devices $*$ 3 federated aggregation methods). The evaluation metrics included accuracy, recall, precision and AUC, assessed over 100 training rounds. Additionally, the SK test and BC system were employed to rank the FL configurations for each device count and aggregation method.
- Step 4: Compare the performance of FedAVG, FedProx and FedSGD, as well as evaluate the performance of DNN and CNN for each dataset, data setup, feature threshold and device count. Finally, the optimal FL configuration for cyber-detection within the NetFlow IoT dataset framework was identified.

3.6 Abbreviation

To make it easier for readers and simplify model names, this paper adopts specific naming conventions for models as follows:

DLBaseLearner_DataSetup&FeatureThreshold_AggregationMethod_NumberOfDevices

The abbreviations for data setup techniques are as follows: B represents balanced data, and R represents raw data. The aggregation techniques are abbreviated as AVG for FedAVG, PROX for FedProx and SGD for FedSGD. For instance, the configuration DNN_R50_SGD_5 denotes the use of DNN as a base learner, raw data as the data setup, a 50% feature threshold and FedSGD as the aggregation method, applied to five devices.

4. Results and Discussions

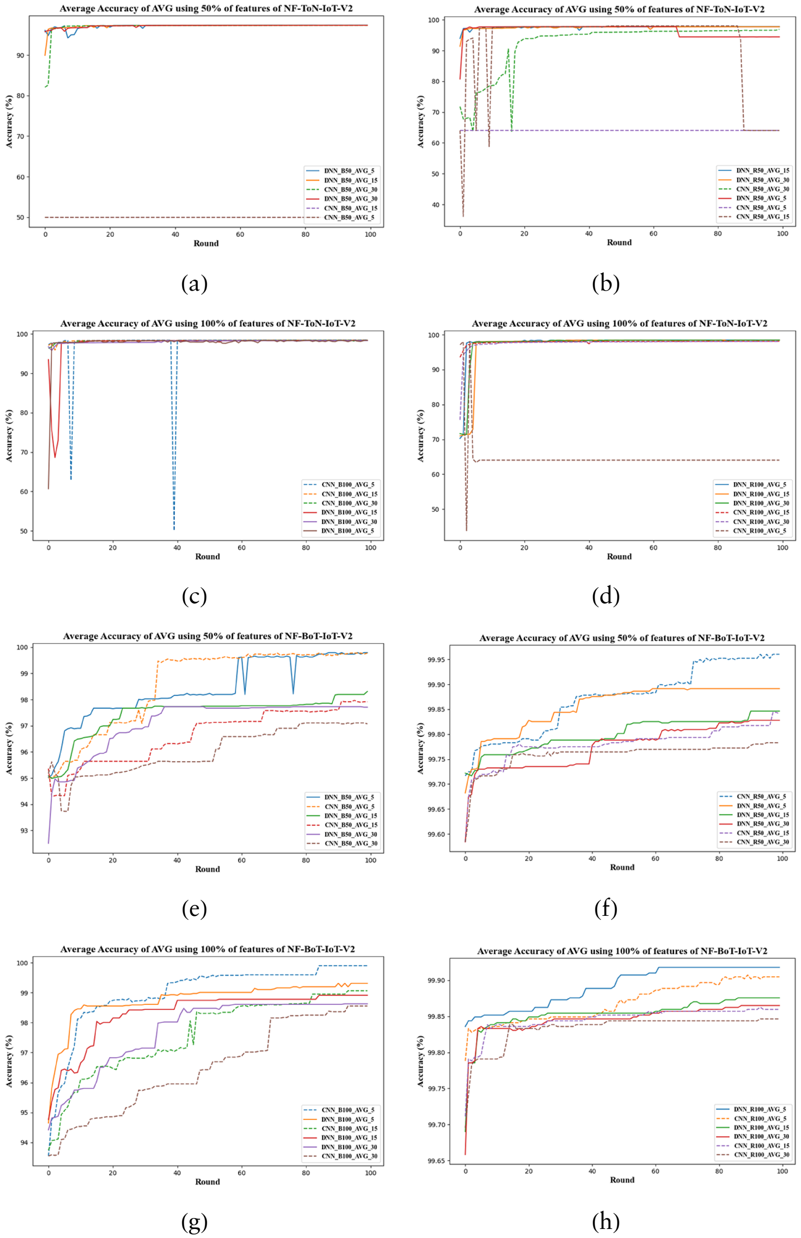
This section examines the results of applying the FL technique with DNN and CNN architectures, evaluating three aggregation methods (FedAVG, FedPROX and FedSGD), two data setups (raw and balanced), two feature thresholds (50% and 100%) and three device configurations (5, 15 and 30). The analysis is conducted over 100 rounds using the NF-ToN-IoT-v2 and NF-BoT-IoT-v2 datasets for binary classification. The presentation and the discussion of the results are structured to address the RQs of Section 1.

4.1 Evaluating and Comparing Aggregation Methods of Federated Deep Learning for Attack Detection (RQ1)

This subsection examines the impact of the three aggregation methods (FedAVG, FedPROX and FedSGD) on the performance of different FL configurations, focusing on identifying the aggregation method that enhances the accuracy of FL-based IDS in IoT contexts. The analysis evaluates the average accuracy of models across different device counts for each dataset, feature threshold and data setup over 100 rounds, as illustrated in Figures 4, 5 and 6. For instance, average accuracy values across device numbers are computed for each round. In Figure 4(a), a balanced data setup using DNN and FedAVG for 5 and 15 devices is represented as DNN_B50_AVG_5 and DNN_B50_AVG_15, respectively. Similarly, in Figure 4(b), a raw data setup using CNN and FedAVG for 5 and 15 devices is denoted as CNN_R50_AVG_5 and CNN_R50_AVG_15, respectively.

Figure 4 illustrates the average accuracy values obtained using FedAVG for NF-TON-IoT-v2 and NF-BoT-IoT-v2 across different feature thresholds, feature setups, DL base learners and device numbers. We observe that:

- From Figure 4(a), which presents the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 10, after which their performance stabilized, achieving an accuracy of 96%. However, the CNN_B50_AVG_5 model deviated from this trend, stabilizing at an accuracy of 50%.
- From Figure 4(b), which presents the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models exhibited progressive improvement up to round 20, after which their accuracy stabilized at 96%. However, two models, CNN_R50_AVG_5 and CNN_R50_AVG_15, deviated from this trend. The CNN_R50_AVG_5 model stabilized at an accuracy of 64%, while the CNN_R50_AVG_15 model experienced a significant drop in accuracy at round 85, ultimately stabilizing at 64%.
- From Figure 4(c), which presents the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a balanced data configuration, all FL models showed progressive improvement up to round 12, after which their performance stabilized, achieving an accuracy of 98%. However, the CNN_B100_AVG_5 model exhibited notable drops in accuracy, with a decrease to approximately 63% in round 9 and another decline to 50% in round 39.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

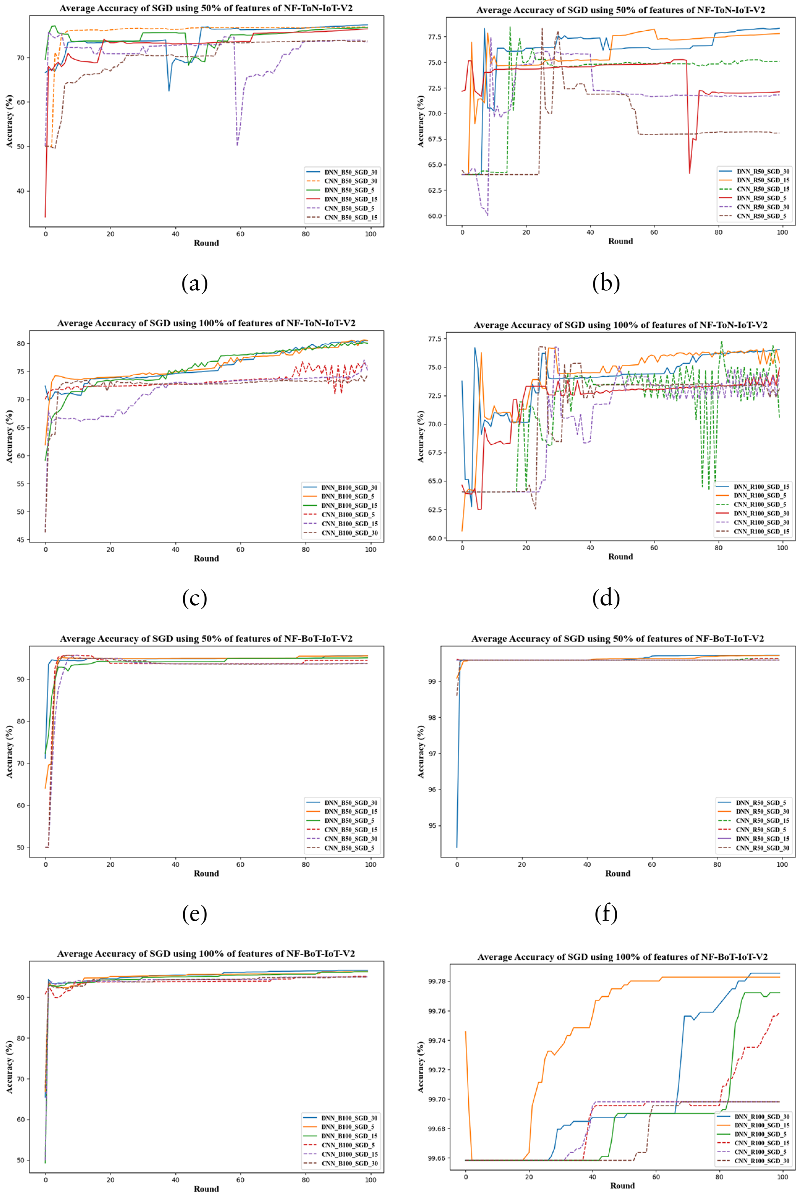
Figure 4. Average accuracy of FedAVG for NF-ToN-IoT-v2 with: (a) balanced data setup and 50% of features; (b) raw data setup and 50% of features; (c) balanced data setup and 100% of features; (d) raw data setup and 100% of features; and for NF-BoT-IoT-v2 with: (e) balanced data setup and 50% of features; (f) raw data setup and 50% of features; (g) balanced data setup and 100% of features; and (h) raw data setup and 100% of features.

- From Figure 4(d), which presents the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a raw data configuration, the FL models showed progressive improvement up to round 5, after which their performance stabilized, achieving an accuracy of 98%. However, the CNN_R100_AVG_5 model deviated from this trend, stabilizing at an accuracy of 64%.
- From Figure 4(e), which presents the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 10 rounds. Accuracy ranged from approximately 97% for the CNN_B50_AVG_30 model to 100% for the DNN_B50_AVG_5 model.
- From Figure 4(f), which presents the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 10 rounds. Accuracy ranged from approximately 99.75% for the CNN_R50_AVG_30 model to 99.95% for the CNN_R50_AVG_5 model.
- From Figure 4(g), which presents the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a balanced data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 10 rounds. Accuracy ranged from approximately 98% for the CNN_B100_AVG_30 model to 100% for the CNN_B100_AVG_5 model.
- From Figure 4(h), which presents the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a raw data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 20 rounds. Accuracy ranged from approximately 99.83% for the CNN_R100_AVG_30 model to 99.92% for the DNN_R100_AVG_5 model.

Under FedAVG, final accuracy for NF-ToN-IoT-V2 ranged from approximately 50% to 98%, typically stabilizing within five to 20 rounds. For NF-BoT-IoT-V2, accuracy spanned about 97% to 100%, reaching stability in five to 20 rounds. Some configurations achieved perfect accuracy, while others dropped to 50% or 64%.

Figure 5 illustrates the average accuracy values obtained using FedSGD for NF-TON-IoT-V2 and NF-BoT-IoT-V2 across different feature thresholds, feature setups, DL base learners and device numbers. We observe that:

- From Figure 5(a), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 60, after which their performance stabilized. Accuracy ranged from approximately 72% for the CNN_B50_ SGD_15 model to 76% for the DNN_B50_SGD_30 model. However, the CNN_B50_ SGD_15 model exhibited notable drops in accuracy, with a decrease to approximately 50% in round 60.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

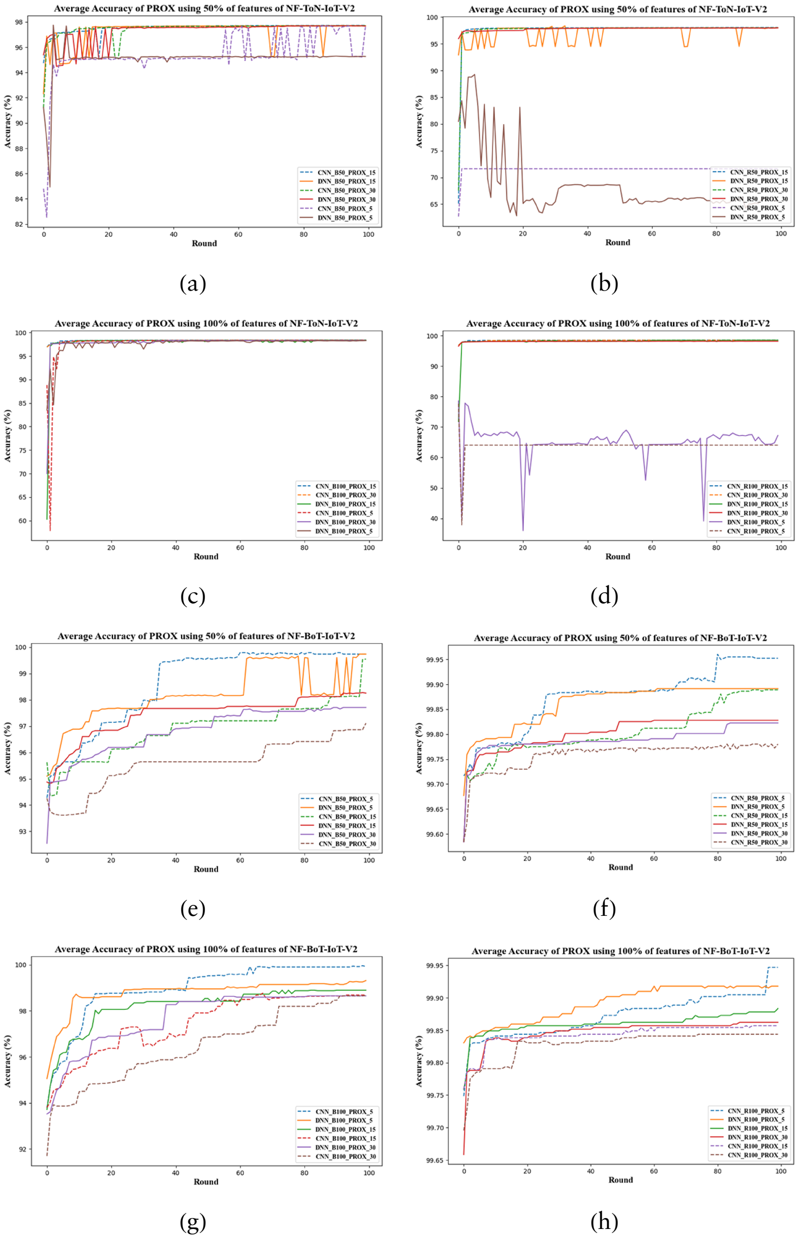
Figure 5. Average accuracy of FedSGD for NF-ToN-IoT-v2 with: (a) balanced data setup and 50% of features; (b) raw data setup and 50% of features; (c) balanced data setup and 100% of features; (d) raw data setup and 100% of features; and for NF-BoT-IoT-v2 with: (e) balanced data setup and 50% of features; (f) raw data setup and 50% of features; (g) balanced data setup and 100% of features; and (h) raw data setup and 100% of features.

- From Figure 5(b), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models showed progressive improvement up to round 40, then a decrease up to around 60 revolutions, after which their performance stabilized. Accuracy ranged from approximately 67.5% for the CNN_R50_SGD_5 model to 77% for the DNN_R50_SGD_30 model.
- From Figure 5(c), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 50, after which their performance stabilized. Accuracy ranged from approximately 70% for the CNN_B100_SGD_30 model to 80% for the DNN_B100_SGD_30 model. However, the CNN_B100_SGD_5 model exhibited significant oscillations between rounds 70 and 100.
- From Figure 5(d), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a raw data configuration, all FL models exhibited oscillations within 100 rounds, except for the DNN_R100_SGD_15 model, which stabilized within 70 rounds, achieving a maximum accuracy of 76% and a minimum accuracy of 71%, which was achieved by CNN_R100_SGD_15.
- From Figure 5(e), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 5, after which their performance stabilized. Accuracy ranged from approximately 93% for the CNN_B50_SGD_5 model to 96% for the DNN_B50_SGD_30 model.
- From Figure 5(f), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models showed progressive improvement up to round 2, after which their performance stabilized. Accuracy ranged from approximately 99.5% for the CNN_R50_ SGD_30 model to 99.8% for the DNN_R50_SGD_5 model.
- From Figure 5(g), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 10, after which their performance stabilized. Accuracy ranged from approximately 93% for the CNN_B100_ SGD_30 model to 95% for the DNN_B100_SGD_30 model.
- From Figure 5(h), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a raw data configuration, the FL models showed progressive improvement up to round 80, after which their performance stabilized. Accuracy ranged from approximately 99.7% for the CNN_R50_ SGD_30 model to 99.8% for the DNN_R100_SGD_30 model.

FedSGD models exhibit consistent improvement followed by performance stabilization under varying feature thresholds, data configurations and device numbers. For NF-ToN-IoT-V2, balanced data configurations converge to accuracies between 70% and 80% at rounds 50 to 60, with occasional drops or oscillations. Raw data setups show a more volatile pattern but stabilize around 40 to 70 rounds, attaining 67.5% to 77% accuracy. In contrast, NF-BoT-IoT-V2 reaches higher final accuracies—93% to 99.8%— and typically stabilizes earlier.

Figure 6 illustrates the average accuracy values obtained using FedPROX for NF-TON-IoT-V2 and NF-BoT-IoT-V2 across different feature thresholds, feature setups, DL base learners and device numbers. We observe that:

- From Figure 6(a), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 20, after which their performance stabilized. Accuracy ranged from approximately 95% for the DNN_B50_ PROX_15 model to 97% for the DNN_B50_PROX_5 model. However, the CNN_B50_PROX_5 and DNN_ B50_PROX_15 models exhibited significant oscillations between rounds 50 and 100.
- From Figure 6(b), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models showed progressive improvement up to round 3, after which their performance stabilized, achieving an accuracy of 97%. However, the CNN_R50_PROX_5 and DNN_R50_PROX_5 models deviated from this trend. The CNN_R50_PROX_5 model stabilized at an accuracy of 72%, while the DNN_R50_PROX_5 model exhibited oscillations, ultimately achieving an accuracy of 65%.
- From Figure 6(c), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a balanced data configuration, the FL models showed progressive improvement up to round 5, after which their performance stabilized, achieving an accuracy of 97%.
- From Figure 6(d), presenting the FL models generated using the NF-ToN-IoT-V2 dataset with 100% of the features and a raw data configuration, the FL models showed progressive improvement up to round 2, after which their performance stabilized, achieving an accuracy of 97%. However, the DNN_R100_PROX_5 and CNN_R100_PROX_5 models deviated from this trend. The CNN_R100_PROX_5 model stabilized at an accuracy of 62%, while the DNN_R100_PROX_5 model exhibited oscillations, ultimately achieving an accuracy of 63%.
- From Figure 6(e), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a balanced data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 10 rounds. Accuracy ranged from approximately 96% for the CNN_B50_PROX_30 model to 100% for the CNN_B50_PROX_5 model. However, the DNN_B50_PROX_5 model exhibited significant oscillations between rounds 75 and 100.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6. Average accuracy of FedPROX for NF-ToN-IoT-v2 with: (a) balanced data setup and 50% of features; (b) raw data setup and 50% of features; (c) balanced data setup and 100% of features; (d) raw data setup and 100% of features; and for NF-BoT-IoT-v2 with: (e) balanced data setup and 50% of features; (f) raw data setup and 50% of features; (g) balanced data setup and 100% of features; and (h) raw data setup and 100% of features.

- From Figure 6(f), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 50% of the features and a raw data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 20 rounds. Accuracy ranged from approximately 99.75% for the CNN_R50_PROX_30 model to 99.96% for the CNN_R50_PROX_5 model.
- From Figure 6(g), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a balanced data configuration, the FL models exhibited steady improvement over the rounds, stabilizing during the final 30 rounds. Accuracy ranged from approximately 98% for the CNN_B100_PROX_30 model to 100% for the CNN_B100_PROX_5 model.
- From Figure 6(h), presenting the FL models generated using the NF-BoT-IoT-V2 dataset with 100% of the features and a raw data configuration, the FL models showed progressive improvement up to round 50, after which their performance stabilized. Accuracy ranged from approximately 99.85% for the CNN_R100_PROX_30 model to 99.95% for the CNN_R100_PROX_5 model.

Under FedPROX, FL models generally converge to high accuracy for both NF-ToN-IoT-V2 and NF-BoT-IoT-V2, though some configurations exhibit performance drops or oscillations. For NF-ToN-IoT-V2, balanced data setups typically stabilize around 95% to 97% accuracy, while certain raw configurations fall to 62% to 72%. In contrast, NF-BoT-IoT-V2 models frequently exceed 96%, with multiple cases reaching 100%.

Overall, FedAVG and FedPROX tend to achieve higher final accuracies and faster stabilization than FedSGD, especially on NF-BoT-IoT-V2, where multiple configurations consistently reach or approach 100%. However, FedAVG occasionally experiences sharp accuracy drops on NF-ToN-IoT-V2, while FedPROX shows more consistent but still imperfect stability in a few raw-data cases. FedSGD generally converges to lower or more volatile accuracy on NF-ToN-IoT-V2, although it remains competitive on NF-BoT-IoT-V2. Hence, for robust performance across both datasets, FedPROX offers a slight edge overall, with FedAVG performing comparably or better in certain NF-BoT-IoT-V2 scenarios.

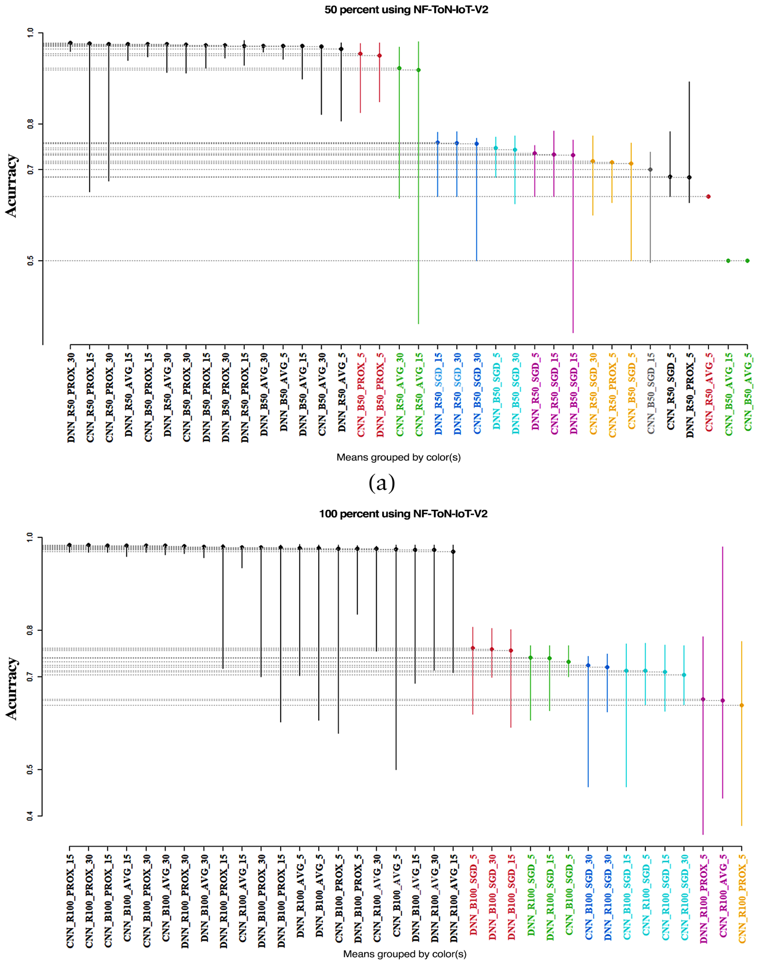
4.2 Evaluating Optimal Federated Learning Configurations for Attack Detection across Devices (RQ2)

In this subsection, we compare different FL configurations by mixing aggregation methods, data setups, DL base learners and numbers of devices for each dataset and feature threshold. We used the SK test, focusing on accuracy, to group models and pinpoint the most effective SK clusters, as illustrated in Figures 7 and 8. Additionally, we use the BC method to prioritize models within the top SK clusters, based on metrics such as accuracy, AUC, recall and precision, as shown in Tables 5 and 6. The SK test results are displayed on a graph where the x axis categorizes FL classifier variants by cluster, arranging the best clusters from left to right, and the y axis shows accuracy scores. Each vertical line's central dot represents the mean accuracy, while the line itself illustrates the accuracy outcomes over 100 rounds for a given FL classifier. The analysis involves calculating the average accuracy of each round i across devices, denoted as Average_i, using equation (6). For example, Figure 7(a) presents the SK test results of 36 FL configurations using 50% of the features of the NF-ToN-IoT-v2 dataset. In this context, DNN_R50_AVG_5 represents the accuracy averages of DNN_R50_AVG (a DNN model with a raw data setup, 50% of features and FedAVG) applied to five devices over 100 rounds ($\text{Average}_1, \ldots, \text{Average}_{100}$):

$$\text{Average}_i = \frac{\sum_{j=1}^{\#\text{Devices}} \text{Accuracy}_j}{\#\text{Devices}}. \tag{6}$$

For the NF-ToN-IoT-v2 dataset:

- Figure 7(a) presents the SK results for 50% of the features, identifying 11 clusters: (1) The first cluster contains 15 FL models, including 10 DNN models and 5 CNN models. Among the DNN models, five use raw data (DNN_R50_PROX_30, DNN_R50_PROX_15, DNN_R50_AVG_30, DNN_R50_AVG_15 and DNN_R50_AVG_5), and five use a balanced data setup with the same configurations as the raw data models. For the CNN models, two use raw data with FedPROX (CNN_R50_PROX_15 and CNN_R50_PROX_30), two use a balanced data setup with the same configurations as the raw data models, and one additional model is CNN_B50_AVG_30; (2) the second cluster includes two models, CNN_B50_PROX_5 and DNN_B50_PROX_5; (3) the third cluster includes two CNN models using FedAVG and raw data (CNN_R50_AVG_30 and CNN_R50_AVG_15); (4) the fourth cluster contains two DNN models (DNN_R50_SGD_30 and DNN_R50_SGD_15) and one CNN model (CNN_B50_SGD_30); (5) the fifth cluster includes two DNN models using balanced data and FedSGD (DNN_B50_SGD_5 and DNN_B50_SGD_30); (6) the sixth cluster includes two DNN models (DNN_R50_SGD_5 and DNN_B50_SGD_15) and one CNN model (CNN_R50_SGD_15); (7) the seventh cluster contains three CNN models (CNN_R50_SGD_30, CNN_R50_PROX_5 and CNN_B50_SGD_5); (8) the eighth cluster contains a single model, CNN_B50_SGD_15; (9) the ninth cluster contains two models, CNN_R50_SGD_5 and DNN_R50_PROX_5; (10) the tenth cluster contains a single model, CNN_R50_AVG_5; and (11) the final cluster, includes two CNN models using SGD with device counts of 5 and 15.



(a)

(b)

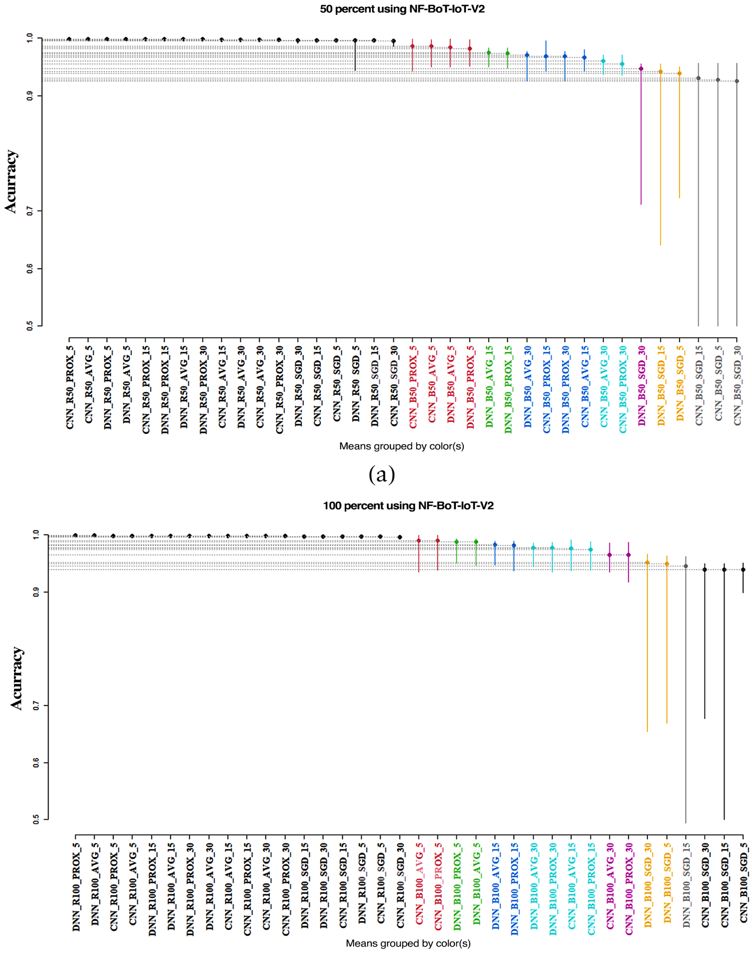
Figure 7. SK test results for FL configurations using NF-ToN-IoT-v2 with: (a) 50% of features; and (b) 100% of features.

- Figure 7(b) presents the SK results for 100% of the features, identifying seven clusters: (1) The first cluster contains 21 FL models, comprising 11 DNN models and 10 CNN models. The DNN models include all configurations using both data setups (raw and balanced), two aggregation methods (FedAVG and FedPROX) and three device counts (5, 15 and 30), except for DNN_R100_PROX_5, which is not included.

Similarly, the CNN models follow the same configurations as the DNN models, except that CNN_R100_AVG_5 is absent; (2) the second cluster includes three DNN models configured with balanced data, FedSGD and the three device counts (5, 15 and 30); (3) the third cluster contains two DNN models (DNN_R100_SGD_5 and DNN_R100_SGD_15) and one CNN model (CNN_B100_SGD_5); (4) the fourth cluster includes two models, CNN_B100_SGD_30 and DNN_R100_SGD_30; (5) the fifth cluster consists of four CNN models using FedSGD: CNN_B100_SGD_5, CNN_R100_SGD_15, CNN_R100_SGD_30 and CNN_B100_SGD_15; (6) the sixth cluster contains two models, DNN_R100_PROX_5 and CNN_R100_AVG_5; and (7) the final cluster includes a single model, CNN_R100_PROX_5.

For the NF-BoT-IoT-v2 dataset:

- Figure 8(a) displays the SK results for 50% of the features, revealing the presence of eight clusters. The first cluster consists of all classifiers based on the raw data setup (18 models), combining various DL base learners, federated aggregation methods and numbers of devices. Models using balanced data setups are distributed as follows: (1) the second cluster includes four models (two DNN and two CNN) using FedPROX and FedAVG with five devices; (2) the third cluster contains two DNN models using FedPROX and FedAVG with five devices; (3) the fourth cluster comprises two DNN models (DNN_B50_AVG_30 and DNN_R50_PRX_30) and two CNN models (CNN_B50_AVG_15 and CNN_R50_PRX_15); (4) the fifth cluster includes two CNN models, one using FedAVG with 30 devices and the other combining FedAVG and FedPROX; (5) the sixth cluster contains a single model, DNN_R50_SGD_30; (6) the seventh cluster consists of two DNN models using SGD with five and 15 devices; and (7) the final cluster comprises three CNN models using SGD with device counts of 5, 15 and 30.
- Figure 8(b) presents the SK results for 100% of the features, identifying nine distinct clusters. The first cluster includes all classifiers based on the raw data setup (18 models), incorporating various DL base learners, federated aggregation methods and device counts. The models using balanced data setups are distributed as follows: (1) the second cluster includes two CNN models using five devices with FedAVG and FedPROX; (2) the third cluster includes two DNN models using five devices with FedAVG and FedPROX; (3) the fourth cluster includes two DNN models using 15 devices with FedAVG and FedPROX; (4) the fifth cluster comprises two DNN models (DNN_B100_AVG_30 and DNN_B100_PROX_30) and two CNN models (CNN_B100_AVG_15 and CNN_B100_PROX_15); (5) the sixth cluster includes two CNN models using 30 devices with FedAVG and FedPROX; (6) the seventh cluster includes two DNN models employing the FedSGD aggregation method with five and 30 devices; (7) the eighth cluster contains a single model, DNN_B100_SGD_15; and (8) the final cluster comprises three CNN models using FedSGD with device counts of 5, 15 and 30.



(a)

(b)

Figure 8. SK test results for FL configurations using NF-BoT-IoT-v2 with: (a) 50% of features; and (b) 100% of features.`

In summary, for the NF-ToN-IoT-v2 dataset, the most effective configurations involve combining DNN with FedAVG and FedPROX, using both raw and balanced data setups. CNN models paired with FedAVG and FedPROX also demonstrated robust performance. For the NF-BoT-IoT-v2 dataset, the raw data setup showed optimal results when combined with various FL aggregation methods, including FedAVG, FedPROX and FedSGD. These findings indicate that combining a raw data setup with FedPROX or FedAVG and DNN architecture is generally the most effective approach for IoT security tasks. However, specific scenarios within the NF-BoT-IoT-v2 dataset may favor CNN architecture and benefit from balanced data setups.

The best model from the first SK cluster was identified using the BC system for each feature threshold, based on accuracy, AUC, recall and precision. The results are detailed in Tables 5 and 6 for NF-ToN-IoT-v2 and NF-BoT-IoT-v2, respectively. Furthermore, the comparison of top-performing models across different feature thresholds (17 and 35 features) shows that their performance is relatively consistent over both the NF-ToN-IoT-v2 and NF-BoT-IoT-v2 datasets, as detailed in Tables 5 and 6. Specifically, the best model: (1) for NF-ToN-IoT-v2 is DNN_R50_PROX_30, achieving an accuracy of 97.80%, an AUC of 0.9970, a recall of 98.86% and a precision of 97.72% using only 17 features; and (2) for NF-BoT-IoT-v2 is CNN_R50_PROX_5, with an accuracy of 99.87%, an AUC of 0. 9892, a recall of 99.99% and a precision of 99.88% using 17 features.

# of Features	Model	Accuracy	AUC	Recall	Precision	BC
17	DNN_R50_PROX_30	97.80%	0.9970	98.86%	97.72%	54
	CNN_R50_PROX_15	97.65%	0.9901	98.96%	97.58%	46
	CNN_R50_PROX_30	97.57%	0.9893	98.48%	97.71%	41
	DNN_R50_AVG_15	97.57%	0.9861	98.86%	97.39%	38
	CNN_B50_PROX_15	97.53%	0.9948	98.14%	96.97%	34
	DNN_R50_PROX_15	97.19%	0.9921	97.63%	97.98%	32
	DNN_R50_AVG_30	97.51%	0.9858	98.84%	97.32%	31
	DNN_B50_PROX_30	97.26%	0.9956	97.43%	97.12%	27
	DNN_B50_PROX_15	97.27%	0.9939	97.37%	97.18%	26
	CNN_B50_PROX_30	97.44%	0.9894	98.01%	96.92%	24
	DNN_B50_AVG_30	97.17%	0.9860	98.13%	96.28%	18
	DNN_B50_AVG_15	97.15%	0.9858	98.18%	96.23%	16
	DNN_B50_AVG_5	97.16%	0.9852	98.15%	96.26%	15
	DNN_R50_AVG_5	96.49%	0.9787	96.90%	97.68%	11
	CNN_B50_AVG_30	96.94%	0.9839	98.01%	96.03%	7
35	CNN_R100_PROX_15	98.45%	0.9988	98.97%	98.60%	78
	CNN_R100_PROX_30	98.39%	0.9986	99.02%	98.48%	76
	CNN_B100_PROX_15	98.32%	0.9987	98.28%	98.36%	66
	CNN_B100_PROX_30	98.26%	0.9986	98.23%	98.28%	59
	DNN_R100_PROX_30	98.08%	0.9982	99.00%	98.03%	59
	CNN_B100_AVG_15	98.27%	0.9985	98.19%	98.35%	57
	CNN_B100_AVG_30	98.22%	0.9985	98.18%	98.26%	51
	DNN_R100_PROX_15	97.98%	0.9970	98.91%	98.07%	46

# of Features	Model	Accuracy	AUC	Recall	Precision	BC
35	CNN_R100_AVG_15	97.92%	0.9973	98.80%	97.97%	42
	DNN_B100_AVG_30	98.05%	0.9981	98.10%	98.00%	39
	DNN_R100_AVG_5	97.78%	0.9971	98.79%	97.99%	38
	DNN_B100_PROX_30	97.83%	0.9982	98.12%	97.76%	35
	DNN_B100_PROX_15	97.83%	0.9976	98.22%	97.78%	35
	DNN_B100_AVG_5	97.73%	0.9977	98.23%	97.57%	31
	CNN_R100_AVG_30	97.59%	0.9959	98.62%	97.62%	25
	DNN_R100_AVG_30	97.39%	0.9962	98.94%	97.40%	25
	CNN_B100_AVG_5	97.52%	0.9906	96.59%	98.20%	18
	CNN_B100_PROX_5	97.68%	0.9959	98.09%	97.65%	17
	DNN_B100_AVG_15	97.40%	0.9974	98.10%	97.26%	16
	DNN_R100_AVG_15	96.97%	0.9899	98.94%	96.99%	16
	DNN_B100_PROX_5	97.61%	0.9955	97.92%	97.44%	11

Table 5. BC ranking of FL variants within the best SK cluster for each feature threshold over NF-ToN-IoT-v2.

# of Features	Model	Accuracy	AUC	Recall	Precision	BC
17	CNN_R50_PROX_5	99.87%	0.9892	99.99%	99.88%	60
	CNN_R50_AVG_5	99.87%	0.9901	99.99%	99.88%	57
	DNN_R50_PROX_5	99.86%	0.9934	99.99%	99.87%	50
	DNN_R50_AVG_5	99.86%	0.9894	99.99%	99.87%	48
	CNN_R50_PROX_15	99.81%	0.9744	99.99%	99.81%	48
	DNN_R50_PROX_15	99.80%	0.9874	99.99%	99.81%	47
	DNN_R50_AVG_15	99.80%	0.9880	99.99%	99.81%	42
	DNN_R50_PROX_30	99.79%	0.9836	99.99%	99.80%	41
	DNN_R50_AVG_30	99.78%	0.9570	99.99%	99.78%	37
	CNN_R50_AVG_15	99.78%	0.9770	99.99%	99.79%	36
	CNN_R50_SGD_15	99.59%	0.2579	100.00%	99.59%	25
	CNN_R50_PROX_30	99.76%	0.9573	99.98%	99.77%	22
	CNN_R50_AVG_30	99.76%	0.9506	99.98%	99.77%	21
	DNN_R50_SGD_15	99.58%	0.4102	100.00%	99.58%	21
	CNN_R50_SGD_5	99.59%	0.2083	100.00%	99.59%	20
	DNN_R50_SGD_30	99.62%	0.6877	99.99%	99.63%	17
	DNN_R50_SGD_5	99.59%	0.7425	99.94%	99.64%	12
	CNN_R50_SGD_30	99.57%	0.2178	99.99%	99.58%	8
35	DNN_R100_AVG_5	99.89%	0.9957	99.99%	99.90%	59
	DNN_R100_PROX_5	99.89%	0.9950	99.99%	99.90%	58
	DNN_R100_PROX_15	99.86%	0.9894	99.99%	99.87%	52
	DNN_R100_AVG_15	99.85%	0.9857	99.99%	99.86%	47
	CNN_R100_PROX_5	99.87%	0.9780	99.99%	99.89%	43
	CNN_R100_AVG_5	99.87%	0.9808	99.99%	99.88%	41
	DNN_R100_PROX_30	99.85%	0.9823	99.99%	99.86%	40

Table 6. (*continues*).

# of Features	Model	Accuracy	AUC	Recall	Precision	BC
35	DNN_R100_AVG_30	99.85%	0.9789	99.99%	99.86%	35
	CNN_R100_AVG_15	99.85%	0.9687	99.99%	99.86%	34
	CNN_R100_PROX_15	99.84%	0.9674	99.99%	99.86%	30
	DNN_R100_SGD_15	99.75%	0.9206	100.00%	99.75%	27
	DNN_R100_SGD_30	99.71%	0.8199	100.00%	99.71%	25
	CNN_R100_AVG_30	99.83%	0.9537	99.99%	99.85%	23
	CNN_R100_SGD_15	99.69%	0.6649	100.00%	99.69%	22
	DNN_R100_SGD_5	99.69%	0.8171	100.00%	99.69%	22
	CNN_R100_PROX_30	99.83%	0.9444	99.99%	99.84%	18
	CNN_R100_SGD_5	99.68%	0.5103	100.00%	99.68%	18
	CNN_R100_SGD_30	99.67%	0.6192	100.00%	99.68%	18

Table 6. BC ranking of FL variants within the best SK cluster for each feature threshold over NF-BoT-IoT-v2.

4.3 Optimal Deep Learning Base Learners for Federated Learning Models in IoT Intrusion Detection (RQ3)

This subsection evaluates the impact of DNN and CNN base learners on the performance of FL methods to identify the most effective learner for enhancing classification accuracy in an FL setup. The analysis considers each dataset, data configuration, feature threshold, aggregation method and device count. The SK test, based on accuracy as shown in Figures 7 and 8, was employed to compare the performance of different models. Additionally, the BC method was used to rank the FL models within the top SK clusters based on accuracy, AUC, recall and precision metrics. Tables 5 and 6 present the rankings of the FL models within the best SK clusters for the NF-ToN-IoT-v2 and NF-BoT-IoT-v2 datasets, respectively. To identify the overall top-performing DL base learner, independent of the dataset and feature thresholds, the occurrences of each learner in the best SK clusters and their BC ranks were tallied, as shown in Table 7.

DL Base Learner	Occurrences in the Best Cluster	Median Ranking in the Best Cluster
DNN	39	9
CNN	33	10

Table 7. BC ranking and occurrences in the best cluster for DL base learners.

As shown in Table 7, the results indicate that DNN is the top-performing DL base learner for cyber detection in the FL context. DNN appears in the best cluster 39 times, compared to CNN, which appears 33 times. Additionally, DNN achieves a better median rank of 9 in the best cluster, compared to CNN's median rank of 10. These findings demonstrate that DNN-based techniques significantly enhance anomaly detection in the FL context within IoT environments.

4.4 Results Comparison with State-of-the-Art Methods

In this subsection, we contrast our experimental outcomes with those from earlier research. Table 8 shows the performance of the best data setups, aggregation methods and device counts from our study, alongside results from previous studies, focusing on key performance metrics. Table 8 acts as a reference for assessing the progress and success of our method. This comparison aims to provide a clear and impartial evaluation of our approach's effectiveness in the field of FL-IDS.

Table 8 succinctly juxtaposes the performance metrics of our investigation with the antecedent studies. This comparative analysis underscores a pivotal advancement: Our methodology exhibits superior performance across critical dimensions, including accuracy, AUC, recall and precision, thereby substantiating its efficacy as a formidable FL-IDS framework. The empirical evidence presented in Table 8 attests to the methodological enhancements facilitated by our approach, thereby augmenting the efficacy and adaptability of FL-IDS paradigms.

Publication	Dataset	Technique	# of Features	Accuracy	AUC	Precision
Our study	NF-ToN-IoT-v2	DNN	17	97.80%	0.9970	98.86%
	NF-BoT-IoT-v2	CNN		99.87%	0.9892	99.99%
Chen et al. [24]	NF-ToN-IoT-v2	F-NIDS		96.20%	-	88.50%
Sarhan et al. [31]	NF-BoT-IoT-v2	F-DNN	43	93.08%	0.9560	-
Sarhan et al. [32]	NF-BoT-IoT-v2	HBFL		99.46%	-	96.86%
Yu [33]	NF-ToN-IoT-v2	FedAVG		92.78%	0.9170	92.76%

Table 8. Performance results comparison with previous studies.

5. Limitations and Validity

To ensure the reliability of this study, it is important to clearly define the scope and limitations of the conclusions. Certain classification tasks were not conducted due to the large size of some datasets and the significant time required for exhaustive model parameter tuning. Additionally, the base learners used in this paper were designed without detailed hyperparameter optimization. Adjusting these hyperparameters could influence the performance of FL models and may lead to improved outcomes in IDS-based FL applications.

This paper also relied on a simulated FL environment using TFF to emulate the behavior of IoT devices. While this approach provides flexibility and scalability during experimentation, future work should aim to deploy these models on real IoT hardware using lightweight frameworks such as TinyML. Furthermore, evaluating FL performance over resource-constrained wireless communication protocols like LoRaWAN would provide more realistic insights into the practical deployment challenges of FL in IoT networks.

The findings presented here offer valuable insights for advancing future research on FL, especially in designing more practical and efficient FL models for cybersecurity in IoT and related domains.

6. Conclusion and Future Work

The study evaluated and compared 144 federated learning (FL) configurations for binary classification of network intrusions, using dense neural networks (DNNs) and convolutional neural networks (CNNs) as base learning models. The configurations incorporated three aggregation methods (FedAVG, FedPROX and FedSGD), three device scenarios (5, 15 and 30 devices), two feature thresholds derived from ANOVA and Chi2 as FS techniques, and two data setups (raw and balanced). The analysis was conducted on two NetFlow IoT datasets, NF-ToN-IoT-v2 and NF-BoT-IoT-v2. Evaluation metrics included accuracy, AUC, recall and precision, complemented by the SK statistical test and the BC ranking system. The key findings of this research are summarized as follows: (1) FedAVG and FedPROX consistently demonstrated superior final accuracy and faster convergence compared to FedSGD; (2) DNN architecture with FedPROX or FedAVG was typically the most effective approach for IoT security; and (3) DNN was identified as the most effective deep learning (DL) base learner for cyber detection in the FL context, consistently outperforming CNN in best cluster appearances and median ranking. These results highlighted the importance of using FL to develop a decentralized IDS tailored for Internet of Things (IoT) networks to detect attacks.

We intend future work to expand empirical evaluations to further validate or challenge these findings. This may involve testing with diverse datasets to assess the robustness and adaptability of FL-based intrusion detection systems (IDS) across various IoT environments. Additionally, exploring alternative models within FL frameworks could offer valuable insights for optimizing performance and efficiency. Further research could also focus on deploying these models on embedded devices using TinyML and FL methodologies.

References

[1] G. Atharvan, S. Koolikkara Madom Krishnamoorthy, A. Dua and S. Gupta, “A Way Forward towards a Technology-Driven Development of Industry 4.0 Using Big Data Analytics in 5G-Enabled IIoT,” *International Journal of Communication Systems*, **35**(1), 2022 e5014. doi:10.1002/DAC.5014.

[2] A. Zoubir and B. Missaoui, “Graph Neural Networks with Scattering Transform for Network Anomaly Detection,” *Engineering Applications of Artificial Intelligence*, **150**, 2025 110546. doi:10.1016/J.ENGAPPAI.2025.110546.

[3] E. T. M. Beltran, M. Q. Pérez, P. M. S. Sanchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez and A. H. Celdrán, “Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges,” *IEEE Communications Surveys and Tutorials*, **25**(4), 2023 pp. 2983–3013. doi:10.1109/COMST.2023.3315746.

[4] X. Hei, X. Yin, Y. Wang, J. Ren and L. Zhu, “A Trusted Feature Aggregator Federated Learning for Distributed Malicious Attack Detection,” *Computers & Security*, **99**, 2020 102033. doi:10.1016/J.COSE.2020.102033.

[5] T. T. Huong, T. P. Bac, D. M. Long, B. D. Thang, N. T. Binh, T. D. Luong and T. K. Phuc, “LocKedge: Low-Complexity Cyberattack Detection in IoT Edge Computing.” arxiv.org/abs/2011.14194v1.

[6] 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,” *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, Los Alamitos, CA: IEEE Computer Society, 2019 pp. 756–767. doi:10.1109/ICDCS.2019.00080.

[7] P. Boobalan, S. P. Ramu, Q.-V. Pham, K. Dev, S. Pandya, P. K. R. Maddikunta, T. R. Gadekallu and T. Huynh-The, “Fusion of Federated Learning and Industrial Internet of Things: A Survey,” *Computer Networks*, **212**, 2022, 109048. doi:10.1016/j.comnet.2022.109048.

[8] A. Hamdouchi and A. Idri, “Optimizing Federated Learning for Intrusion Detection in IoT Networks,” in *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Volume 1*, Porto, Portugal (F. Coenen, A. Fred and J. Bernardino, eds.), Setúbal, Portugal: SciTePress, 2024 pp. 448–456. doi:10.5220/0013040000003838.

[9] A. Hamdouchi and A. Idri, “Comprehensive Evaluation of Federated Learning Configurations for Intrusion Detection in IoT Contexts,” *2024 World Conference on Complex Systems (WCCS)*, Mohammedia, Morocco, (M. Essaaidi, M. Nemiche and S. Tayane, eds.), Piscataway, NJ: IEEE, 2024 pp. 1–6. doi:10.1109/WCCS62745.2024.10765581.

[10] Z. Yang, M. Zhou, H. Yu, R. O. Sinnott and H. Liu, "Efficient and Secure Federated Learning with Verifiable Weighted Average Aggregation," *IEEE Transactions on Network Science and Engineering*, **10**(1), 2023 pp. 205–222. doi:10.1109/TNSE.2022.3206243.

[11] K. Pillutla, S. M. Kakade and Z. Harchaoui, "Robust Aggregation for Federated Learning," *IEEE Transactions on Signal Processing*, **70**, 2022 pp. 1142–1154. doi:10.1109/TSP.2022.3153135.

[12] A. Hamdouchi and A. Idri, "Enhancing IoT Security through Boosting and Feature Reduction Techniques for Multiclass Intrusion Detection," *Neural Computing and Applications*, **37**(31), 2025 pp. 25721–25744. doi:10.1007/S00521-025-11001-2.

[13] A. Hamdouchi and A. Idri, "Multiclass Intrusion Detection in IoT Using Boosting and Feature Selection," *Good Practices and New Perspectives in Information Systems and Technologies (WorldCIST 2024)*, Lodz, Poland (Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira and A. Poniszewska-Maranda, eds.), Cham, Springer: 2024 pp. 128–137. doi:10.1007/978-3-031-60221-4_13.

[14] A. Hamdouchi and A. Idri, "New Design Strategies for IoT Intrusion Detection Using Boosting and Feature Selection," *The Journal of Supercomputing*, **81**(13), 2025 1273. doi:10.1007/S11227-025-07755-0.

[15] A. Hamdouchi and A. Idri, "Empowering IoT Security: Deploying TinyML Ensemble Techniques for Cyberattack Detection," *Scientific African*, **29**, 2025 e02809. doi:10.1016/J.SCIAF.2025.E02809.

[16] S. Chauhan, L. Mahmoud, S. Gangopadhyay and A. K. Gangopadhyay, "A Comparative Study of LAD, CNN and DNN for Detecting Intrusions," *Intelligent Data Engineering and Automated Learning–IDEAL 2022*, Manchester, UK (H. Yin, D. Camacho and P. Tino, eds.), Cham: Springer, 2022 pp. 443–455. doi:10.1007/978-3-031-21753-1_43.

[17] A. Hamdouchi and A. Idri, "Evaluating the Performance of TinyML Singular and Ensemble Techniques for Intrusion Detection in IoT Networks," *Microprocessors and Microsystems*, **117**, 2025 105172. doi:10.1016/J.MICPRO.2025.105172.

[18] M. Arief and S. H. Supangkat, "Comparison of CNN and DNN Performance on Intrusion Detection System," in *Proceedings of the 9th International Conference on ICT for Smart Society (ICISS 2022)*, Bandung, Indonesia, Piscataway, NJ: IEEE, 2022 pp. 1–7.
doi:10.1109/ICISS55894.2022.9915157.

[19] L. Fu, H. Zhang, G. Gao, M. Zhang and X. Liu, "Client Selection in Federated Learning: Principles, Challenges, and Opportunities," *IEEE Internet of Things Journal*, **10**(24), 2023 pp. 21811–21819. doi:10.1109/JIOT.2023.3299573.

[20] W. Chen, S. Horvath and P. Richtarik, "Optimal Client Sampling for Federated Learning." arxiv.org/abs/2010.13723v3.

[21] G. Naidu, T. Zuva and E. M. Sibanda, “A Review of Evaluation Metrics in Machine Learning Algorithms,” *Artificial Intelligence Application in Networks and Systems (CSOC 2023),* On-line Conference, (R. Silhavy and P. Silhavy, eds.), Cham: Springer, 2023 pp. 15–25. doi:10.1007/978-3-031-35314-7_2.

[22] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha and G. Srivastava, “Federated-Learning-Based Anomaly Detection for IoT Security Attacks,” *IEEE Internet of Things Journal*, **9**(4), 2022 pp. 2545–2554. doi:10.1109/JIOT.2021.3077803.

[23] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini and A. Skarmeta, “Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges,” *Computer Networks*, **203**, 2022 108661. doi:10.1016/J.COMNET.2021.108661.

[24] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen and W. Pan, “Intrusion Detection for Wireless Edge Networks Based on Federated Learning,” *IEEE Access*, **8**, 2020 pp. 217463–217472. doi:10.1109/ACCESS.2020.3041793.

[25] M. Althunayyan, A. Javed, and O. Rana, “A Robust Multi-stage Intrusion Detection System for In-Vehicle Network Security Using Hierarchical Federated Learning,” *Vehicular Communications*, **49**, 2024 100837. doi:10.1016/J.VEHCOM.2024.100837.

[26] S. M. S. Bukhari, M. H. Zafar, M. A. Houran, S. K. R. Moosavi, M. Mansoor, M. Muaaz and F. Sanfilippo, “Secure and Privacy-Preserving Intrusion Detection in Wireless Sensor Networks: Federated Learning with SCNN-Bi-LSTM for Enhanced Reliability,” *Ad Hoc Networks*, **155**, 2024 103407. doi:10.1016/J.ADHOC.2024.103407.

[27] Z. Jin, J. Zhou, B. Li, X. Wu and C. Duan, “FL-IIDS: A Novel Federated Learning-Based Incremental Intrusion Detection System,” *Future Generation Computer Systems*, **151**, 2024 pp. 57–70. doi:10.1016/J.FUTURE.2023.09.019.

[28] L. Deri NETikos SpA, “nProbe: an Open Source NetFlow Probe for Gigabit Networks,” (Dec 2, 2025) www.ntop.org/products/netflow-probes/nprobe.

[29] M. Sarhan, S. Layeghy and M. Portmann, “Towards a Standard Feature Set for Network Intrusion Detection System Datasets,” *Mobile Networks and Applications*, **27**(1), 2022 pp. 357–370. doi:10.1007/S11036-021-01843-0.

[30] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood and A. Anwar, “TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems,” *IEEE Access*, **8**, 2020 pp. 165130–165150. doi:10.1109/ACCESS.2020.3022862.

[31] M. Sarhan, S. Layeghy, N. Moustafa and M. Portmann, “Cyber Threat Intelligence Sharing Scheme Based on Federated Learning for Network Intrusion Detection,” *Journal of Network and Systems Management*, **31**(1), 2023 3. doi:10.1007/S10922-022-09691-3.

[32] M. Sarhan, W. W. Lo, S. Layeghy and M. Portmann, “HBFL: A Hierarchical Blockchain-Based Federated Learning Framework for Collaborative IoT Intrusion Detection,” *Computers and Electrical Engineering*, **103**, 2022 108379. doi:10.1016/J.COMPELECENG.2022.108379.

[33] J. C. Yu, “Assessing Industrial Internet of Things Security at the Network Edge Using Trust-based Centralized and Federated Machine Learning,” Doctoral dissertation, George Washington University, 2024.
www.proquest.com/openview/cec612306328d69a40d49c6fdf53406c/1.

[34] G. Naidu, T. Zuva and E. M. Sibanda, “A Review of Evaluation Metrics in Machine Learning Algorithms,” *Artificial Intelligence Application in Networks and Systems (CSOC 2023)*, (R. Silhavy and P. Silhavy, eds.), Cham: Springer, 2023 pp. 15–25.
doi:10.1007/978-3-031-35314-7_2.

[35] A. J. Scott and M. Knott, “A Cluster Analysis Method for Grouping Means in the Analysis of Variance,” *Biometrics*, **30**(3), 1974 pp. 507–512. doi:10.2307/2529204.

[36] D. G. Saari, “Selecting a Voting Method: The Case for the Borda Count,” *Constitutional Political Economy*, **34**(3), 2023 pp. 357–366. doi:10.1007/S10602-022-09380-Y.