Collective Intelligence for Cybersecurity: Federated Learning under Non-IID Conditions for Intrusion Detection

Hutheifa Anwar Mohammed [1)]*, **Awos Kh. Ali** [2)]
[1,2)] Department of Computer Science, Mosul University, Mosul, Iraq
[1)] hutheifa.23esp5@student.uomosul.edu.iq, [2)] a.k.ali@uomosul.edu.iq

Submitted :Jul 9, 2025 | **Accepted** : Aug 15, 2025 | **Published** : Oct 2, 2025

Abstract: Cyber threats are becoming increasingly complex in cyberspace, which highlights the necessity for strong Intrusion Detection Systems (IDS). However, traditional centralized IDS methods have large problems with data privacy and scalability. Federated Learning (FL) is an intriguing new idea that lets multiple clients train a model together without sharing data directly, which keeps privacy intact. The proposed federated intrusion detection model develops and assesses FL models for detecting network intrusions, focusing on the important issue of non-independent and non-identically distributed (non-IID) data among clients. This work implements and compares two widely recognized FL algorithms, Federated Averaging (FedAvg) and Federated Proximal (FedProx), using a 1D Convolutional Neural Network (CNN) architecture specifically designed for tabular network traffic data. The authors utilize a Dirichlet distribution (α=0.1) to distribute the data among 10, 20, and 30 clients, thereby simulating non-IID conditions in the experiment. The authors thoroughly compare the performance of algorithms using two benchmark datasets: NSL-KDD and NF-Bot-Net-V2. The comparison reveals that while both FedAvg and FedProx achieve high detection rates on NSL-KDD, FedProx is more capable of maintaining stability and converging on the more complex NF-Bot-Net-V2 dataset, achieving an accuracy of 0.9953. The results highlight that FedProx is a more appropriate algorithm for implementing robust and privacy-preserving federated intrusion detection systems in statistically heterogeneous network environments found in the real world.

Keywords: cybersecurity; Federated learning; Intrusion detection system; Non-independent and not identically distributed data (non-IID); Network

INTRODUCTION

The rise in the number of internet users, the growth of digital transactions, and the rapid technological transformation in various sectors, including healthcare, Industry 4.0, and smart home applications, have increased the risk of sensitive data being exposed to potential hackers for both individuals and organizations (Ahmed et al., 2025).

The growing frequency and sophistication of cyberattacks across various sectors highlight the critical need for an advanced system to detect and categorize threats, ensuring the devices security and confidentiality of data. In this situation, Intrusion detection systems (IDS) perform a pivotal function in ensuring devices and networks security through continuous monitoring of network traffic to identify malicious activities, rule violations, or other suspicious behavior (Aljanabi et al., 2021). IDS are traditionally categorized into two main types: signature-based (SIDS) and anomaly-based (AIDS) systems. SIDS operate by utilizing a database of known attack signatures to detect threats, while AIDS identifies potential anomalies by recognizing deviations from a predefined baseline of normal network activity. SIDS are effective at quickly addressing known attacks with minimal false positives; however, they cannot detect new or previously unknown threats (zero-day attacks) and require regular updates to remain effective (Abdulganiyu et al., 2023).

The AIDS method, currently the most popular approach used, utilizes ML to improve adaptability and efficiency in identifying emerging threats. By analyzing complex patterns in high-dimensional data, ML and DL offer an effective solution for detecting attacks. However, traditional methods of IDS that are based on ML often depend on centralized data processing. This data may be personal or sensitive, which is very dangerous for data privacy

* Hutheifa Anwar Mohammed

and security. Furthermore, data transmission can also lead to significant communication overhead, potentially resulting in delays (Hernandez-Ramos et al., 2025).

To address these challenges, researchers have developed a novel approach to distributed learning that is widely recognized today as federated learning (FL). FL has emerged as a leading distributed learning approach for building robust models while maintaining the data privacy of entities belonging to the FL system. FL is different from centralized learning frameworks because it automatically protects privacy and confidentiality. This is because data created on an end device stays on that device and does not leave it. FL also reduces latency and minimizes the need for extensive data transfers (Brendan McMahan , Eider Moore , Daniel Ramage , Seth Hampson & Arcas, 2017). However, in real-world environments, edge devices typically generate data that is non-IID (non-independent and not identically distributed). Training on such non-IID data with current ML techniques can lead to reduced model accuracy and potential convergence issues. To address the limitations in existing research, the authors conducted a study of FL-based IDS in non-IID settings. The study looks at how non-IID settings affect FL model training. To mimic real-world situations in these experiments, the authors use a method that relies on the Dirichlet distribution to create specific non-IID data groups for each client. Additionally, the authors propose using the SOMTEENN algorithm to enhance the data quality by balancing classes and eliminating unwanted and noisy traffic. **The contributions of this study are as follows:**

- This research proposes a hybrid methodology that combines CNN with FL to improve the detection and classification of cyber threats.
- It investigates the effects of non-IID data distribution scenarios on the performance of the proposed FL-IDS models.
- It examines the influence of varying numbers of participating clients on model performance, with particular emphasis on environments characterized by non-IID data distributions.
- It provides a performance comparison of the FedAvg and FedProx algorithms, supported by extensive experimental evaluations on the NF-Botnet-V2 and NSL-KDD datasets.

The remainder of this paper is organized as follows:

Section 2 looks at previous research on FL-based IDS and points out the issues with dealing with non-IID data distributions. Section 3 explains the proposed structure of the federated system, the distribution of non-IID data, and the architecture of the local model. Section 4 details the experimental setup, including hardware and software configurations, datasets, and evaluation metrics. Section 5 presents the results and analysis, focusing on client data distributions and comparisons between FedProx and FedAvg. Finally, Section 6 discusses insights regarding the application of FL-IDS in non-IID environments.

RELATED WORK

The related work is divided into two parts: the first explores the development of general FL-based IDS, while the second delves into studies tackling non-IID data challenges in FL-based IDS. The second serves as the primary motivation for this proposed approach.

General FL-based IDS

Several survey papers explore the role and influence of FL in cybersecurity, particularly in the context of intrusion detection. The authors (Lavaur et al., 2022) provide a systematic literature review to determine the relevant advancements in FL-based IDSs from their inception in 2016 to 2021.

Moreover, (Fedorchenko et al., 2022) examine several existing FL-based systems for IDS in detail, and they meticulously assess the benefits and challenges of these solutions, encompassing the architecture of the suggested IDS and the methodologies for data division among clients.

In other study, (Alazab et al., 2023) offer of how FL improves IDS. This work evaluated the efficacy of FL for IDS. Using random client selection, FL exceeded DL in IDS, obtaining better accuracy and reducing loss. The experiment used the NSL-KDD dataset for network intrusion detection. The results imply that the federated average in FL can improve IDS solutions, therefore making them safer, more effective, and more efficient.

Furthermore, Generative adversarial networks (GANs) possess the capability to create and augment data. Sometimes, systems lack adequate and varied datasets that are necessary to create an effective model. (Tabassum et al., 2022) propose a system named FEDGAN-IDS, which operates on a federated learning framework for security threat detection. This system uses the GAN algorithm on local devices to train using augmented data. The model achieves 99% and 98% accuracy for binary and multiclass classification, respectively.

Although the aforementioned works discuss various FL challenges for IDS broadly, they do not specifically emphasize the issue of non-IID data distribution.

Non-IID-Aware FL-IDS

Non-iid data denotes the variability in the distribution of data among various clients. The variability presents difficulties in developing FL models, as they must generalize across varied datasets.

To tackle the above challenge (Popoola et al., 2021) proposed Federated Deep Learning (FDL) for IDS in heterogeneous wireless networks. The system employs DNN models on multiple edge nodes to learn representations of private network traffic data. A central server aggregates the parameters of these local models using the Fed+ fusion algorithm to produce a global FDL model. The study utilized four datasets for network intrusion detection (NF-TON-IoT-v2, NF-UNSW-NB15-v2, NF-BoT-IoT-v2, and NF-CSE-CIC-IDS2018-v2) to model a heterogeneous network environment. Simulation results indicated that while local DNN models had a high attack detection rate, their false alarm rate and generalization ability were poor. In contrast, the DNN-FedAvg+ and DNN-CM+ models, which utilized Fed+ fusion algorithms, demonstrated higher Performance of classification and better generalization ability, outperforming state-of-the-art fusion algorithms like Coordinate Median (CM) and FedAvg.

Additionally, (Weinger et al., 2022) explored enhancing the performance of anomaly detection in IoT environments using FL. They point out difficulties in using FL for detecting anomalies in IoT, especially because the data is spread out over many devices, each holding only a small part of the total data, possibly dealing with uneven data distribution and differences between devices. To address these challenges, the authors investigated the application of data augmentation techniques within the FL framework. They looked at how well different data augmentation methods worked, such as random oversampling, stratified oversampling, SMOTE, ADASYN, and GANs, by testing them on three publicly available IoT datasets (Modbus and Weather from the TON_IoT collection and DS2OS). Stratified random sampling and uniform random sampling showed the most significant improvement with a modest increase in computation time, while the GAN-based approach was computationally expensive with limited performance benefits. The study suggests that data augmentation, particularly random and stratified oversampling, is a viable approach to mitigate the performance degradation of FL in IoT anomaly detection caused by class imbalance and data heterogeneity.

Using client data that is non-independent and identically distributed, (Liu et al., 2023) investigate the utility of FL in IDS under such circumstances. While FL enables decentralized training, it faces challenges with performance degradation due to data heterogeneity. The authors describe a data augmentation system based on ACGAN that allows clients to generate synthetic attack data for classes that are uncommon or underrepresented, thereby addressing the issue. The server collects statistical information about the distributions, trains an ACGAN model, and disseminates it to clients, who then construct locally balanced datasets. Tests on the UNSW-NB15 dataset show that their method greatly increases both accuracy and speed in various non-IID situations, doing better than other rebalancing methods like SMOTE, especially when clients face different kinds of attacks.

Additionally, (Wang et al., 2021) proposed a peer-to-peer FL technique with data rebalancing for anomaly detection in IoT edge contexts with non-IID data and poor connection. Traditional FL needs a central orchestrator, which might slow IoT networks with sporadic connection. Unbalanced anomaly detection datasets, which contain a disproportionately small number of aberrant data points, can also degrade model performance. The researchers developed this framework to allow edge devices to natively train anomaly detection models without a central server. They also used data rebalancing to reduce uneven and non-IID device data distributions.

In general, significant limitations persist in current research on FL-based IDS in non-IID situations. Consequently, this research investigates diverse non-IID conditions in detail, presents a specific partitioning mechanism, and recommends an enhanced aggregation strategy to mitigate the poor performance of FL in the presence of non-IID data.

METHOD

This section outlines the methodology employed in this study. It begins with a description of the datasets and the steps taken to prepare them, followed by an explanation of how to simulate non-IID data distributions. The subsequent sections will discuss the local model architecture, system configuration, and the proposed FL-based IDS framework. Finally, the section will conclude with the performance metrics used to evaluate the framework.

Dataset and Preprocessing

This study employs two publicly accessible datasets, NF-BoT-IoT-v2 and NSL-KDD. These datasets were selected because they effectively show both normal and harmful traffic patterns, which helps in testing the proposed intrusion detection framework. The description of both datasets is as follows:

1) NF-BoT-IoT-V2. The NetFlow technique created this version 2 dataset, which includes 43 extended NetFlow features (Sarhan et al., 2022). The original BoT-IoT dataset gave rise to the NF-BoT-IoT-V2 dataset in 2021. Feature data was extracted from both the original and available PCAP files. The dataset comprises a total of

37,763,497 data flows, with 99.64% (37,628,460) classified as attack samples and 0.36% (135,037) identified as benign. The dataset includes four attack scenarios: DDoS, DoS, reconnaissance, and theft.

2) NSL-KDD. The original KDD Cup 1999 dataset had problems with repeated and duplicate records, so this dataset was made to fix those problems and help make classification models less biased (Dhanabal & Shantharajah, 2015). Owing to its public availability and widespread adoption, it has become established as a benchmark dataset within IDS research. The dataset encompasses 41 features characterizing various attributes of network traffic. These include protocol type, service type, connection flags, and several calculated network statistics, such as connection duration and the frequency of failed login attempts. Traffic examples in the NSL-KDD dataset are divided into normal traffic and four different types of attacks: Denial of Service (DoS), probe, User-to-Local (U2L), and Remote-to-Local (R2L).

Preprocessing: In the realm of data preprocessing, a series of pivotal steps were meticulously executed to ensure the integrity and quality of the datasets. These measures were implemented with the objective of facilitating the effective training and evaluation of models.

During the first step of preparing the data, the researchers intentionally omitted the attributes IPV4_SRC_ADDR and IPV4_DST_ADDR from the NF-Botnet-V2 dataset to avoid any bias towards the attacker and victim endpoint nodes, resulting in a simpler set of 41 features. In contrast, the NSL-KDD dataset retained all original attributes. Subsequently, both datasets were carefully inspected for inconsistencies, missing values, duplicate traffic records, and non-numeric entries. Duplicate instances detected in the NF-Botnet-V2 dataset were removed. For the NSL-KDD dataset, three categorical features were converted into numerical form via label encoding. Given the large size of the NF-Botnet-V2 dataset, direct training on the complete dataset was computationally prohibitive. Therefore, a stratified random sampling approach was applied, selecting 377,780 samples while preserving the overall class distribution. In contrast, the NSL-KDD dataset was fully utilized in the analysis.

To overcome possible class imbalance in both datasets, the Synthetic Minority Over-sampling Technique combined with the Edited Nearest Neighbors (SMOTEENN) approach was utilized. This method helps balance the class distribution by oversampling the minority class, while the ENN part of SMOTE-ENN removes noise and mislabeled traffic from the dataset that was increased with SMOTE, making it better and more balanced (Yang et al., 2022). Fig.1 shows the distribution of classes before and after applying SMOTEENN for both datasets.

Once the balancing procedure is complete for both datasets, apply the StandardScaler technique to scale the features. This standardization step transforms the data such that each feature has a mean of zero and a standard deviation of one, which is often crucial for the optimal performance of many DL algorithms. At the same time, the target variable, which shows the classification labels, was changed into numbers using the LabelEncoder technique, making it easier for the model to learn. Finally, the preprocessed dataset was partitioned into training and testing sets, allocating 70% of the data for model training and the remaining 30% for performance evaluation to each dataset. Table 1 shows the number of samples for both datasets.

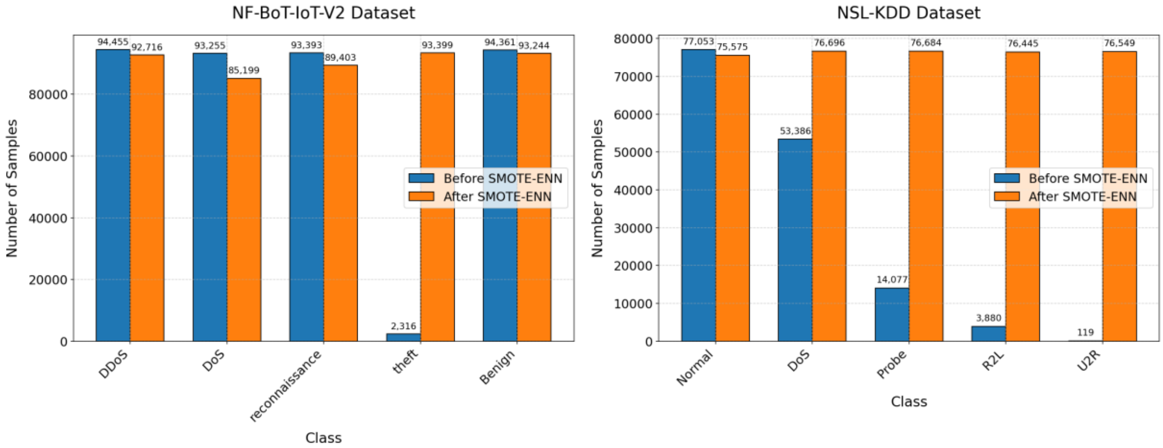


Fig. 1 Class Distribution Before vs. After SMOTE-ENN.

Non-IID Data Distribution

In FL, the IID assumption of data among clients is commonly violated in reality. In realistic IDS, data is gathered from diverse sources, leading to non-IID (statistically different) distributions. This heterogeneity can dramatically affect the convergence rate and performance of traditional FL algorithms (Zhao et al., 2018) (Al-Dabbagh & Ali, 2022). To replicate this real-life scenario in the experiments, a technique using the Dirichlet distribution is applied to create different non-IID groups of data for each client (Hsu et al., 2019). For a dataset with C classes and K clients, the data is distributed among clients using samples from a Dirichlet distribution, Dir (α). The concentration parameter α, which can take any positive value, regulates the degree of variation of the data among the clients. A small value of α (e.g., 0.1 or 0.2) leads to highly skewed distributions where data from only a few classes is received by a majority of the clients. On the other hand, IID settings are approximated with a large value of α, where clients receive a closer to uniform class distribution. In this study, α=0.1 is used, reflecting a high level of statistical heterogeneity. This configuration simulates IDS settings where clients monitor varying network behaviors and different attack profiles. The distribution algorithm operates in two steps: First, for each class c, data indices are first shuffled randomly and then divided among the clients in proportions sampled from Dir (α). Second, each client's final dataset is formed by consolidating its allocated data from all classes, creating skewed and unbalanced class distributions. This way, the execution of federated optimization algorithms such as FedProx (Li et al., 2020) and FedAvg (Brendan McMahan , Eider Moore , Daniel Ramage , Seth Hampson & Arcas, 2017) can be tested in real-world scenarios where data is not uniformly distributed.

Table 1 shows the number of samples for both data sets.

Dataset	Attack Types	Number of samples		Training samples	Test samples
		Before SMOTE-ENN	After SMOTE-ENN		
NF-BoT-IoT-V2	DDoS, DoS, reconnaissance, and theft	377,780	453,960	317,772	136,188
NSL-KDD	DoS, probe, U2L, and R2L.	148,515	381,949	267,364	114,585

Local Model

DL-based CNN models are popular because they can find and study patterns in images, as well as in computer vision, anomaly detection, and other areas. An Enhanced One-Dimensional Convolutional Neural Network (1D-CNN) is used as the main tool for each client in a federated system to detect different kinds of network intrusions in traffic data. The model is built to find patterns in network data, making it suitable for detecting intrusions in which obvious indications of an attack are displayed by the packet's sequential order.

In terms of architecture, the proposed improved 1D-CNN classifier starts with an input layer that takes in fixed-length feature vectors from preprocessed network traffic. These feature vectors include flow-level features like packet length, duration, and bytes per flow. Following this, three 1D convolutional layers are employed, utilizing filter sizes of 32, 64, and 128, respectively, along with a kernel size of 3. The ReLU activation introduces non-linearity after each convolutional layer. It's especially important to normalize the data and reduce the chance of overfitting for client models that might be trained on small or unevenly distributed datasets. Batch normalization and dropout techniques are applied after the convolutional layers. A max pooling layer achieves dimensionality reduction and the extraction of dominant features, thereby enhancing the classifier's ability to detect relevant intrusion patterns. Finally, three fully connected (dense) layers perform high-level feature abstraction. After that, there is a softmax output layer for a multi-class task.

System Details

The research utilizes the following concepts to develop the suggested FL-based IDS:

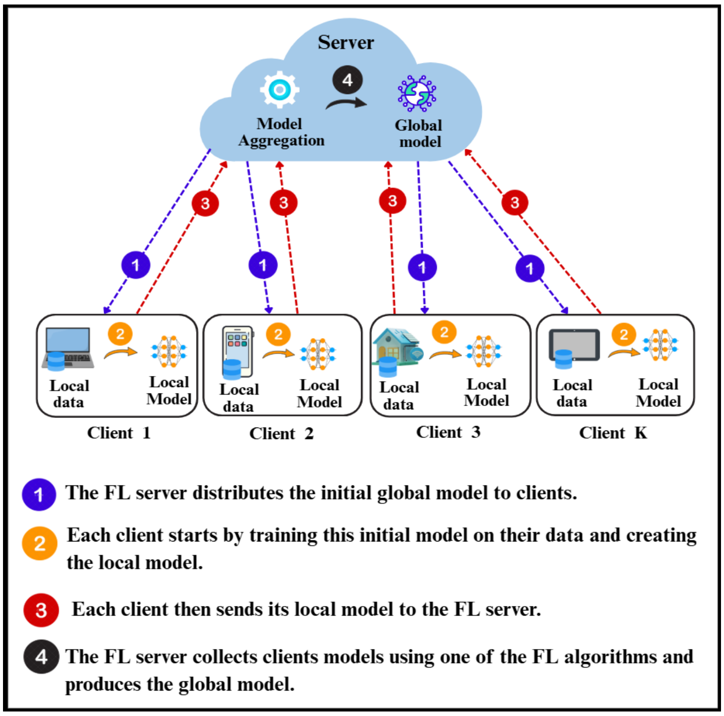
• **Participant clients (k):** Clients are individuals or institutions that participate in the process of building the global federated model. The study examines three valuable scenarios related to the number of participating clients: K = 10, K = 20, and K = 30, each with its respective datasets.

• **Classification model:** All participant clients utilize the same classifier, a DL-based one-dimensional CNN classifier. Before local training begins, each client will receive initial weights, after which the local training will start using their respective local datasets.

• **Dataset partition:** The study used the "Horizontal FL" method, in which each client's dataset is made up of different samples but shares the same feature space.(Tareq et al., 2024).

• **Data Distribution:** Federated architecture data distribution can either be IID (independent and identically distributed) or non-IID (non-independent and non-identically distributed). For the sake of simulating practical implementation, this research considers non-IID data distribution through imbalanced datasets. In non-IID data distribution, a subset of classes is owned by each client, whereas in IID data distribution, all classes are present for each client.

• **Aggregator Server:** The research employed a centralized FL framework, in which a central node known as the aggregator utilizes either the FedAvg or FedProx algorithm to generate global parameters. All participant clients present trainable parameters to the aggregator, which loops back to them until the global model converges. The aggregator is also responsible for sharing cyber threat knowledge globally while maintaining adherence to applicable laws and keeping each client informed about current trends in cyber threats. Fig. 2 depicts the architecture of the proposed FL-based IDS.



1 The FL server distributes the initial global model to clients.

2 Each client starts by training this initial model on their data and creating the local model.

3 Each client then sends its local model to the FL server.

4 The FL server collects clients models using one of the FL algorithms and produces the global model.

Fig. 2 a framework for the suggested FL-based IDS.

FL-Based IDS Framework

This section explains the steps and stages involved in building the proposed IDS model. Fig. 3 displays the flowchart of the FL-driven IDS. Following the preprocessing of the cybersecurity dataset, the SMOTEENN algorithm is applied to balance the class distribution and eliminate mislabeled traffic data. After that, split the data into a training and a test set. Next, apply the Dirichlet distribution to partition the training data in a non-IID manner, simulating heterogeneous client data. Each federated round commences with the distribution of the current global model's parameters to all participating clients. Subsequently, each client independently performs local training using a CNN classifier for ten epochs with its private dataset. Upon completion of local training, the updated model parameters from each client are transmitted back to the FL server. The FL server then executes the aggregation algorithm (FedAvg or FedProx), where the global model is updated by computing a weighted average of the client models' parameters. After the aggregation, the updated IDS global model is carefully tested on a central dataset to see how well it works using different metrics, such as accuracy, F1-score, precision, and recall. This iterative cycle of distributing the global model, conducting local training, and performing weighted aggregation continues for 50 rounds, culminating in a collaboratively trained federal model that leverages distributed data without direct data sharing.

Performance Metrics

The assessment of ML and DL models for classification tasks, like the one in this study, primarily depends on metrics derived from a confusion matrix (CM). This matrix is a cross-tabulation that indicates how frequently a model accurately classifies a data sample according to its true label. The model aims to identify the correct category of a data sample, and its prediction is then compared to the actual label. The CM helps evaluate the number of

instances where the model correctly or incorrectly classifies the data. Within the framework of IDS, a CM may be applied to check the rate at which a model manages to:
- True Positive (TP): Denotes the number of attack samples correctly classified as attacks.
- False Positive (FP): Indicates the quantity of benign samples mistakenly identified as attacks.
- True Negative (TN): Represents the number of benign samples accurately classified as benign.
- False Negative (FN): Denotes the quantity of attack samples erroneously categorized as benign.

A CM can be used to compute specific essential metrics. These represent:
- Accuracy: Represents the proportion of correct classifications out of the total number of samples, as shown in Equation (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (1)$$

- Precision: Refers to the proportion of correctly identified attack samples out of all predicted attacks, as displayed in Equation (2).

$$Precision = \frac{TP}{TP + FP} \qquad (2)$$

- Recall: recall quantifies the proportion of correctly detected attack samples out of all samples that should have been classified as attacks, as shown in Equation (3).

$$Precision = \frac{TP}{TP + FN} \qquad (3)$$

- F1-Score: Represents the harmonic mean of precision and recall, as shown in Equation (4).

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \qquad (4)$$

In multi-category classification, an averaging method is applied to calculate an overall score for each metric. Specifically, a weighted average is employed, which accounts for class imbalances based on the sample count of each class in the dataset.

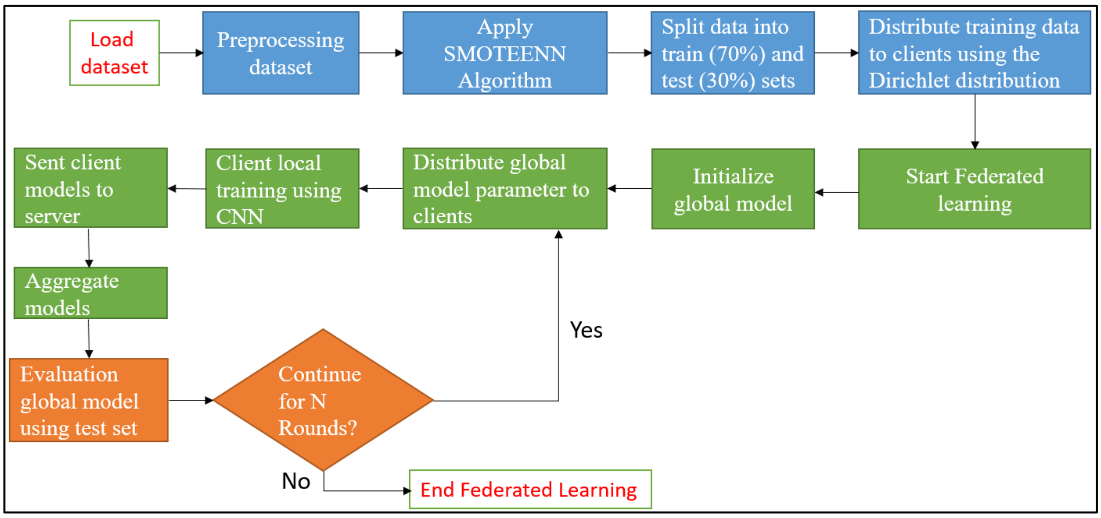

Load dataset
Preprocessing dataset
Apply SMOTEENN Algorithm
Split data into train (70%) and test (30%) sets
Distribute training data to clients using the Dirichlet distribution
Start Federated learning
Initialize global model
Distribute global model parameter to clients
Client local training using CNN
Sent client models to server
Aggregate models
Evaluation global model using test set
Continue for N Rounds?
Yes
No
End Federated Learning

Fig. 3 Flow chart of IDS based on FL

EXPERIMENT SETTING

All the experiments in this work are performed on the computing cluster with the hardware as an Intel Core i5-7440HQ CPU @ 2.80 GHz, 16GB RAM, and NVIDIA GeForce 940 MX, and the software as Python 3.12 and PyTorch. Docker containers manage the simulation environment, ensuring consistent performance across multiple runs. The hyper parameters are set as follows: the learning rate is 0.0001, the batch size is 128, the number of communication rounds is 50, and the number of local epoch training is 10. Additionally, there are three scenarios for the number of clients (k =10, k =20, and k =30).

RESULT

This section presents the actual outcomes from testing two FL algorithms, FedAvg and FedProx, using two different cybersecurity datasets: NSL-KDD and NF-Bot-net-v2. A non-IID data setting was created using a Dirichlet distribution with a concentration parameter (alpha) of 0.10 to divide the training data among the clients, showing real-world situations in FL where the client data distributions are quite different. Figs. 4 and 5 show the data distribution for clients.

The FedProx algorithm incorporated a proximal term with a coefficient (mu) of 0.01 to mitigate the impact of this statistical heterogeneity during local model training. Performance was evaluated across varying numbers of participating clients (K = 10, 20, and 30) using standard classification metrics: precision, recall, F1-score, and accuracy are detailed in the methodology section.

Performance on NSL-KDD Dataset

Table 2 summarizes the performance of the last round of the CNN-based FL models on the NSL-KDD dataset. On this dataset, both FedAvg and FedProx achieved exceptionally high performance across all metrics, with F1-scores and accuracy values consistently exceeding 98.7%. The FedProx algorithm, with a proximal term (μ=0.01), demonstrated a slight advantage in the 10- and 20-client scenarios, achieving an F1-score of 0.9949 and 0.9920, respectively. The study observed a marginal decrease in performance for both algorithms as the number of clients increased to 30. The finding suggests that with greater data fragmentation, the statistical heterogeneity across clients may introduce a slightly greater challenge for model convergence.

Performance on the NF-Bot-Net-V2 Dataset

To further validate this approach, the authors evaluated the models on the more complex NF-Bot-Net-V2 dataset. The results of the last round, detailed in Table 3, highlight the models' effectiveness in a different and more challenging network environment. The results on the NF-Bot-Net-V2 dataset show a more distinct performance pattern. The 10-client setup significantly impacted FedAvg's performance, resulting in an F1-score of 0.8895, which indicated significant struggles with the non-IID data distribution. However, its performance improved dramatically as the number of clients increased, reaching an F1-score of 0.9973 with 30 clients.

In contrast, FedProx demonstrated robust and superior performance across all client scales, achieving an F1-score of 0.9953 even with only 10 clients. This highlights the efficacy of the proximal term in mitigating the effects of statistical heterogeneity, ensuring stable and rapid convergence even when data is highly skewed. For all scenarios with 20 and 30 clients, both algorithms achieved near-perfect scores, indicating that a sufficient number of participating clients can help overcome the Non-IID challenge.

Table 2 summarizes the model performance results on the NSL-KDD Dataset.

Method	Clients	precision	Recall	F1-score	Accuracy
FedAvg	K=10	0.9917	0.9915	0.9915	0.9915
	K=20	0.9903	0.9903	0.9903	0.9903
	K=30	**0.9891**	**0.9889**	**0.9889**	**0.9889**
FedProx	**K=10**	**0.9949**	**0.9949**	**0.9949**	**0.9949**
	K=20	**0.9920**	**0.9920**	**0.9920**	**0.9920**
	K=30	0.9881	0.9877	0.9877	0.9877

Table 3 summarizes the model performance results on the NF-Bot-Net-V2 Dataset

Method	Clients	precision	Recall	F1-score	Accuracy
FedAvg	K=10	0.9284	0.8951	0.8895	0.8951
	K=20	0.9940	0.9940	0.9940	0.9940
	K=30	**0.9973**	**0.9973**	**0.9973**	**0.9973**
FedProx	**K=10**	**0.9953**	**0.9953**	**0.9953**	**0.9953**
	K=20	**0.9964**	**0.9964**	**0.9964**	**0.9964**
	K=30	0.9971	0.9971	0.9971	0.9971

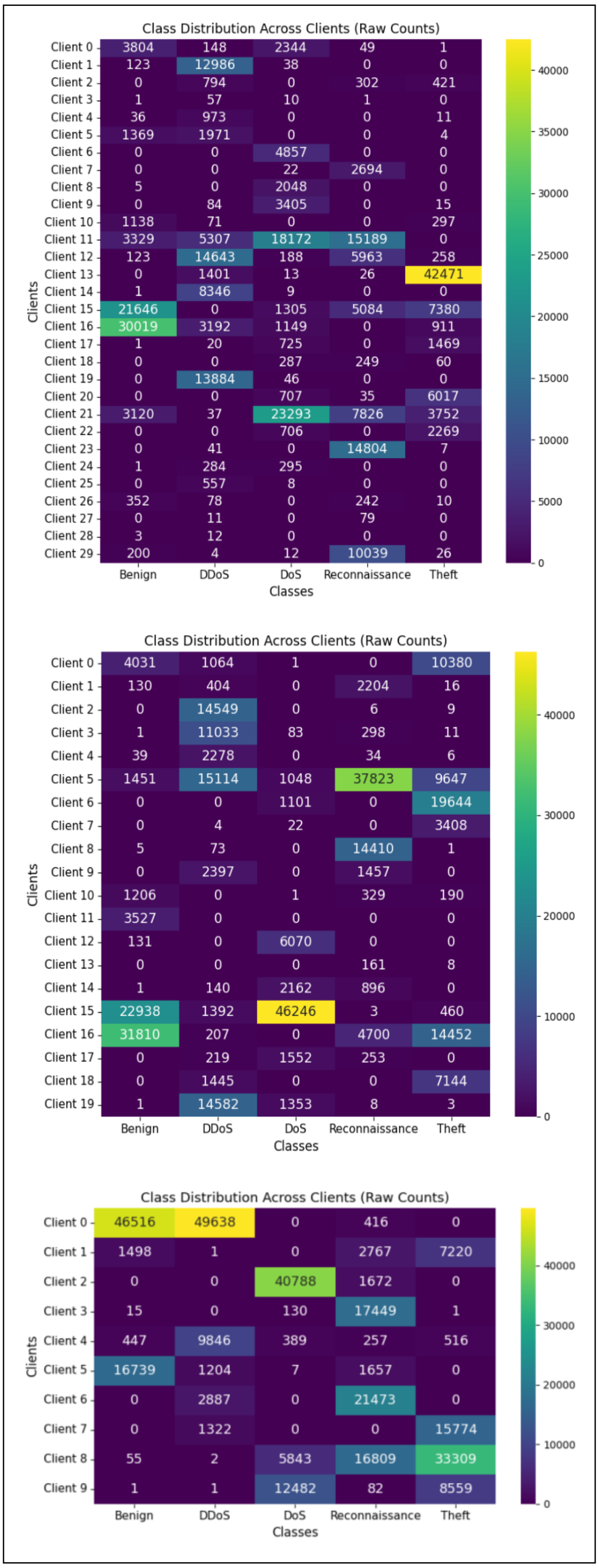


Fig.4: shows Client data distribution for the NF-Bot-Net-V2 dataset

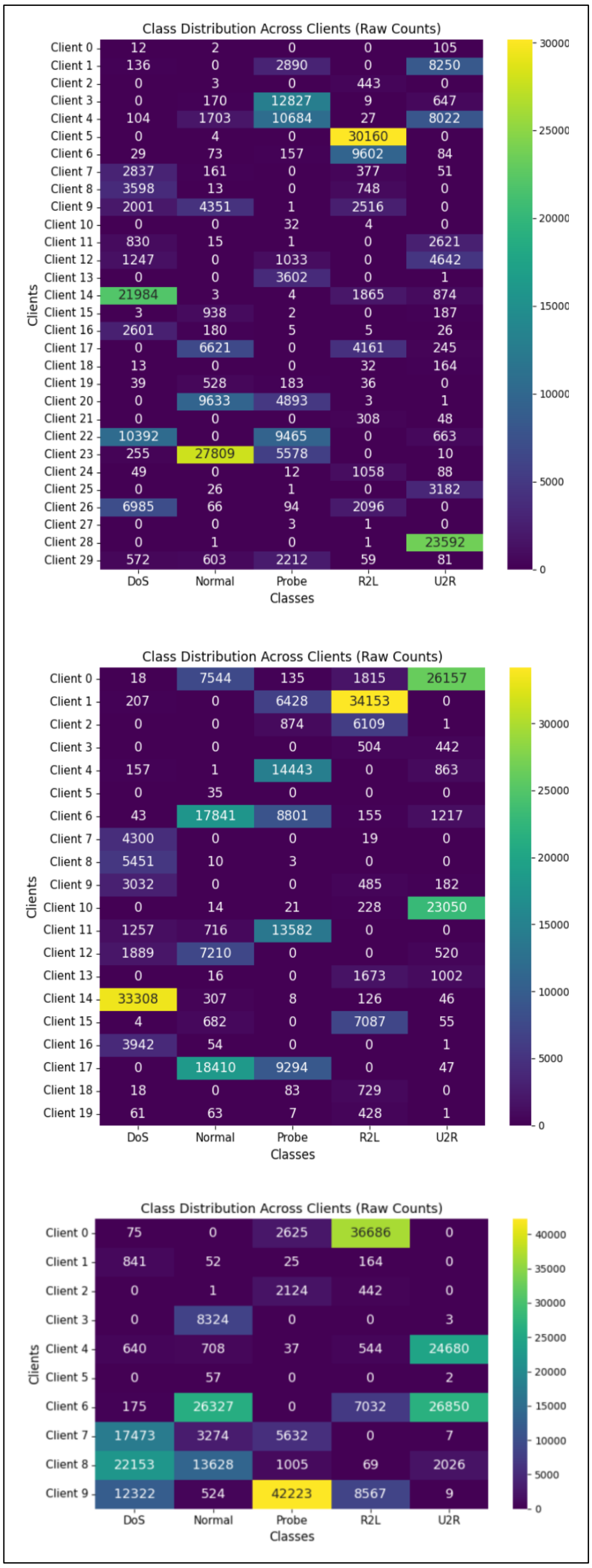


Fig.5: shows Client data distribution for the NSL-KDD dataset

DISCUSSIONS

The results presented in the previous section provide several key insights into the application of FL-IDS in Non-IID environments.

Impact of Statistical Heterogeneity

The findings clearly show that having different distributions of data (non-IID data) among clients is a major problem for FL-IDS. This was especially clear when we looked at how the standard FedAvg algorithm performed on the NF-Bot-Net-V2 dataset with only a few clients (K=10), where its performance dropped significantly. The skewed data distribution across clients causes the local models to drift toward their specific data biases, making it difficult for the global model to converge on an optimal solution for the overall data distribution.

The Efficacy of FedProx

The FedProx algorithm consistently outperformed or matched the performance of FedAvg. By introducing a proximal term to the local client objective function, FedProx effectively regularizes the local training process. This term discourages significant changes from the global model's parameters, which helps keep the local models from drifting too far and leads to a more stable and efficient improvement of the global model. The strength of FedProx, especially when there are fewer clients, makes it a better option for real-world FL situations where the number of devices and the types of data they have can vary a lot.

Implications for Real-World IDS

This study demonstrates the potential for FL to build collaborative and privacy-preserving intrusion detection systems. The high accuracy and F1 scores achieved, particularly with FedProx, indicate that this approach is viable for effectively identifying network attacks without centralizing sensitive data. The use of a robust 1D-CNN architecture proves effective for learning from tabular network traffic data. For practical deployment, FedProx is the recommended algorithm due to its resilience against statistical heterogeneity, a common characteristic of real-world network environments.

CONCLUSION

This paper explores the implementation of FL-powered IDS, focusing on addressing the challenges posed by non-IID data distributions in real-world settings. The study developed and evaluated CNN-based federated models by contrasting the efficacy of the standard FedAvg and FedProx algorithms across different client counts (10, 20, and 30) utilizing two IDS datasets: NSL-KDD and NF-Bot-Net-V2. The models that used the FedAvg algorithm were affected by data heterogeneity, as shown by a big decline in performance with fewer clients (for example, an F1-score of 0.8895 with 10 clients on NF-Bot-Net-V2). The experiment demonstrates that when data distributions are highly unbalanced, local models become susceptible to drift. Conversely, FedProx consistently demonstrated greater stability and robustness across all tested configurations. Its proximal term successfully standardized local updates, reducing the negative effects of statistical heterogeneity and encouraging more stable global model convergence. Because it is powerful, FedProx is a favorable choice for real-world use in FL-IDS systems in heterogeneous network environments where data is constantly changing and is extremely difficult to predict. This study did not evaluate adversarial threats or analyze communication efficiency under bandwidth constraints, which may affect system performance in practical implementations. Future research endeavors may rectify these deficiencies by incorporating secure aggregation protocols to bolster privacy preservation and formulating adaptive client selection algorithms to improve training efficiency and model accuracy.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the University of Mosul for the technical support.

REFERENCES

Abdulganiyu, O. H., Ait Tchakoucht, T., & Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). *International Journal of Information Security*, *22*(5), 1125–1162. https://doi.org/10.1007/s10207-023-00682-2

Ahmed, I., Ali, A. K., & Mahmood, M. S. (2025). Employing Hybrid Watermarking to Improve Email Security Against Cyber Attacks. *Journal of Soft Computing and Data Mining*, *6*(1), 435–447. https://doi.org/10.30880/jscdm.2025.06.01.029

Al-Dabbagh, M., & Ali, A. K. (2022). Employing light fidelity technology in health monitoring system. *Indonesian Journal of Electrical Engineering and Computer Science*, *26*(2), 989. https://doi.org/10.11591/ijeecs.v26.i2.pp989-997

Alazab, A., Khraisat, A., Singh, S., & Jan, T. (2023). Enhancing Privacy-Preserving Intrusion Detection through

Federated Learning. *Electronics*, *12*(16), 3382. https://doi.org/10.3390/electronics12163382

Aljanabi, M., Ismail, M. A., & Ali, A. H. (2021). Intrusion Detection Systems, Issues, Challenges, and Needs. *International Journal of Computational Intelligence Systems*, *14*(1), 560. https://doi.org/10.2991/ijcis.d.210105.001

Brendan McMahan , Eider Moore , Daniel Ramage , Seth Hampson, B. A., & Arcas. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics.*, 1273–1282.

Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, *4*(6), 446–452. https://doi.org/10.17148/IJARCCE.2015.4696

Fedorchenko, E., Novikova, E., & Shulepov, A. (2022). Comparative Review of the Intrusion Detection Systems Based on Federated Learning: Advantages and Open Challenges. *Algorithms*, *15*(7), 247. https://doi.org/10.3390/a15070247

Hernandez-Ramos, J., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., Marmol, E., Gonzalez-Vidal, A., & Kambourakis, G. (2025). Intrusion Detection Based on Federated Learning: A Systematic Review. *ACM Computing Surveys*. https://doi.org/10.1145/3731596

Hsu, T.-M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *ArXiv Preprint ArXiv:1909.06335*.

Lavaur, L., Pahl, M.-O., Busnel, Y., & Autrel, F. (2022). The Evolution of Federated Learning-Based Intrusion Detection and Mitigation: A Survey. *IEEE Transactions on Network and Service Management*, *19*(3), 2309–2332. https://doi.org/10.1109/TNSM.2022.3177512

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, *2*, 429–450.

Liu, Y., Wu, G., Zhang, W., & Li, J. (2023). Federated Learning-Based Intrusion Detection on Non-IID Data. In *International Conference on Algorithms and Architectures for Parallel Processing* (pp. 313–329). Springer. https://doi.org/10.1007/978-3-031-22677-9_17

Popoola, S. I., Gui, G., Adebisi, B., Hammoudeh, M., & Gacanin, H. (2021). Federated Deep Learning for Collaborative Intrusion Detection in Heterogeneous Networks. *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 1–6. https://doi.org/10.1109/VTC2021-Fall52928.2021.9625505

Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mobile Networks and Applications*, *27*(1), 357–370. https://doi.org/10.1007/s11036-021-01843-0

Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., & Guizani, M. (2022). FEDGAN-IDS: Privacy-preserving IDS using GAN and Federated Learning. *Computer Communications*, *192*, 299–310. https://doi.org/10.1016/j.comcom.2022.06.015

Tareq, I., Elbagoury, B. M., El-Regaily, S., & El-Horbaty, E. S. M. (2024). A survey about deep learning and federated Learning in cyberse-curity. *Periodicals of Engineering and Natural Sciences*, *12*(1), 75–100. https://doi.org/10.21533/pen.v12i1.3963.g1355

Wang, H., Muñoz-González, L., Eklund, D., & Raza, S. (2021). Non-IID data re-balancing at IoT edge with peer-to-peer federated learning for anomaly detection. *WiSec 2021 - Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 153–163. https://doi.org/10.1145/3448300.3467827

Weinger, B., Kim, J., Sim, A., Nakashima, M., Moustafa, N., & Wu, K. J. (2022). Enhancing IoT anomaly detection performance for federated learning. *Digital Communications and Networks*, *8*(3), 314–323. https://doi.org/10.1016/j.dcan.2022.02.007

Yang, F., Wang, K., Sun, L., Zhai, M., Song, J., & Wang, H. (2022). A hybrid sampling algorithm combining synthetic minority over-sampling technique and edited nearest neighbor for missed abortion diagnosis. *BMC Medical Informatics and Decision Making*, *22*(1), 1–14. https://doi.org/10.1186/s12911-022-02075-2

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. *ArXiv Preprint ArXiv:1806.00582*. https://doi.org/10.48550/arXiv.1806.00582