Ingénierie des Systèmes d'Information
Vol. 31, No. 2, February, 2026, pp. 625-637
Journal homepage: http://iieta.org/journals/isi

Federated Deep Learning for Intrusion Detection in Internet of Things Networks Using Software-Defined Networking

Rand Nabeel Dawood

Al-Turath University, Computer engineering technology department, Baghdad 10071, Iraq

Corresponding Author Email: rand.nabeel@uoturath.edu.gen

https://doi.org/10.18280/isi.310229

Received: 10 September 2025
Revised: 20 November 2025
Accepted: 17 February 2026
Available online: 28 February 2026

Keywords:
Intrusion Detection System, Software-Defined Networking, federated learning, IoT security, Anomaly Detection, Deep Learning

ABSTRACT

Traditional Intrusion Detection Systems (IDS) frequently depend on centralized data aggregation, encountering substantial obstacles due to data privacy rules and escalating network traffic. Federated Learning (FL) provides a decentralized, privacy-conscious framework for collaborative model training across distributed devices, addressing these challenges by preserving data localization and minimizing bandwidth and storage requirements. Due to resource limitations and various threats in Internet of Things (IoT) contexts, traditional Intrusion Detection Systems (IDSs) frequently prove insufficient. This paper presents a federated deep learning framework for intrusion detection in Internet of Things (IoT) networks, leveraging Software-Defined Networking (SDN) to enhance privacy and security. The model combines federated learning with a deep neural network (DNN) to address the challenges of centralized Intrusion Detection Systems (IDS), such as data privacy concerns and high bandwidth usage. The proposed method is evaluated using the UNSW-NB15 dataset, achieving an accuracy of 87%, outperforming the baseline model with 86% accuracy. The federated approach allows for data privacy preservation, while improving generalization across various attack types. The results demonstrate that the federated model, in conjunction with SDN, can provide an efficient and scalable solution for IoT security. This study contributes to the growing body of research on federated learning and SDN integration for secure IoT networks, offering practical insights for deploying real-time intrusion detection systems..

1. INTRODUCTION

The Intrusion Detection Systems (IDS), commonly known by their acronym, are the initial defense line against malicious incursions in the cybersecurity domain. Such systems utilize advanced analytical tools to identify and report abnormal behavior that exists in network traffic [1]. Recent progress in the field of Machine Learning (ML) and Deep Learning (DL) has significantly increased the effectiveness of the IDS, which makes it possible to implement them in an ever-growing range of application areas to respond to the emergence of threats. However, traditional ML-based and DL-based IDS systems are characterized by the reliance on centralized servers to accumulate and analyse network traffic. The main drawbacks of such a centralized paradigm are tougher data-privacy laws that limit central accumulation of raw information and the increasing amount of network traffic that overloads current infrastructure. As an alternative, Federated Learning (FL) [2] is a privacy-preserving, decentralized model that allows a localisation [4] of data and allows collaborative training of a model across distributed devices. The paradigm has been developed to deal with the above challenges, and when compared to centralized solutions, the benefits are realized in terms of protecting the privacy of users by eliminating raw data sharing and minimizing operational expenses due to the reduction of bandwidth usage and storage overheads [5].

The Internet of Things (IoT) is a multi-purpose network that connects a large number of resource-restricted devices. As a result, a massive amount of data is produced, which creates a new system of security vulnerabilities [6]. The traditional IDS is often inapplicable to the IoT setting because it has inherent hardware constraints in terms of its computational and memory capabilities. Additionally, the methods of the adversary are getting more advanced. Therefore, there is a dire need to come up with a thoroughly effective IDS that would be able to protect against a wide variety of IoT threats. In the past few years, researchers have discussed numerous ways of enhancing the security of IoT [7, 8]. These, however, have not been successful in overcoming the wide threat terrain, hence the need for a centralized architecture. Software-Defined Networking (SDN) provides a revolutionary response to the requirements of programmable network management and proactive threat detection by real-time traffic monitoring in large heterogeneous IoT networks. SDN gives unmatched flexibility by separating the control plane and the data plane. Its inherent properties make it particularly suitable to deploy

IDS that are able to scale to defend large-scale Internet of Things deployments without compromising the low-latency and high-efficiency operation required by devices with limited resources [9].

In this work, a federated deep learning model has been developed for different clients with a global model within an SDN environment for an IoT system to detect different cyberattacks. The proposed system utilized the UNSW-NB15 dataset to assess the IDS system accuracy. Loss, precision, and recall metrics have been used to measure the training performance of the federated deep neural network system.

The rest of sections is organized as follows: in Section 2 a description of related work. Section 3 provides the theortical background used in this work, while section 4 demonstrates the proposed IDS design. Section 5 presents the results and their analysis. Finally, section 6 provides the paper's conclusion and shows some suggestions for future work.

2. RELATED WORK

The exponential growth of IoT devices has resulted in an impressive growth of network vulnerabilities, creating a need for strong IDS. While centralized machine learning approaches have shown promise, they are facing critical issues with respect to data privacy, as well as high transmission costs. FL has become a decentralized alternative for enabling multiple clients to jointly train a global model without having to share their local raw data. Recent literature is dedicated to increasing the detection accuracy, dealing with the heterogeneity of the data, and securing FL against adversarial attacks.

Nguyen et al. [10] showed that FL-IDS are prone to backdoor poisoning attacks where compromised IoT devices (not only gateways) may corrupt the global model. The paper is about the weaknesses and successes of the attack instead of giving a final universal defense mechanism. That's why Rashid et al. [11] created a framework for Industrial IoT (IIoT) with a balanced dataset approach and also compared various algorithms, such as GANs and GRUs, for better discovery accuracy. Specific limitations noted for some referenced medical IDS models include the need to install numerous systems through a network. Lazzarini et al. [12] evaluated several FL aggregation algorithms (FedAvg, FedAvgM). In their particular IoT scenario, they found that adaptive algorithms like FedAdam and FedAdagrad were worse than simpler algorithms (FedAvg). Alazab et al. [13] presented their study to improve privacy-preserving IDS by processing large amounts of sensitive data using decentralized local model training. The approach needs substantial local processing capacity back at every client location to manage the "large volumes of sensitive data". Fed-ANIDs proposed by Idrissi et al. [14] treat non-IID data and privacy through autoencoders (AE, VAE, AAE), and the FedProx algorithm. Adversarial Autoencoders (AAE) exhibited much smaller F1-scores (~45%) as compared to standard VAEs (~90%) in their tests. Thein et al. [15] conducted a PFL-IDS, an individualized FL strategy that applies one-shot mini-batch education coupled with robust aggregation to protect against the hazard of poisoning attacks on disparate audiovisual information of the Internet of Things. The results of the study demonstrate that even in the presence of defenses, the introduction of a malicious client can degrade performance compared to working in a perfectly benign environment. Jin et al. [16] proposed FL-IIDS, an extended learning system, which is supported by a tailored loss function and knowledge distillation to avoid catastrophic forgetting of attack types that have already been trained. The effectiveness of the framework is highly reliant on the functionality of the "relay client" and sample reconstruction, which could potentially add complexity to the global aggregation process. Devine et al. [17] proposed a federated Support Vector Machine (SVM) model optimized to serve as a low-resource model on the IoT edge device with emphasis on data processing without the use of synthetic data. Although it is efficient, the paper does not ignore the trade-offs between the high detection accuracy and the computational limitations of IoT devices.

Table 1 shows a comparative analysis of recent research papers focused on FL for IDS, specifically within IoT and IIoT environments.

Table 1. Related work comparative analysis

Ref.	Framework	Algorithm	Datasets	Contribution
Rashid	FL-based IDS for IIoT	Balanced Dataset approach	Not specified in snippet	Focused on improving IIoT security via balanced data
Devine	Federated SVM	SVM / Federated Framework	Real-world non-synthetic	First-of-its-kind federated SVM for edge device efficiency[18].
Jin	**FL-IIDS**	Incremental Learning / Knowledge Distillation	UNSW-NB15, CICIDS2018	Addresses catastrophic forgetting in dynamic environments
Thein	**PFL-IDS**	Personalized FL	Not specified in snippet	Combats poisoning attacks on heterogeneous data
Lazzarini	Collaborative FL	Shallow ANN / FedAvg, FedAdam	ToN_IoT, CICIDS2017	Comparison of different FL aggregation algorithms
Idrissi	**Fed-ANIDS**	Autoencoders (AE, VAE, AAE) / FedProx	USTC-TFC2016, CIC-IDS2017, CSE-CIC-IDS2018	High performance in anomaly-based detection with privacy[26].
Nguyen	Poisoning Attack Study	Data Poisoning / Backdoor Attack	3 Real-world IoT datasets	Demonstrated vulnerabilities of FL-IDS to stealthy backdoor attacks

3. METHODOLOGY

Attacks are one of the things that need to be thought about because they can have a big effect on both the network's infrastructure and the people who use it. So, figuring out what kind of attacks are happening and what they are is very important for keeping the network safe and avoiding any problems or dangers that could make it less safe. So, the suggested model is mostly about using DL-based methods to find different types of assaults that can happen on the IoT network.

3.1 Intrusion Detection System

Intrusion detections are typically categorised into two types: signature detection and anomaly detection. The initial approach, known as "misuse intrusion detection," depends on established network patterns and is hence incapable of recognising novel attacks. Anomaly detection, frequently referred to as "behaviour-based IDS," involves constructing a model of typical or normal behaviour to autonomously recognise any departures from it [19]. The principal benefit of anomaly detection systems is their capacity to recognise unforeseen hazards. Nonetheless, they may detect novel or atypical non-intrusive network activity pertinent to a network administrator, which could potentially elevate the false alarm rate [20].

Table 2. Intrusion detection systems types comparative [22]

Type	Detection Scope	Input Data	Key Advantages	Major Limitations
HIDS	Host	Host processes, logs, and configuration.	- Analyzes encrypted data. - Indicate if the attack was successful. - No extra hardware needed.	- Fails if OS compromised. - Can't identify network attack. - Resource-intensive.
NIDS	Network	Network packets, user profiles, preceding events.	- Host-independent operation. - Detects network-layer attacks.	- Can't verify if the attack was successful. - Unable to analyze encrypted traffic. - Limited host visibility.
Hybrid	Host + Network	According to the systems.	- Flexible and efficient.	- Resource-intensive. - Extensive duration to analyze data.

IDS are classified into two principal categories: Network-based IDS (NIDS) and Host-based IDS (HIDS). A Network Intrusion Detection System (NIDS) is implemented at key locations within the network infrastructure to scrutinise and assess traffic for harmful patterns or anomalies. Conversely, a HIDS functions on discrete endpoints to analyse system logs, file integrity, and process behaviour for indications of compromise. Although hybrid IDS architectures, which integrate NIDS and HIDS, have been developed to capitalise on the advantages of both methodologies, their implementation and validation in research are still constrained. Table 2 presents an examination of different types of IDSs [22].

3.2 Software-defined network

The software defiend network architecture consists of three layers that enable centralised, reprogrammable network management and dynamic traffic control. The SDN architecture consists of the following:

• Infrastructure layer: consists of SDN virtual switches, such as Open Virtual Switch (OVS) [23], that relay traffic to the control plane without examination.

• Control plane: consists of an SDN controller that control switches by establishing forwarding rules and maintaining an extensive network overview. Its essential functions include topology identification and dynamic flow configuration.

Application plane: accommodates network management applications that configure the SDN controller to implement dynamic rules, including firewalling and network address translation (NAT) across switches, therefore abstracting policy logic from the underlying infrastructure.

3.3 Federated learning

Figure 1 illustrates the FL training process. In this framework, several edge devices function as autonomous clients connected to a central server that has a global model. A set of clients who have indicated that they are prepared to take part in the training procedure forms the basis for the selection of the number of FL clients (D), which is then picked at random. At the moment *t*, every client that has been selected (d) downloads the global model weight parameters (θ_t) from the server. Then, using its own dataset to train the local model [24].

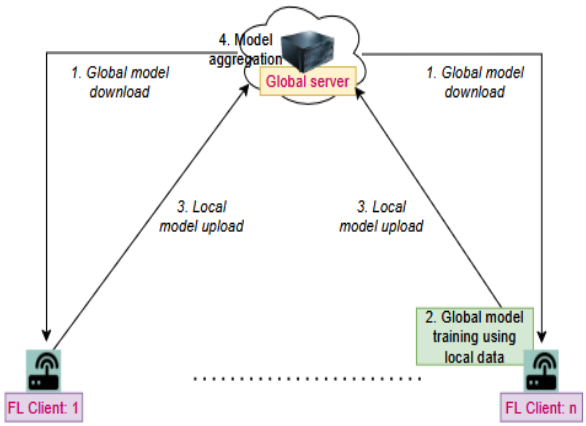


Figure 1. Traditional federated learning [24]

$$\min f(\theta)$$

$$f(\theta) = \sum_{d=1}^{D} \frac{m_d}{M} F_{d\,(\theta)} \tag{1}$$

$$F_d(\theta) = \frac{1}{m_d} \sum_{i \in m_d} f^i(\theta)$$

During the local training, every client strives to optimise the global model by reducing the value of its loss function $f^i(\theta)$, which is associated with the i_{th} sample in its dataset. For this, we use optimization methods like Adam or stochastic gradient descent (SGD). When the client's local training is finished, it sends the global server its updated model, which is represented as θ_{t+1}^d.

$$\theta_{i+1}^d = \theta_i - \alpha_d \lambda_d \tag{2}$$

In this case, α_d stands for the learning rate and λ_d for the

gradient that client d computes with its local dataset and the parameters θ_t . The following is how the global server computes the upgraded global model θ_{t+1} by combining the local models that have been collected.

$$\theta_{i+1} = \sum_{d=1}^{D} \frac{m_d}{M} \theta_{i+1}^{d} \tag{3}$$

Until the model converges, FL iteratively downloads the trained global model for the following rounds. Multiple users work together to train the global model, where the FedAvg aggregation method is used to obtain the global model weight parameters in each round. The centralized learning method, while maintaining the privacy of their raw data. Instead of sending the raw data to a central server, each client trains locally using its dataset and then provides just the changed model parameters. To compile all of these changes, the server takes an average of the model weights, where the weights are based on the size of the client's dataset. Improved privacy and security, less bandwidth utilization, and scalability across many clients are all benefits of this method. But it has problems with communication efficiency and data heterogeneity. The FedAvg formula is:

$$w_{r+1} = \sum_{c=1}^{C} \frac{n_c}{n} w_{r+1}^{c} \tag{4}$$

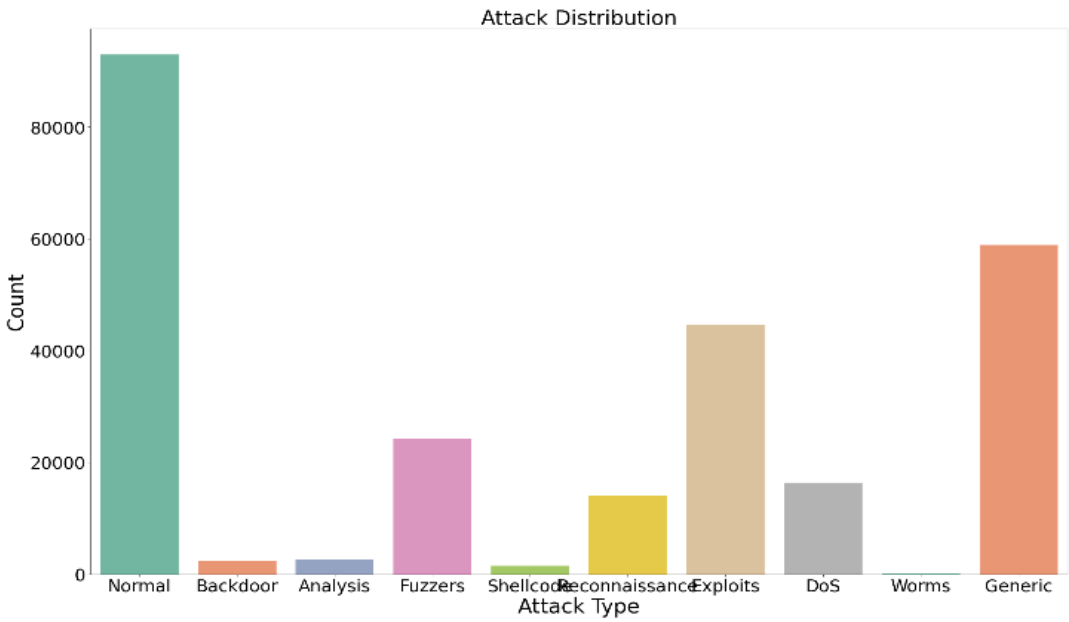
where n_k is the overall number of samples collected by all clients in round $r+1$, and w_{r+1}^{c} is the weight matrix of the local model that client c updated. As shown in Algorithm 1, which shows the global and local model update and the FedAvg aggregation method. The loss function η and a gradient $\nabla L(w_r^c;b)$ are calculated according to the model's weights w_t^k and input data b.

Algorithm 1: FL (FedAvg)

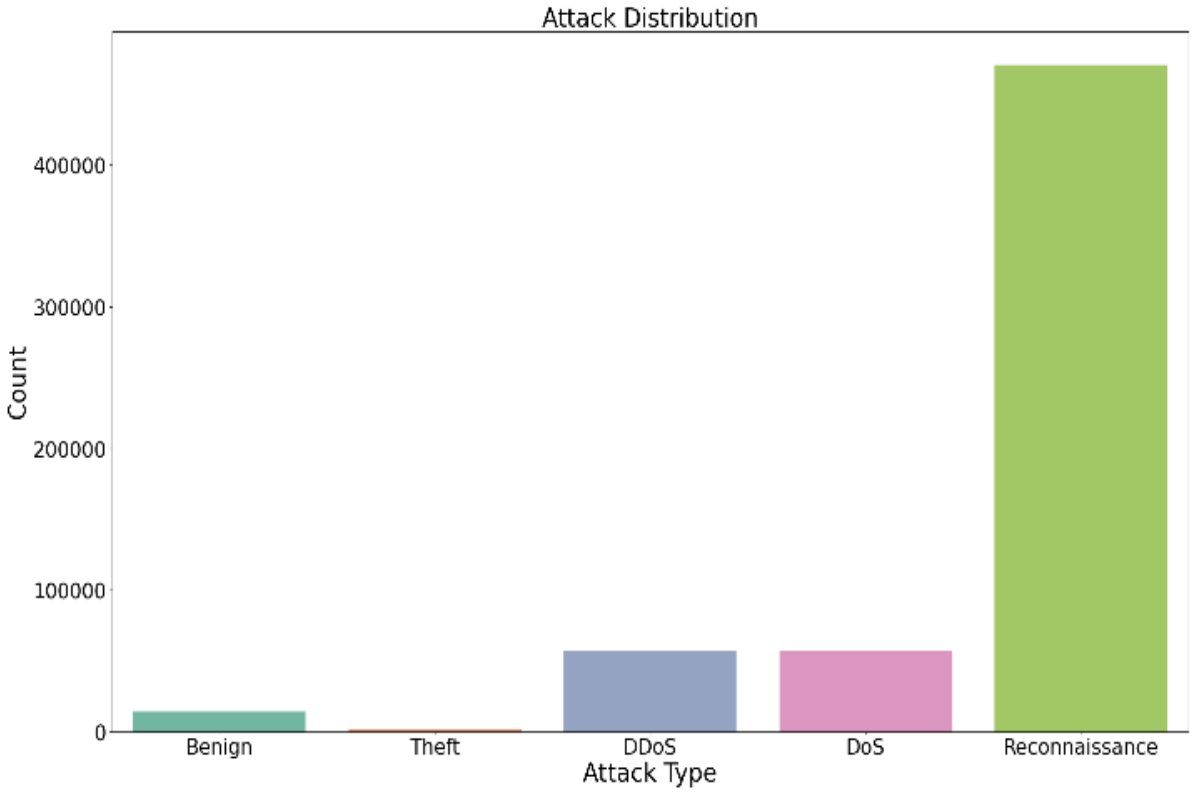
Step 1:Initial global model parameters= m, clients number = c,
training rounds= r
step2:
for round r = 1, 2 ..., r
for client = 1,2, ..., c
batches = divide the total dataset D for each client into size b
$m_r^c \leftarrow m_r$, initialize each model client and global
for each batch b $\in$ D :
$w_{r+1} \leftarrow \frac{1}{K} \sum_{c=1}^{c} w_{r+1}^{c}$;
end
end
step3:Updated global model m_t
$m_{r+1} \leftarrow \frac{1}{c} \sum_{c=1}^{c} w_{r+1}^{c}$;
end for

3.4 Dataset benchmark

In this work, two types of public datasets have been utilized to evaluate the proposed FL framework IDS, which are UNSW-NB15 and Bot-IoT. The nine attack types included in the UNSW-NB15 dataset, which aims to simulate modem network traffic, include fuzzers, worms, and reconnaissance, among others. For all-encompassing analysis, it combines basic, content-based, and time-based attributes with 40 features. Produced using the IXIA PerfectStorm program, it presents class imbalance scenarios that are both realistic and challenging. Researchers interested in network intrusion detection can use this dataset for both supervised and unsupervised learning. Modern intrusion detection systems rely on it for testing purposes. Here is the attack distribution for this dataset, as seen in Figure 2(a). The CSE-CIC-IDS2018, capturing 10 days of normal and malicious network traffic. Its features include attacks like DDoS, brute force, and web infiltration, with labeled network flows for anomaly detection. The dataset provides a wide range of features, including flow duration and protocol type, for developing intrusion detection models. It simulates realistic network environments with rich diversity. Researchers commonly use it for benchmarking flow-based IDS models. Figure 2(b) shows the attack distribution for this dataset.



(a)



(b)

Figure 2. Attack Classes Distribution a) UNSW-NB15
b) CSE-CIC-IDS2018

4. PROPOSED SYSTEM DESIGN

The specified models were run on TensorFlow, Keras, and Scikit-learn to assess and determine the model that yields optimal results. Several experiments were performed to determine the best model hyperparameters. The purpose of including batch normalization and dropout layers is to enhance the networks performance and their generalization capability. Activation function used in the dense layers was the Rectified Linear Unit (ReLU) because it is simple, efficient and can address the vanishing gradient issue. Moreover, the output layer applies the softmax activation function, which is suitable with the categorical classes. The model has used the Adam optimizer to prevent the rapid decrease in the learning rate with a batch size of 64 in 50 epochs.

The proposed system will include four steps, which are dataset preparation, dataset training, model optimization techniques, and model testing and evaluation. The brief description of each of the stages is presented below:

a) Dataset Preprocessing: Dataset preprocessing has the following sub-stages:

i) Cleaning: Removal of undesired numbers (including NaN) and other anomalies that might occur during the implementation of datasets (as provided by some tools e.g., the CIC flowmeter) should be removed. Also, some features like the source and destination IP addresses ought to be eliminated.

ii) One-hot encoding: This is used to encode categorical labels in a numerical format that can be effectively used by deep learning models to identify meaningful patterns and relationships between multiple classes where a one indicates the existence of a class and a zero indicates the absence of a class.

iii) Normalization: All datasets should be normalized between -1 and 1. It is an essential preprocessing step, due to several reasons: it increases stability, convergence speed, performance, and reduces bias.

iv) Dataset Splitting: This crucial method presupposes dividing the data into training (60%), validation (10%), and testing (30) parts.

b) Table 3 specifies the hyperparameters of the proposed Deep Neural Network (DNN) model. The model has six levels, where the input layer is the feature dimensions, and the output layer is five types of faults [25]. The ReLU activation function of the hidden layers also enables the network to learn more complex patterns by introducing non-linearity. The output layer makes use of the SoftMax function, which is commonly used in multi-class classification problems, to convert the output into a probability distribution. The loss used is categorical cross-entropy that is suitable in multi-class classification. Adam optimizer is used to optimize the weights of the model during the training stage, which is known to be efficient and effective [25]. The DNN model layers in this work are chosen based on number of input features influence the decision taken. The matching of the model capacity to the complexity of the problem enables the network to find abstract patterns without extraneous details. This architecture preserves computational efficiency while minimizing the risk of overfitting, especially in scenarios with limited data. It is compatible with the scientific experiments and principles of architecture. Generally, eight layers of 50 neurons would be effective in various datasets and classifications. is categorical

cross-entropy loss function has been used in this work, which is the suitable when working with multiclass labels.

The controller coordinates the federated learning process and updates and maintains the global model. The training cycle involves five involved clients who start with the controller applying the global model to the clients, who then carry out local training on their own data partitions. The clients are interacting via their respective IP addresses to ensure a secure and verifiable interaction. The clients update their models after every training round and send the models back to the controller and apply the FedAvg algorithm. The FedAvg works by averaging the weights produced by federated learning clients at every training step. Figure 3 shows the overall federated learning workflow, including model distribution and encrypted parameter aggregation, but with a special focus on the collaborative training process and the high security measures that are guaranteed at each step.

Table 3. Deep neural model hyperparameter

Hyperparameter	Value
Neuron per layer	50
Number of layers	6
No. of epochs	20
No. of rounds	20
Activation function	ReLU(hidden layers), SoftMax(output layer)
Loss function	Categorical-cross-entropy
Optimizer	Adam
Attacks Classes	10
Batch-size	64
Use L2 Regularization	True
L2 Weight	0.005
Dropout Rate	0.4
Number of Clients	5

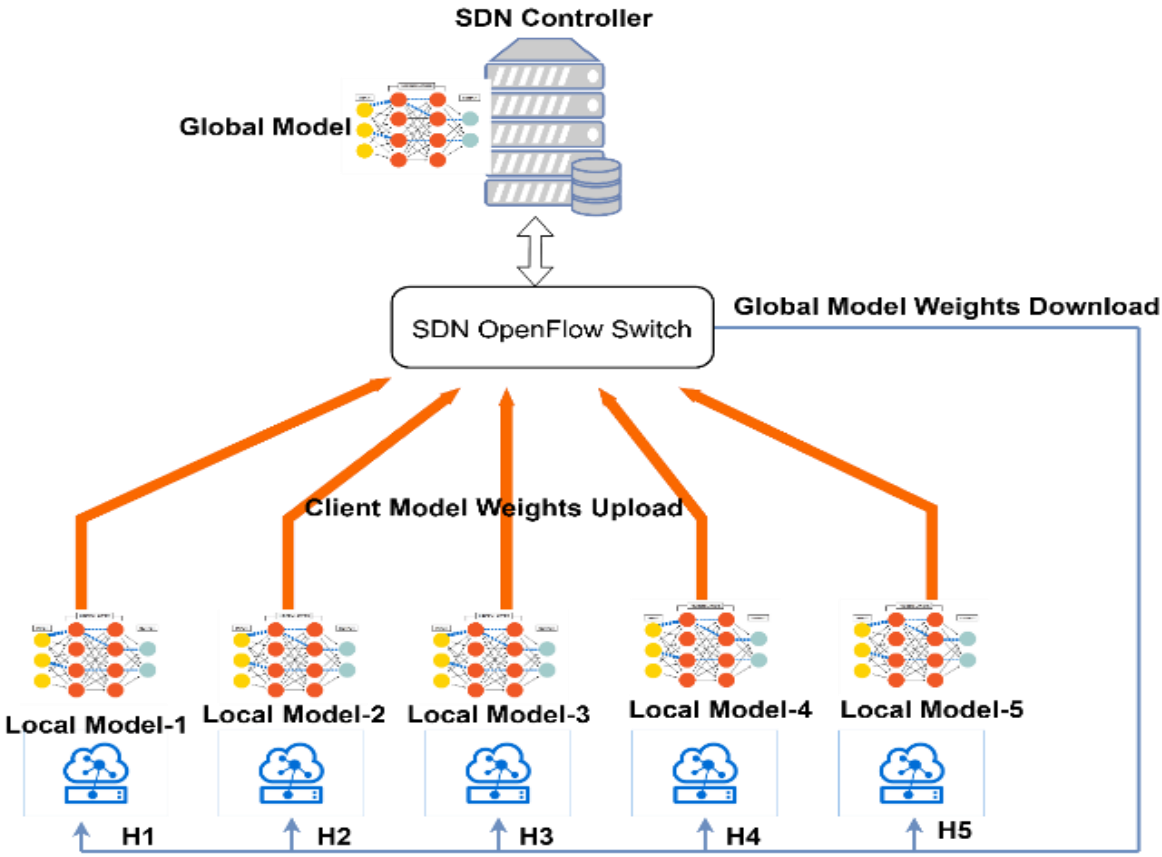


Figure 3. Proposed system design

5. RESULTS AND DISCUSSION

Figures 4 and 5 show the training and validation curves; the comparison is based on the different behavior of the following datasets in the training of the DNN, the accuracy, loss, precision, and recall of the UNSW-NB15 and Bot-IoT datasets, respectively. The convergence rate of Bot -IoT is often rapid. Due to the extremely specialized nature of Bot-IoT on IoT-specific attacks, including those of DDoS and DoS, which have highly distinctive, repetitive characteristics, the DNN can acquire the ability to differentiate normal and malicious traffic extremely rapidly. Conversely, the convergence in UNSW-NB15 is normally more gradual. The dataset includes a larger range of weak fuzzing and backdoors that are more modern attacks, and which look more like regular traffic. The loss curve of UNSW-NB15, therefore, tends to oscillate initially and level off as compared to Bot-IoT.

Bot-IoT can then note that the difference between the training and validation accuracy is very low. This is due to the fact that the dataset is huge and its trends are very uniform. The specificity of the data, however, presents an opportunity that threatens that instead of learning generalized behavioral knowledge, the model is basically learning botnet signatures. UNSW-NB15, in turn, has a slightly broader gap between the training and validation metrics, showing that the model is more challenging to extrapolate the training information to the unseen validation information, as the categories of attacks are too complex and diverse.

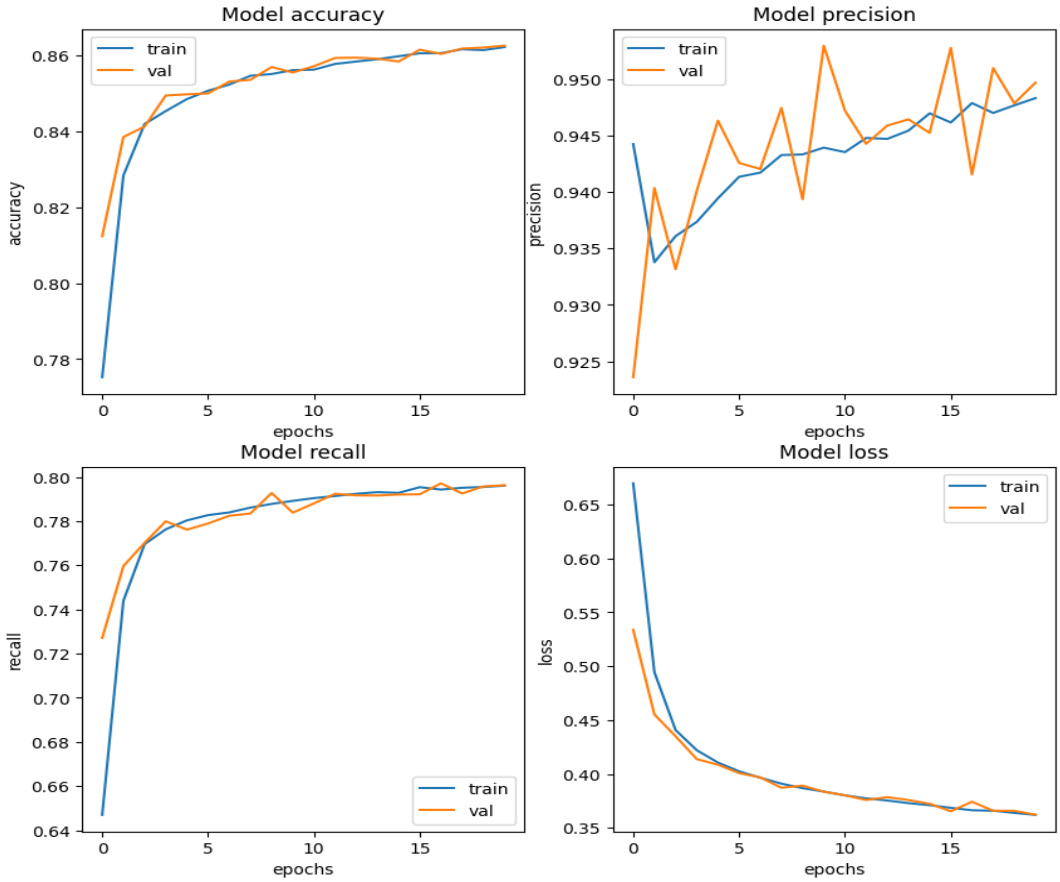


Figure 4. UNSW-NB15 baseline model training performance

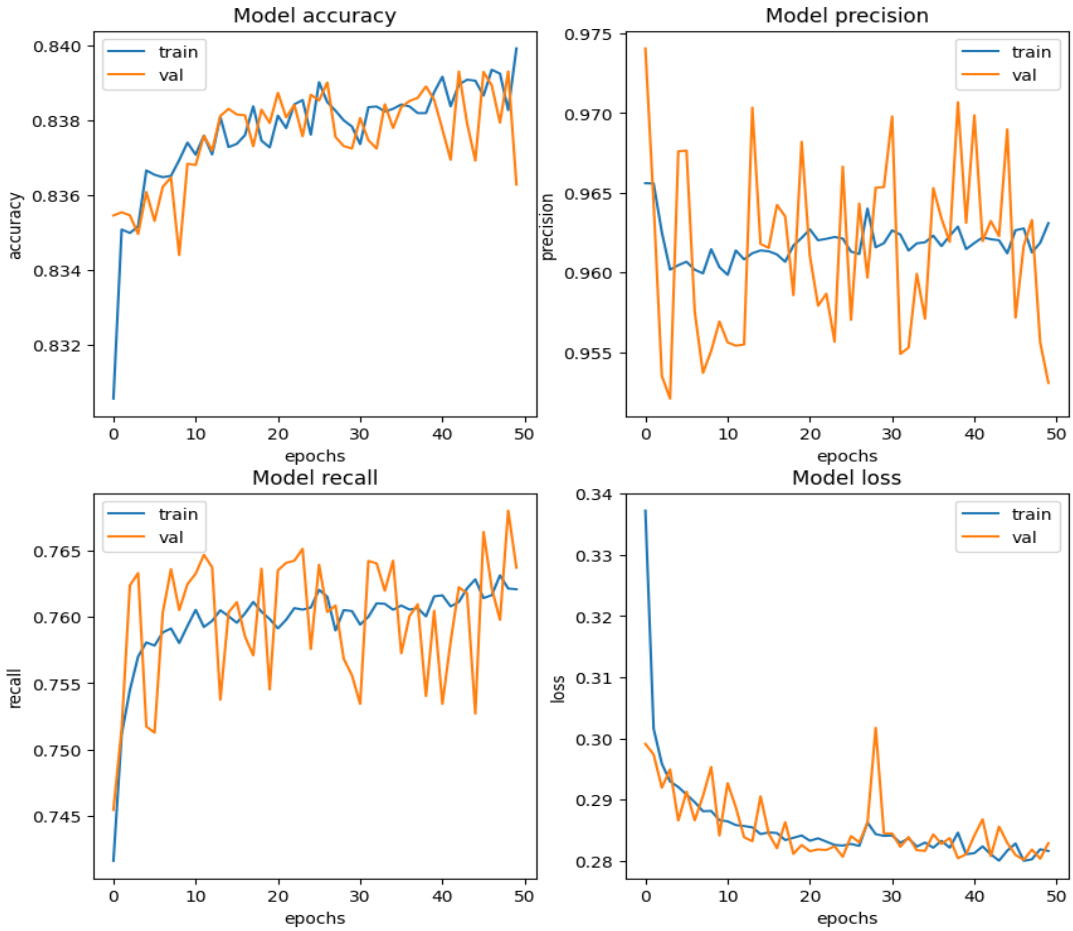
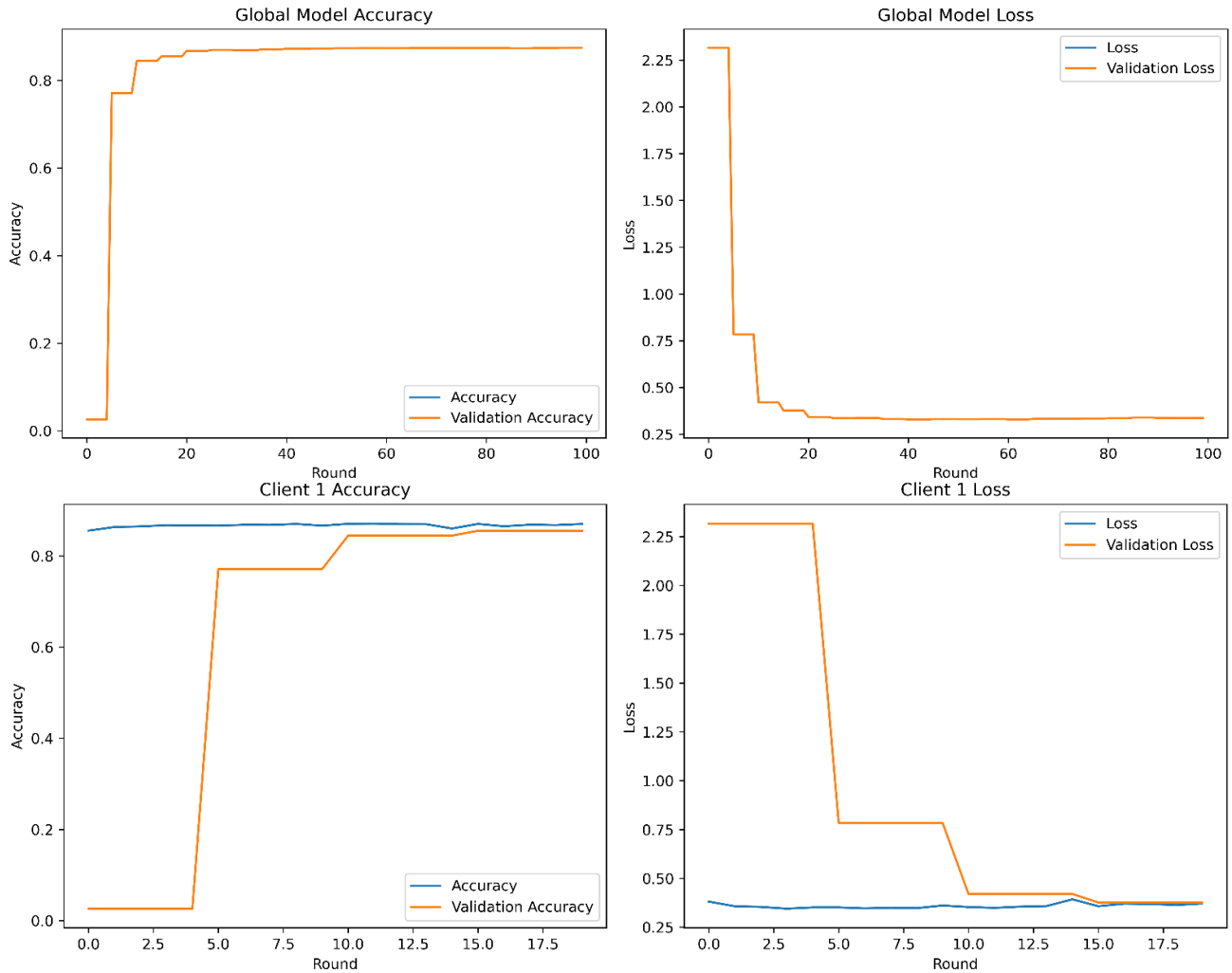


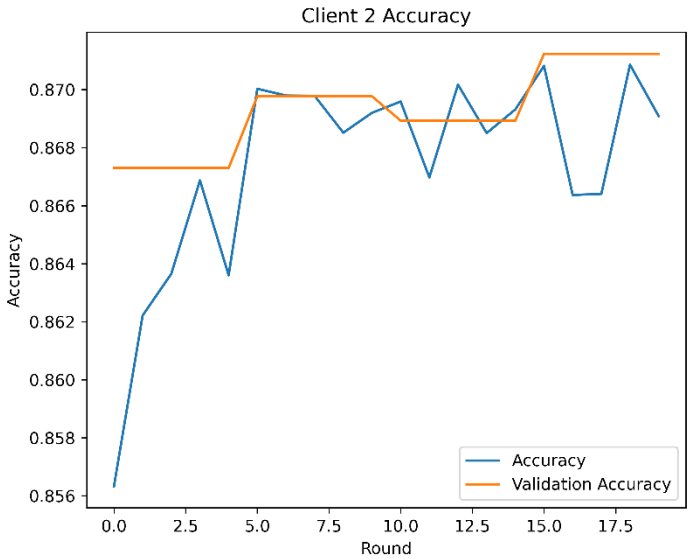
Figure 5. CSE-CIC-IDS2018 baseline model training performance

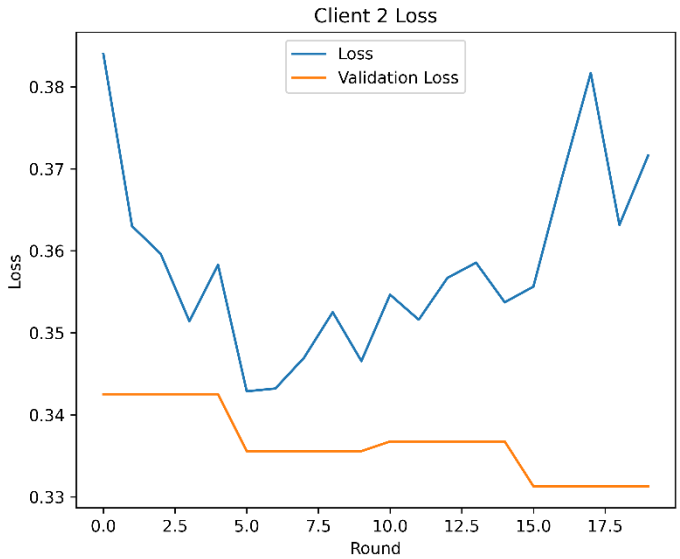
The loss curve of UNSW-NB15 can be seen to stay high longer; the jaggedness is caused by the difficulty that the model has in classes that are difficult to detect but underrepresented in the data, like Analysis or Worms. The loss curve of Bot-IoT is usually steeper and lower, illustrating the high separability of the classes; the DNN gives a definite mathematical distinction between a DDoS attack and an ordinary temperature sensor measurement. In spite of the fact that both of these datasets are used by network intrusion detection systems, Bot-IoT tends to give smoother curves with high accuracy due to the more aggressive and clear patterns of attacks. UNSW-NB15 is considered a more difficult real-world task, where the DNN must work harder in order to identify some hidden anomalies among normal background noise.

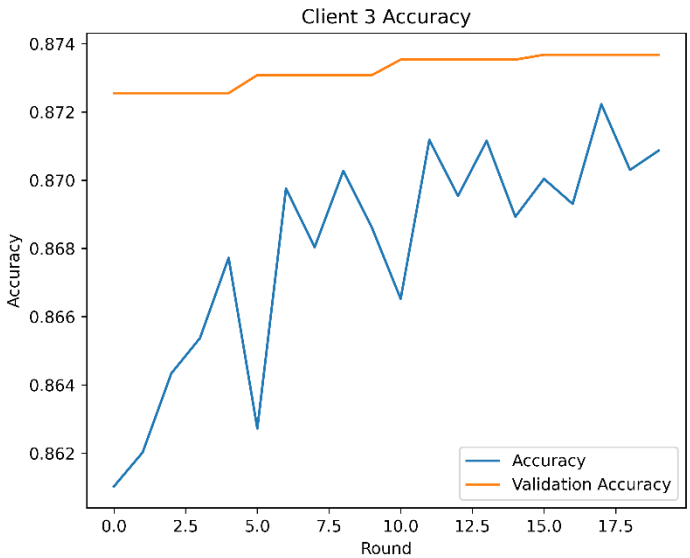
Figure 6 outlines the performance of a federated deep neural network trained on the UNSW-NB15 dataset and highlights an effective convergence of the global model in addition to the significant local variance between the involved clients, which is due to the non-IID characteristics of network intrusion data. The international paradigm shows a steep learning curve with the first ten rounds, and reaches the desired stability of an excellent accuracy of around 85% timer centered on the exemplary generalization as shown by the very similar training and evaluation statistics. However, a client-level analysis reveals a heterogeneous topography where some nodes, like Client 5, hit local stability early on, and others, like Client 2 and 4, experience local overfitting and increased volatility. Specifically, the trend in training loss observed to increase in Client 4 and the oscillations in Client 2 point to the effects that these nodes are facing complex and minority attack classes of fuzzers or backdoors, which are difficult to balance in the global weight paradigm. Finally, the fact that the global model is resilient despite these local anomalies shows that the federated aggregation process is an effective method of scaling out the noise in individual nodes, thus providing a generalized intrusion-detection signature.

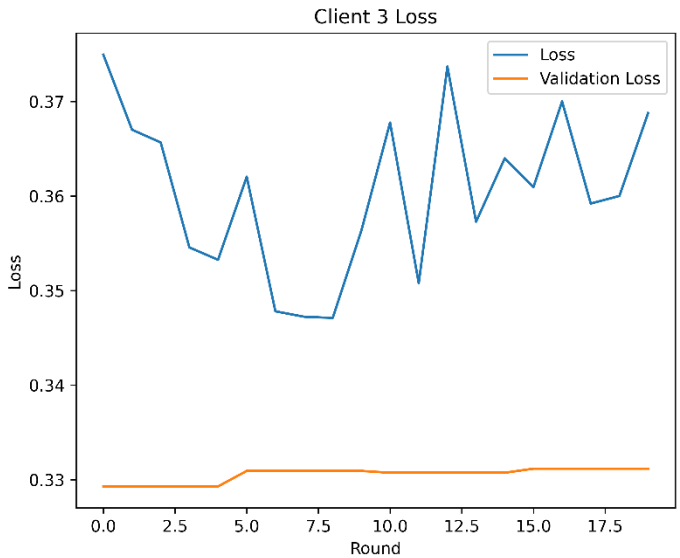
The results of the work, obtained by using a DNN on the Bot-IoT corpus as in Figure 7, show that the IDS has an impressive level of efficiency, and its learning curve reaches an almost perfect accuracy level of approximately 99.9% in the first few training epochs. The agreement between the training and validation accuracy curves indicates that the model extrapolates very well to unexamined network traffic, hence averting the fear of significant overfitting. In terms of the loss function, categorical cross- entropy decreases exponentially and plateaus at zero, indicating that the DNN has learnt the distinguishing patterns required to distinguish between benign and hostile IoT traffic, with high confidence, a range of hostile signatures such as DDoS, DoS, and reconnaissance attacks. In comparison between different clients, usually in a federated or distributed implementation, the performance is significantly stable; small fluctuations in validation loss can be explained by the sample bias of the Bot-IoT dataset, where the cases of the attack are overrepresented. Taken together, the ability of the DNN to reduce complex features of the network flows without grooming makes the framework a robust tool to protect the IoTs with limited resources against advanced botnet attacks.

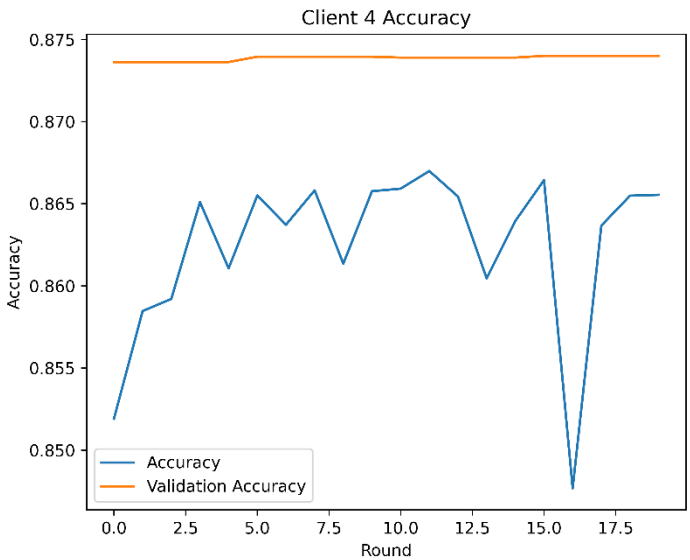


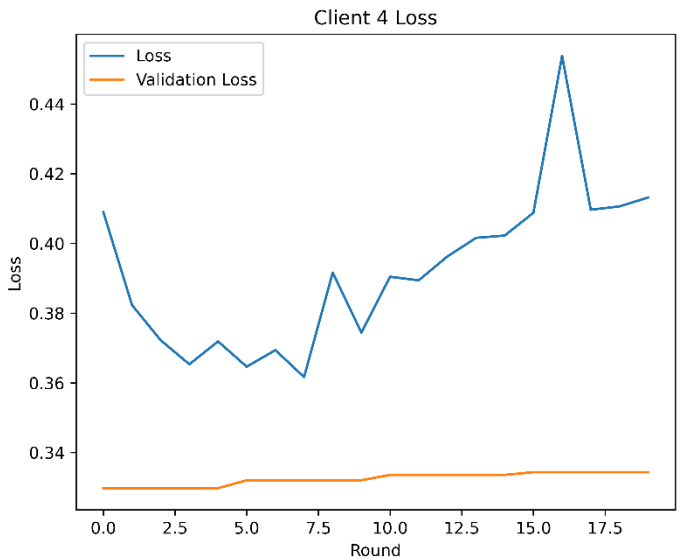

Client 2 Accuracy
Accuracy
Round
Accuracy
Validation Accuracy


Client 2 Loss
Loss
Round
Loss
Validation Loss


Client 3 Accuracy
Accuracy
Round
Accuracy
Validation Accuracy


Client 3 Loss
Loss
Round
Loss
Validation Loss


Client 4 Accuracy
Accuracy
Round
Accuracy
Validation Accuracy


Client 4 Loss
Loss
Round
Loss
Validation Loss

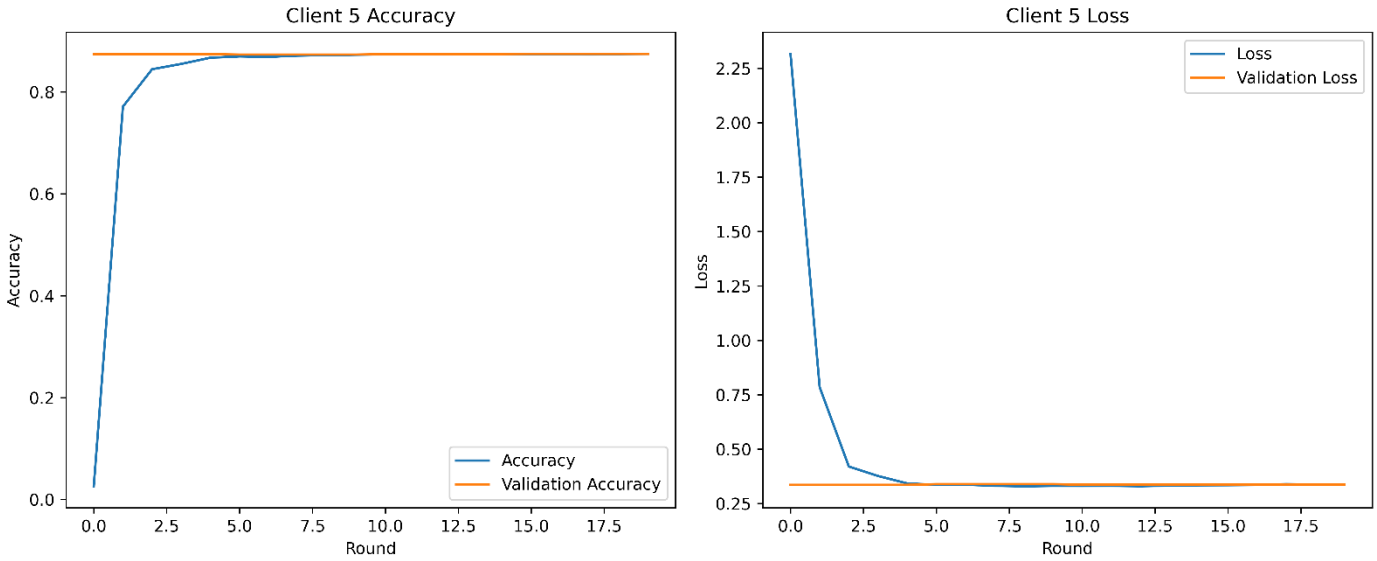
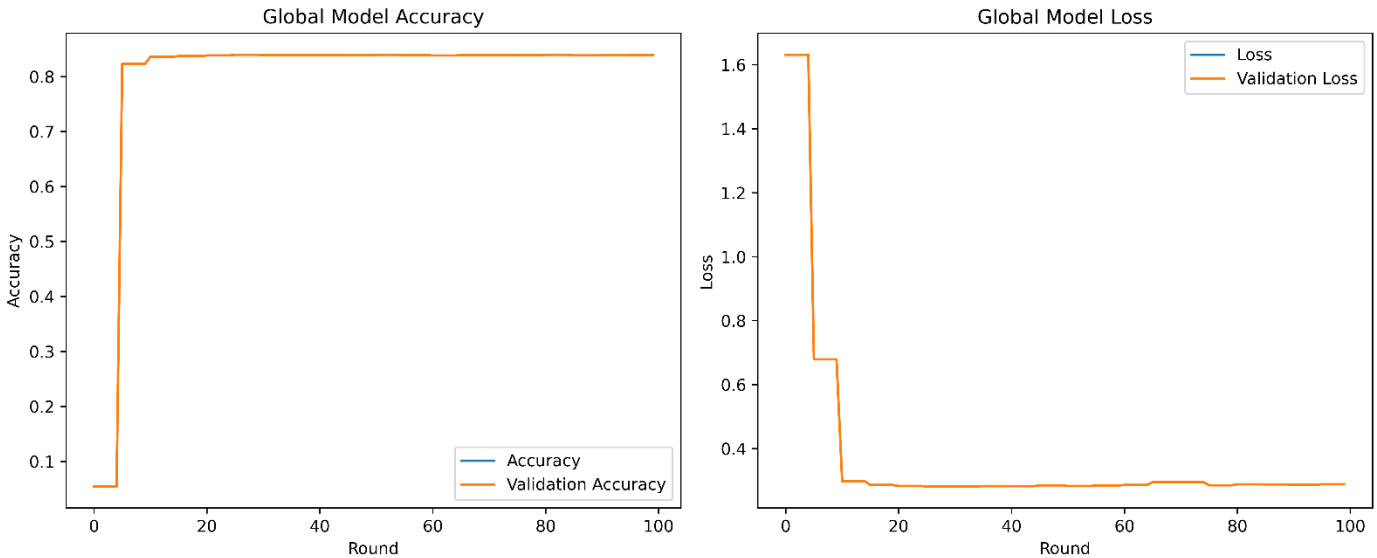
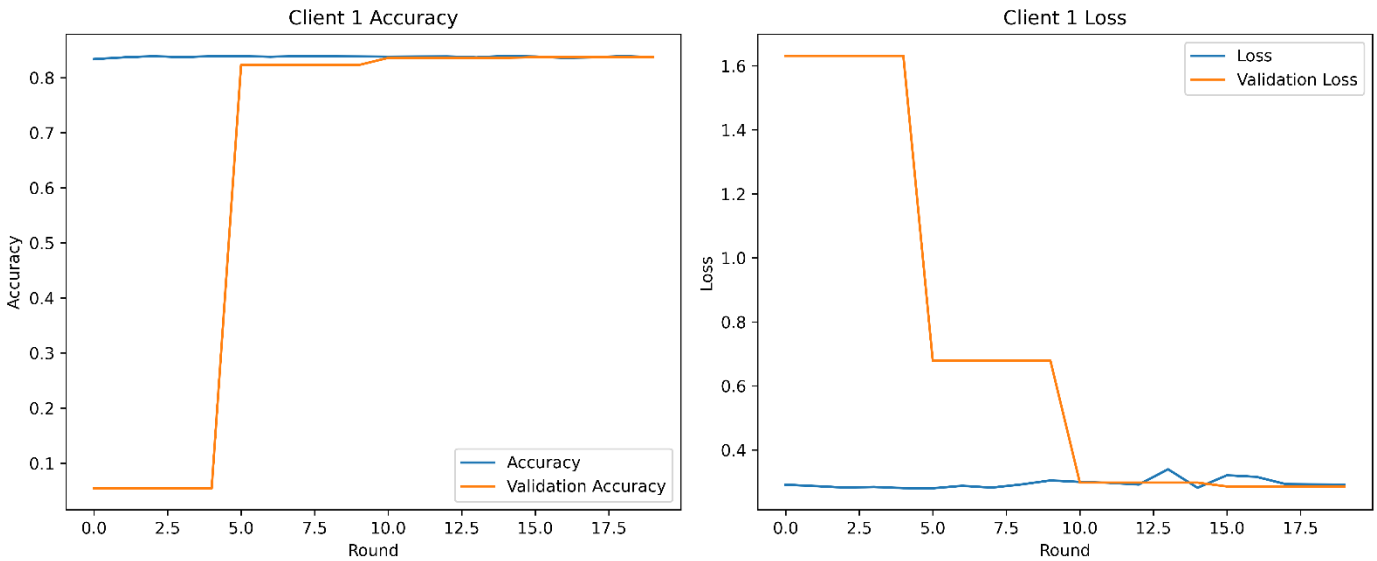
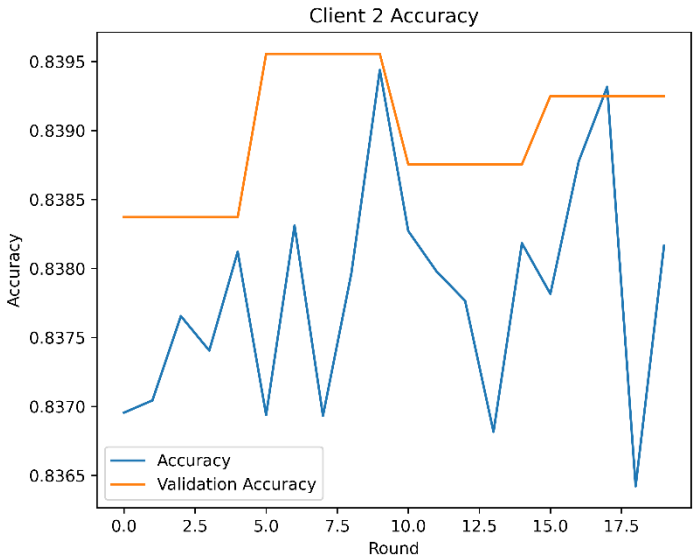
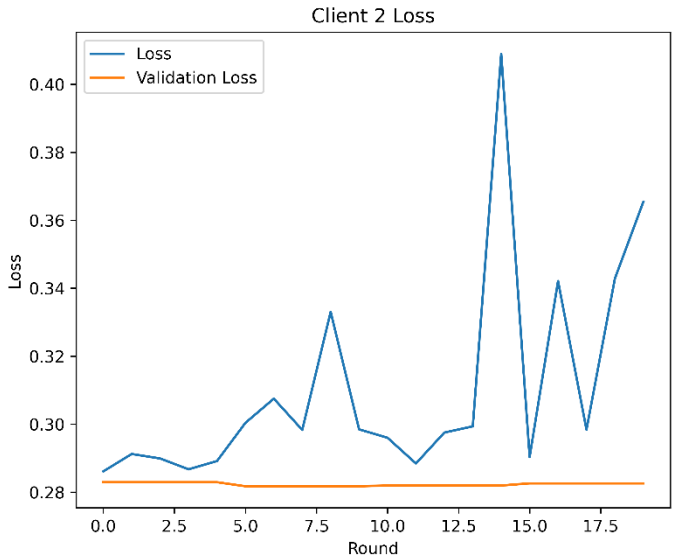


Figure 6. UNSW-NB15 federated learning client model training performance

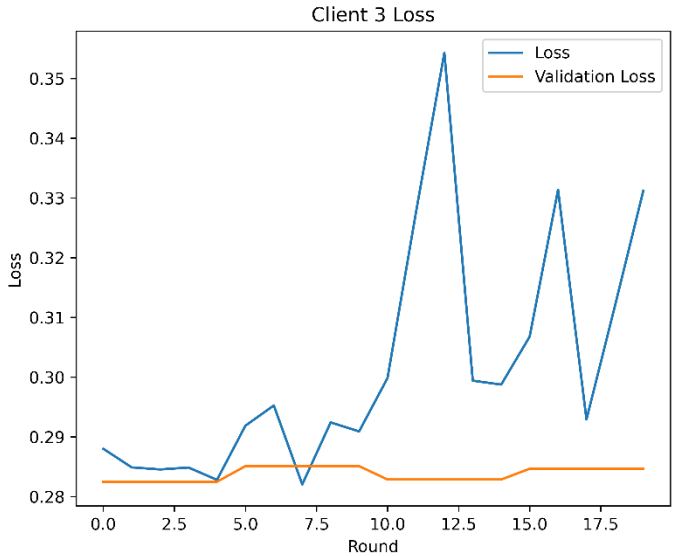


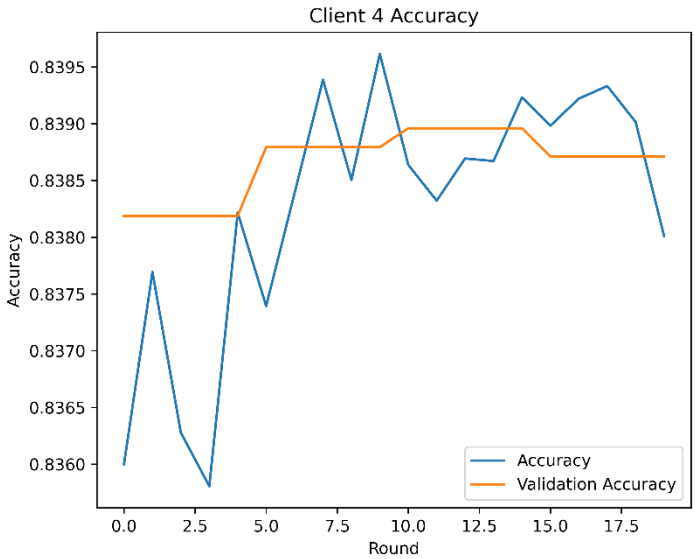


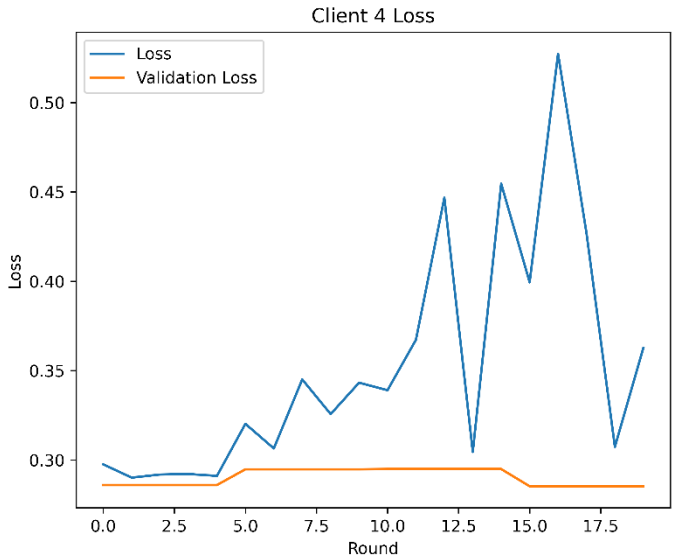

Client 2 Accuracy
Accuracy
Validation Accuracy
Round
Accuracy


Client 2 Loss
Loss
Validation Loss
Round
Loss


Client 3 Accuracy
Accuracy
Validation Accuracy
Round
Accuracy


Client 3 Loss
Loss
Validation Loss
Round
Loss


Client 4 Accuracy
Accuracy
Validation Accuracy
Round
Accuracy


Client 4 Loss
Loss
Validation Loss
Round
Loss

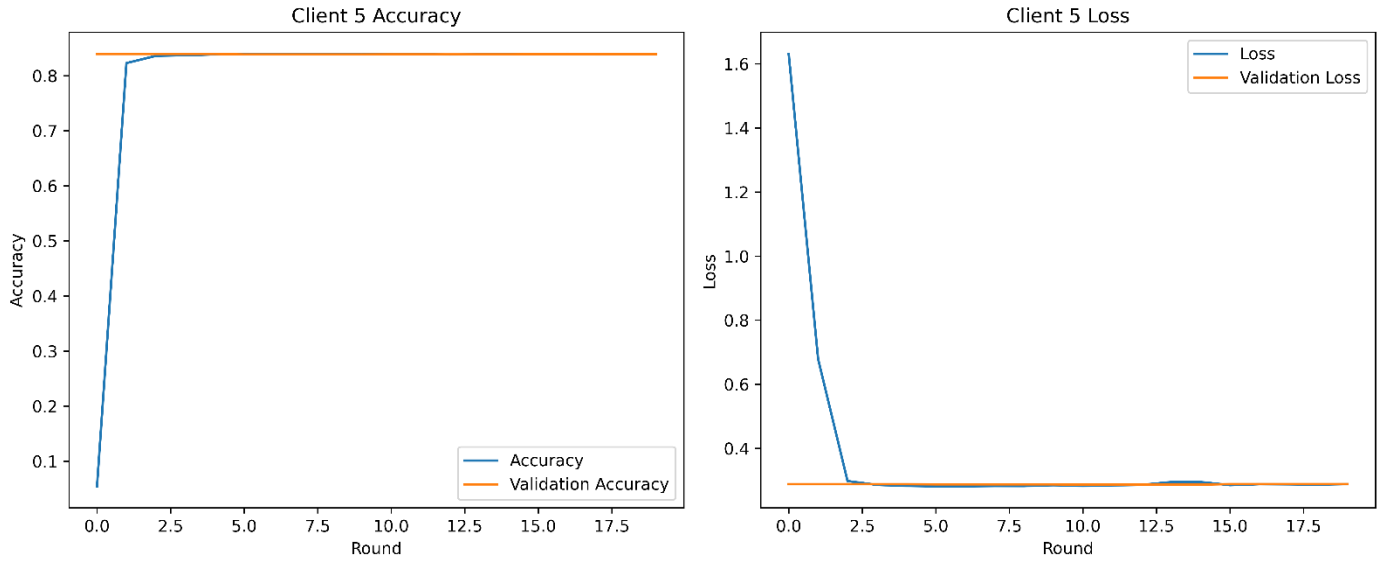


Figure 7. CSE-CIC-IDS2018 federated learning client training performance

6. CONCLUSIONS

This paper has managed to design and deploy an IDS containing a Centralized Federated Deep Learning architecture in an SDN environment modified to IoT networks. The experimental results, summarized in Table 4, demonstrate that the FL model achieved a superior test accuracy of 87.40% on the UNSW-NB15 dataset compared to the 85.92% achieved by the baseline model, while maintaining comparable performance on the Bot-IoT dataset with an accuracy of 84.06%. These findings prove that the proposed decentralized approach not only secures sensitive data by keeping it local to IoT devices but also provides robust detection capabilities against sophisticated network threats. By leveraging the programmability of SDN to manage global model updates, this framework offers a scalable and efficient solution for modem IoT security, with future work aiming to enhance its resilience against adversarial poisoning attacks and further optimize communication overhead. Through FL, the system effectively addresses the acute data privacy and bandwidth overhead issues that are commonly linked to traditional centralized IDS, and keeps the raw information on the clients and broadcasts only model parameters. The experimental assessment has used a six-layer DNN and the UNSW-NB15 and Bot-IoT data sets, so it proves the strength of the toolset offered:

- Performance: The global federated model achieved an excellent correctness of 87 percent on the UNSW-NB15 dataset, compared with the 86 percent of the centralized model, and a similar loss of 0.3708.
- Heterogeneity resilience: Despite the training volatility faced at each client due to non-IID data distributions, specifically when facing complex classes of attackers, such as fuzzers, the federated aggregation mechanism (FedAvg) is skilled at reducing local noise, which contributed to a stable and generalizable global model.
- Efficiency: Programmable, real-time traffic policing and orchestration is provided by the confluence with SDN, which is essential to resource-constrained IoT devices that cannot support heavy, standalone detection subsystems.

Altogether, Federated Learning in conjunction with SDN provides a scalable, privacy-saving, and effective strategy to protect modern IoT systems against an array of advanced cyber-attacks. Future studies will focus on enhancing the resiliency of the system to adversarial perturbations, e.g., poisoning attacks, and investigate more advanced adaptive aggregation algorithms to increase performance in strongly heterogeneous network settings.

Table 4. Train and test a comparison for both datasets

Dataset	Model	Accuracy		Loss		Precision		Recall	
		Train	Test	Train	Test	Train	Test	Train	Test
NB15	baseline	0.8621	0.8592	0.3658	0.3716	0.9480	0.9501	0.7944	0.7890
	FL	0.8711	0.8740	0.3614	0.3708	0.9455	0.9480	0.8122	0.8084
Bot-Iot	baseline	0.8397	0.8411	0.2789	0.3213	0.9645	0.9689	0.7612	0.7592
	FL	0.8395	0.8406	1.0387	1.4174	0.9693	0.9538	0.7567	0.7712

REFERECES

[1] Jang, S.B. (2015). Collaborative documentation process model. International Journal of Software Engineering and its Applications. https://doi.org/10.14257/ijseia.2015.9.5.21

[2] Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492. https://doi.org/10.48550/arXiv.1610.05492

[3] Abdalah, R.W., Abdulateef, O.F., Hamad, A.H. (2025). A predictive maintenance system based on the Industrial Internet of Things for multimachine multiclass using

deep neural network. Journal Européen des Systèmes Automatisés, 58(2): 373-381. https://doi.org/10.18280/jesa.580218

[4] Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G. (2021). A survey on security and privacy of federated learning. Future Generation Computer Systems, 115: 619-640. https://doi.org/10.1016/j.future.2020.10.007

[5] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y. (2021). A survey on federated learning. Knowledge-Based Systems, 216: 106775. https://doi.org/10.1016/j.knosys.2021.106775

[6] Wen, Z., Yang, R., Garraghan, P., Lin, T., Xu, J., Rovatsos, M. (2017). Fog orchestration for internet of things services. IEEE Internet Computing, 21(2): 16-24. https://doi.org/10.1109/MIC.2017.36

[7] Liu, L., Xu, B., Zhang, X., Wu, X. (2018). An intrusion detection method for internet of things based on suppressed fuzzy clustering. EURASIP Journal on Wireless Communications and Networking, 2018(1): 1-7. https://doi.org/10.1186/s13638-018-1128-z

[8] Saeed, S.H., Hadi, S.M., Hamad, A.H. (2022). Iraqi paradigm E-voting system based on hyperledger fabric blockchain platform. Ingénierie des Systèmes d'Information, 27(5): 737-745. https://doi.org/10.18280/isi.270506

[9] Valdivieso Caraguay, Á.L., Benito Peral, A., Barona López, L.I., García Villalba, L.J. (2014). SDN: Evolution and opportunities in the development IoT applications. International Journal of Distributed Sensor Networks, 10(5): 735142. https://doi.org/10.1155/2014/735142

[10] Nguyen, T.D., Rieger, P., Miettinen, M., Sadeghi, A.R. (2020). Poisoning attacks on federated learning-based IoT intrusion detection system. In Workshop on Decentralized IoT Systems and Security (DISS) 2020, San Diego, CA, USA, pp. 23-26. https://doi.org/10.14722/diss.2020.23003

[11] Rashid, M.M., Khan, S.U., Eusufzai, F., Redwan, M.A., Sabuj, S.R., Elsharief, M. (2023). A federated learning-based approach for improving intrusion detection in industrial internet of things networks. Network, 3(1): 158-179. https://doi.org/10.3390/network3010008

[12] Lazzarini, R., Tianfield, H., Charissis, V. (2023). Federated Learning for IoT Intrusion Detection. AI, 4(3): 509-530. https://doi.org/10.3390/ai4030028

[13] 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

[14] Janati Idrissi, M., Alami, H., El Mahdaouy, A., El Mekki, A., Oualil, S., Yartaoui, Z., Berrada, I. (2023). Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems. Expert Systems With Applications, 234: 121000. https://doi.org/10.1016/j.eswa.2023.121000

[15] Thein, T.T., Shiraishi, Y., Morii, M. (2024). Personalized federated learning-based intrusion detection system: Poisoning attack and defense. Future Generation Computer Systems, 153: 182-192. https://doi.org/10.1016/j.future.2023.10.005

[16] Jin, Z., Zhou, J., Li, B., Wu, X., Duan, C. (2024). FL-IIDS: A novel federated learning-based incremental intrusion detection system. Future Generation Computer Systems, 151: 57-70. https://doi.org/10.1016/j.future.2023.09.019

[17] Devine, M., Ardakani, S.P., Al-Khafajiy, M., James, Y. (2025). Federated machine learning to enable intrusion detection systems in IoT networks. Electronics, 14(6): 1176. https://doi.org/10.3390/electronics14061176

[18] Fatani, A., Abd Elaziz, M., Dahou, A., Al-Qaness, M.A.A., Lu, S. (2021). IoT intrusion detection system using deep learning and enhanced transient search optimization. IEEE Access, 9: 123448-123464. https://doi.org/10.1109/ACCESS.2021.3109081

[19] Masdari, M., Khezri, H. (2021). Towards fuzzy anomaly detection-based security: A comprehensive review. Fuzzy Optimization and Decision Making, 20 (1): 1-49. https://doi.org/10.1007/s10700-020-09332-x

[20] Dixit, P., Silakari, S. (2020). Deep learning algorithms for cybersecurity applications: A technological and status review. Computer Science Review, 39: 100317. https://doi.org/10.1016/j.cosrev.2020.100317

[21] Soniya, S.S., Vigila, S.M.C. (2016). Intrusion detection system: Classification and techniques. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, pp. 1-7. https://doi.org/10.1109/ICCPCT.2016.7530231

[22] Othman, S.M., Nabeel, T., Zahary, A.T., Alsohybe, N.T., Ba-Alwi, F.M. (2018). Survey on intrusion detection system types. International Journal of Cyber-Security and Digital Forensics, 7(4): 444-463. https://www.researchgate.net/publication/329360916

[23] Chen, G., Hu, Z., Qu, Y., Jin, D. (2025). Enhancing P4-based network emulation fidelity through a lightweight virtual time system and application evaluation. ACM Transactions on Modeling and Computer Simulation, 35(2): 1-24. https://doi.org/10.1145/3725530

[24] Bharti, S., Mcgibney, A. (2021). Privacy-aware resource sharing in cross-device federated model training for collaborative predictive maintenance. IEEE Access, 9: 120367–120379. https://doi.org/10.1109/ACCESS.2021.3108839

[25] Murtadha, M.K., Mushgil, B.M. (2025). Proactive hybrid half and joint D2D handover scheme in multi-access Mobile Edge Computing (MEC). Journal of Communications Software and Systems, 21(4): 436-446. https://doi.org/10.24138/jcomss-2025-0125