International Journal of Safety and Security Engineering
Vol. 13, No. 6, December, 2023, pp. 1135-1141
Journal homepage: http://iieta.org/journals/ijsse

# Intelligent Intrusion Detection Based on Multi-Model Federated Learning for Software Defined Network

Asraa A. Abd Al-Ameer[1,2*], Wesam Sameer Bhaya[3]

[1] Department of Information Networks, University of Babylon, Babel 51001, Iraq
[2] Department of Information Technology, University of Kerbala, Karbala 56001, Iraq
[3] Department of Information Security, University of Babylon, Babel 51001, Iraq

Corresponding Author Email: asraaabdalhussien@student.uobabylon.edu.iq

https://doi.org/10.18280/ijsse.130617

**Received:** 20 October 2023
**Revised:** 29 November 2023
**Accepted:** 18 December 2023
**Available online:** 25 December 2023

***Keywords:***
*artificial intelligence, machine learning, deep learning, federated learning, software-defined networks, network security*

## ABSTRACT

To address the challenges posed by traditional network architectures, the Software-Defined Network (SDN) architecture was introduced. However, SDNs are not immune to many security threats (e.g. Dos, Backdoors). In this paper, we present an advanced intrusion detection system that leverages federated learning (FL) and deep learning (DL) techniques to check whether attacks occur or not on SDN. FL has been employed as a collaborative learning technique, enabling various data planes to conduct local training on their respective client datasets. Following local training on each data plane, the local model parameters are securely transmitted to the controller server. At the controller server, these local training parameters are aggregated to construct a global model. The resulting aggregation outcome is then shared back with each local model to update them, enhancing their ability to detect attacks. Three datasets were used to evaluate the efficacy of the suggested method: UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017. The obtained results demonstrate a strong performance in anomaly detection, with an accuracy value reach to 95.68%.

## 1. INTRODUCTION

As networks undergo continuous evolution, there is an increasing interest in exploring innovative methods for designing network architectures capable of addressing the intricacies of modern networks. Consequently, this has given rise to a novel networking paradigm known SDN [1]. SDN is widely acknowledged as a highly promising architecture for the future of computer networks. It introduces a division of the network infrastructure into two distinct components: the control plane and the data plane [2]. Although this architecture has many benefits, it is also vulnerable to many threats, such as security breaches. Existing efforts focusing on SDN security face certain limitations related to safeguarding network data privacy. Resolve resource utilization costs, minimize communication overhead, and address data island issues. Therefore, to tackle these limitations and given the multitude of security threats present in networks [3], this paper introduces a method that utilizes FL to establish an anomaly detection system for SDNs. This approach enables collaborative learning while preserving data privacy. This work makes use of several neural network architectures based on FL, such as the Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) [4, 5], each trained on unique datasets sourced from different data planes. The reason for using this approach is to facilitate the acquisition of diverse knowledge and insights from these sources, allowing data planes to collaborate in detecting attacks against any host by securely sharing their only encrypted model parameters with the control plane without needing to share the raw data [6].

## 2. RELATED WORK

FL works on the tenet that data that is decentralized and never leaves the local environment in which it was generated is used to train a central model. FL brings the computational processes to the data itself, eliminating the need to move the data to another location for analysis. Numerous researchers apply FL in various domains, including network security, as SDN security. Mehta et al. [7] have used FL to predict DDoS attacks on SDN environments where a neural network model is cooperatively trained by several devices or clients without requiring them to share their raw data with a central server. Instead, they utilize a global model by aggregating gradients to compute an average. The experimental findings of this method for detecting DDoS attacks reach a 99% accuracy rate. Ali et al. [8] have proposed an intrusion detection system that safeguards the privacy of end-user data. It revolves around the concept of locally training data using Artificial Neural Networks (ANN) to derive the model's weights. Subsequently, these weights are transmitted to a federated server for aggregation. Furthermore, it has achieved a rate of 98.85% of prediction accuracy, along with an F1-Score of 94.21%. Wang et al. [9] enhanced traffic anomaly detection by fusing FL with an unsupervised convolutional autoencoder and smoothly integrating it into the SDN architecture. They also devised an approach for selecting aggregating models that consider data volume, which leads to a reduction in the federation's training time and an improvement in the accuracy of the models. The evaluation, conducted on the CICIDS 2017 dataset, clearly indicates that the federated model outperforms the local model. Notably, the average Area Under the Curve (AUC) exceeded 90%, and the accuracy surpassed 80%. Ropout [10] has created the FedIoT platform, which contains the FedDetect algorithm designed especially for the Internet of Things (IoT) context's on-device anomaly data detection. To enhance the platform's performance, the researchers incorporated FL in combination with Deep Autoencoder. The experimental findings provided compelling evidence of federated learning's effectiveness in detecting various types of attacks across numerous devices. The accuracy of their system was 96%. All previous related works used FL aided with a single type of model while in the proposed work multi-models have been used and tested on many datasets one of them is the CICIDS 2017 dataset which was used by Wang et al. [9] and our work got more accurate results.

## 3. BACKGROUND

This paper proposed a system to enhance the security of SDN based on applying Federated Machine Learning (FML). The foundation of the SDN paradigm is the idea of dividing the control plane and data plane of a network, as Figure 1 illustrates and describes the fundamental architecture of SDN. One of the most amazing aspects of this transition is that it replaces the complicated routing devices present in traditional networks with simpler switches, which are then responsible for implementing policies as directed by an intelligent and programmable logically centralized controller [11]. One of the main security benefits of SDNs over traditional networks is the controller's ability to see the complete network within the SDN framework. However, it's important to note that even though SDNs offer this advantage it has disadvantages they are not immune to various security threats [12].

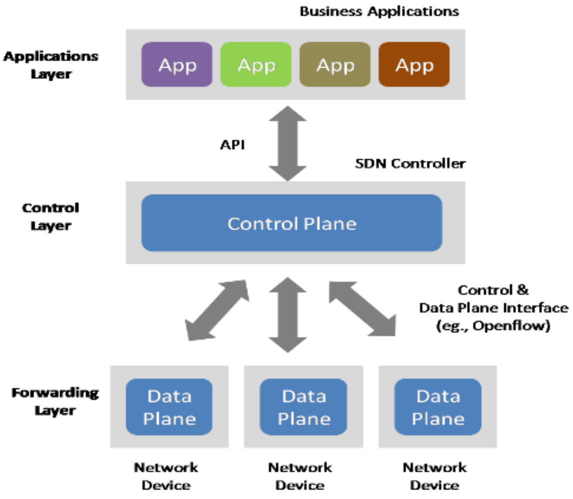


**Figure 1.** Software defined network architecture

FML is a methodology in which numerous devices collaboratively train a shared model. This is accomplished by sending their locally-computed updates to a central server. The client devices download the shared global model from the central server. As seen in Figure 2, each client utilizes its local dataset to update the model parameters [13, 14]. The benefit of this method is that it can train on a bigger dataset than any one device could handle by itself. Additionally, it improves data privacy by keeping data locally on the devices rather than sending it to a centralized location, which lowers communication overhead and strengthens the system [15].

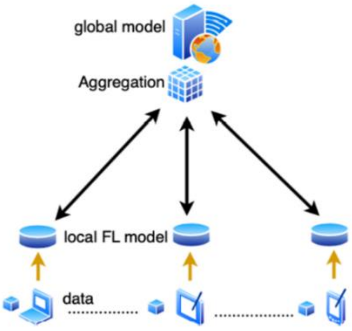


**Figure 2.** Federated learning architecture

A subset of machine learning known as DL was first introduced by ANNs [16] which has become a hot topic in many fields. One of the ANN types is the RNN [17].

In this paper, FML has been used to allow different data planes of SDN to train their DL model locally using their client's datasets to create a global model used to help each data plane detect attacks. This has harnessed the capabilities of deep neural networks to bolster security in SDN while upholding data privacy as has been shown in the next sections. HFL not only aids in detecting attacks but also addresses the challenge of data sharing among nodes while preserving data privacy. By training the ML model collaboratively without directly sharing raw data, HFL makes sure that private concerns and data sharing do not conflict and that sensitive data is secured [18].

## 4. METHODOLOGY

This research paper presents a method for detecting anomalies within SDNs using Multi-Model FL. SDN is split into two segments: Planes of Data and Control. At the controller server, the LSTM model will be built as a structure of a global model. Each data plane server uses its local DL model (RNN, LSTM, or GRU) to train its data and get the parameters (weights, bias, recurrent weights) for the model layers after training. Then, the parameters of each data plane are encrypted by homomorphic encryption and sent to the controller server. After decrypting it and averaging these characteristics, the controller uses each local model's significance in detecting the attack to create a global model. This technique allows SDN data planes to collaboratively gain insights from a shared detection model without disclosing sensitive data.

In the first communication iteration, the data plane server sends updates to the controller, assuming that the controller and data planes communicate for a total of 't' iterations. $P_G$ represents the controller's average aggregation of the model parameter as a result.

The definition of 'N' in this context is N = $|D_1| + |D_2| + \ldots + |D_s|$, where $D_i$ stands for the data samples for each switch's data plane. This sums up the total number of data samples for all data planes. Eq. (1) is the formula that the controller uses to update the global model in the first iteration.

$$P_G(1) = \sum_{k=1}^{s} \frac{|D_k|}{N} P_K \tag{1}$$

where, $P_K$ is the local model parameter (weight, recurrent weight, or bias) in the first iteration of updating. S in the count of DL local model (data plane), K refers to the data plane, $P_G$ refer to the averaged parameter in the first iteration.

In the iterations t after the first and in the aggregation process step that occurs at the server to create the global model, not only the averaged aggregation for the parameters have been taken, but these parameters are considered with the importance of each sub-model according to its contribution to detecting attacks at the data plane, as well as the other part of the importance has assigned according to the least loss that has been achieved through the training of its model (the difference between the real and expected value) based on the following formulas:

First, using Eq. (2), the Euclidian distance was utilized to calculate the difference between each local model $P_K$ parameter and the parameter of the global model $P_G$. It is utilized in order to measure the impact of each client's local parameters on the global parametric model optimization.

$$d(P_G(t), P_K(t)) = \sqrt{\sum_{i=1}^{n}(P_G\,(t)_i - P_K(t)_i)^2} \tag{2}$$

Sigmoid function denoted as Eq. (3) has used for normalize the result of Eq. (2).

$$f(X) = \frac{1}{1 + e^{-(x)}} \tag{3}$$

$$a_k(t) = f_{sigmoid}\,(d(P_G(t), P_K(t))) \tag{4}$$

The normalized result of Eq. (4) has been used in Eq. (5) to assign the first part of importance $h_k$ to the data plane local model.

$$h_k(\mathrm{t}) = \frac{a_k(t)}{\sum_{k=1}^{s} a_k(t)} \tag{5}$$

To enhance the importance, the second part of it was a weight will be added to each model according to the lowest loss obtained as shown in Eq. (6) and Eq. (7), where $w_k$ is the weight for specific data plane local model based on its loss $L_k$, $R_k$ is the rank value for the data plane local model compared with the others.

$$w_k(\mathrm{t}) = \frac{1}{L_k(t)} \tag{6}$$

$$R_k(\mathrm{t}) = \frac{w_k(\mathrm{t})}{\sum_{k=1}^{s} w_k(t)} \tag{7}$$

Therefore the last equation that give the importance for the data plane local model is:

$$m_{\,k}(\mathrm{t}) = h_k(\mathrm{t}) + R_k(\mathrm{t}) \tag{8}$$

The controller server will use the following Eq. (9) to aggregate model parameters for each data plane data $m_k$ based on its importance:

$$P_G(t) = \sum_{k=1}^{s} \frac{|D_k|}{N^*}\, m_k(t-1) * P_k^t \tag{9}$$

where, $N^*$ is all the data samples for all the data planes.

As a result, as previously mentioned, each iteration will dynamically assign the importance of each data plane based on how much the updated parameters from the previous iteration contributed to the global model optimization, with the exception of the controller server's average parameter aggregation in the first iteration. The controller receives this global model's encrypted transmissions and forwards them to the switches in each data plane partition. This procedure repeats itself on a regular basis to guarantee that the data is current. The SDN controller decides if a new flow entering the SDN network should be dropped (is an attack) based on the outputs of the updated local model. The suggested work diagram is displayed in Figure 3.

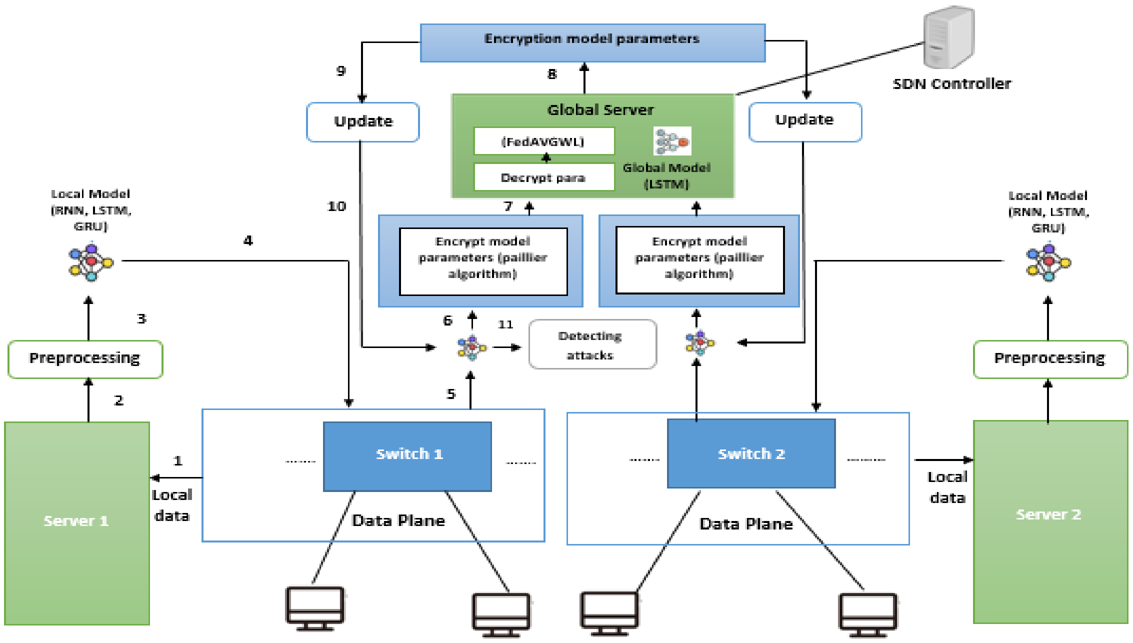


**Figure 3.** Proposed multi-model federated learning intrusion detection system

### 4.1 Intrusion detection in SDN based on modeled FL

In the proposed work, a FL model is employed to enhance the security of SDNs. This section provides a detailed description of the model. N edge servers, denoted as S, are responsible for collecting shared data D from hosts within their respective switches in the data plane. These servers, represented as $\{S_1, ..., S_N\}$, conduct training using their shared data. The training employs multiple DL models, including RNN, LSTM, and GRU models, as part of our approach.

Concurrently, a centralized deep learning model is trained by the SDN controller server, represented by the letter "c." It does this by using different weighted parameters that are obtained from several data planes and trained models. Through their combined efforts, these models help train the FL framework's primary model. We use the matrix $M_i$ to represent the data owned by data owner i (the edge server). In this illustration, a sample is represented by each row, and a feature is represented by each column. The sample IDs space is designated as Y, and the feature space is denoted as X. The suggested methodology is built utilizing a Horizontal Federated Learning (HFL) technique [19], in which agents cooperate on the common characteristics but have different sample IDs.

### 4.2 Multi-model learning for attack detection

In the proposed work, a combination of multiple deep learning models, including RNN, LSTM, and GRU, has been employed. These models are well-known known in the public domain for their prowess as dynamic classifiers [5]. The neural network nodes form a directed cycle that allows them to store information about past calculations [20]. As such, our model is able to understand the feature-to-feature correlations that result in attacks and can identify attacks within SDN. We have used three datasets, UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017, to validate our methodology. 700 thousand samples have been taken from each dataset. These datasets contain a blend of genuine, up-to-date normal network activities and contemporary synthesized attack behaviors to provide an intrusion detection system for SDN.

The network architecture for each model in the proposed system comprises an input layer implemented as a Linear layer (RNN/LSTM/GRU), a hidden layer with the output being passed through a ReLU activation function [21], Dropout layer, and Sigmoid Layer. The inclusion of the Dropout layer is essential to address potential overfitting issues that could otherwise lead to a decrease in performance when applied to the test dataset. In the analysis of the UNSW-NB15 dataset, all deep learning models utilize 19 input features encompassing attributes related to basic information, flow characteristics, content, and various features associated with network traffic across different links within the network. When working with the UQ-NIDS-v2 dataset and the CICIDS2017 dataset, the models have 5 features as input.

A prediction value that shows the likelihood of an attack happening when new flow traffic reaches the SDN on the data plane is the model's output. When such a prediction is made, the controller can then take action by issuing a notification to raise an alert and provide instructions to the network switch, directing it to drop the suspicious or potentially malicious traffic. All of the models used in the proposed study have unique configurations that define how many layers there are, how many neurons (units) there are in each layer, and how these layers are connected to one another. In the beginning, it's crucial to fine-tune these parameters to discover the optimal configuration that produces the highest accuracy and the lowest loss. Through experimentation, it was found that the models achieved their best performance when configured with 4 layers and between 265 to 512 neurons in each layer. Furthermore, it was noted that the Multi-model FL approach provides a broader range of knowledge and insights compared to using a single model as shown in the next section which exhibits superior performance and effectiveness in achieving the desired results.

## 5. EXPERIMENTS AND RESULTS

This section provides a detailed evaluation of the suggested federated learning technique's efficacy in identifying abnormalities in SDN. The evaluation is conducted using datasets including UNSW-NB15, CICIDS2017, and NF-UQ-NIDS-v2. Within this section, proposed work steps are presented. This includes dataset preprocessing, the utilization of a federated learning-assisted multi-neural network, as well as the assessment of categorization performance.

The studies that have been carried out involve the edge servers that are linked to the data planes using the switch-specific data to train the model. They then upload the modified model parameters for aggregation to the controller server.

### 5.1 Dataset preprocessing

Every original dataset has been pre-processed before being passed to the learning model for training. The pre-processing procedure includes cleaning, balancing, transformation, and normalization for the raw data.

5.1.1 Data cleaning

In the proposed system, when dealing with samples (rows) that contain missing data, the approach taken has been to handle them by simply ignoring or omitting those samples.

5.1.2 Data balancing

After preparing the data, each data plane has data with class imbalance (different distribution for normal and attack traffic classes). An imbalance in data distribution can result in less than optimal performance of deep learning models. Consequently, adjusting class distribution has been employed to address the challenges stemming from such class imbalance by making programmatically the sample count of the abnormal class equal to the sample count of the normal class.

5.1.3 Data transformation

In the proposed system, Min-Max Scaling has been employed as a normalization technique for numerical features [22]. This process transforms the numerical features so that they all fall within the same scale, typically ranging from 0 to 1.

$$x^* = \frac{x-x_m}{x_{max}-x_{min}} \tag{10}$$

where, $x^*$ represents the data after normalization, while $x$ stands for the initial, unnormalized data. Additionally, $x_{max}$ and "$x_{min}$" correspond to the maximum and minimum values, respectively, observed within the dataset for the specific.

### 5.2 Applicability of FL model

The FL approach has been integrated into the SDN architecture to detect potential threats in the event that they arise, after the data has been normalized. Every network switch collects information from the hosts that are connected to it and sends it to the data plane server that corresponds to it. The server will then use the shared data to develop a local machine learning model. By utilizing Paillier homomorphic encryption, all data plane servers will securely transmit model parameter weights, bias, and recurrent weights to the controller. Next, the controller decrypts these parameters, aggregates them, and calculates their average, taking into account their respective importance, as previously described. This process is essential for constructing the global model. The controller then safely sends the encrypted global model parameters to each data plane connected to the network, enabling them to retrain their local models using the decrypted values.

### 5.3 Classification performance evaluation

A table that represents a confusion matrix has been utilized to explain how well the suggested method has performed. Figure 4 shows a binary confusion matrix.

Accuracy is the preferred performance metric in the suggested FL-based system [23]. The percentage of accurately detected instances over the course of the full traffic trace is measured by accuracy. The ratio of packets correctly classified as normal or attack, divided by the total number of packets correctly and wrongly classified by the proposed system, is used to calculate accuracy. Eq. (11) represents this computation.

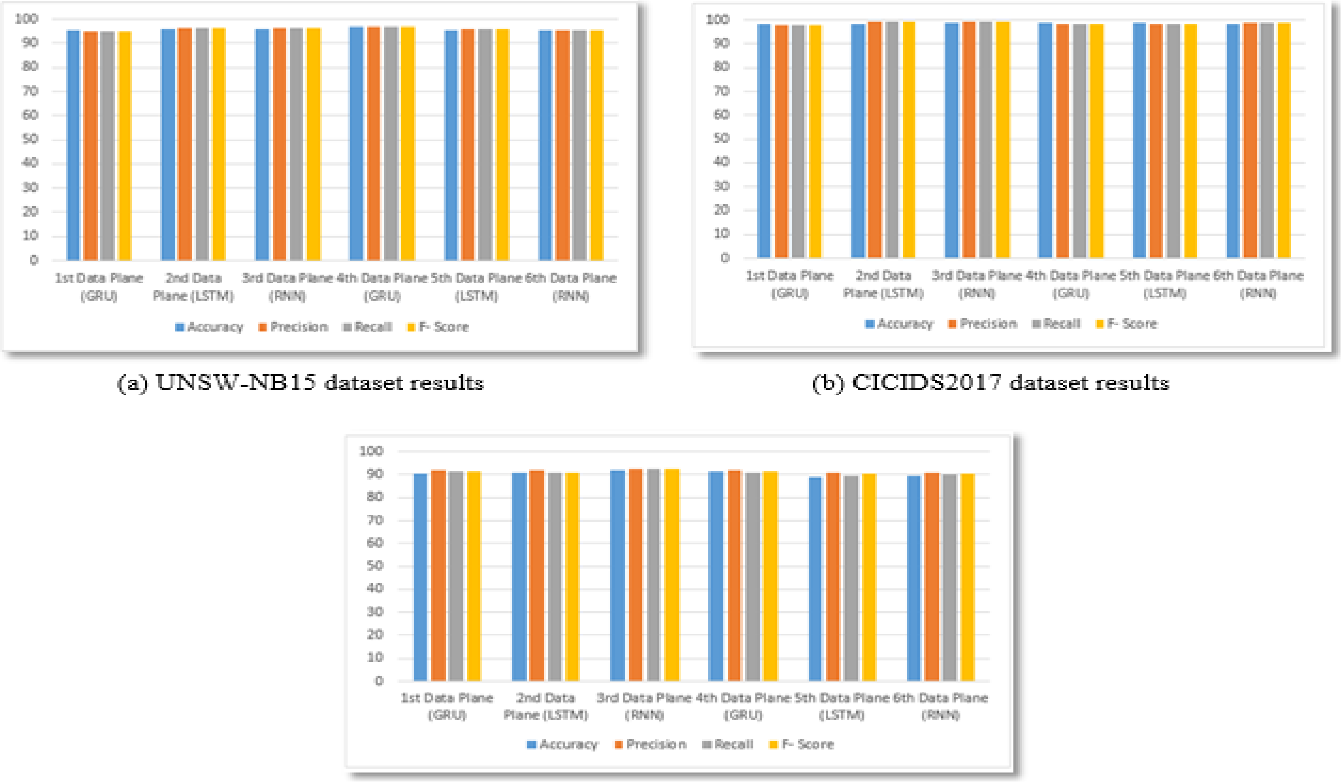
$$Accuracy = \frac{number\ of\ tru\ classifications}{total\ number\ of\ classifications} = \frac{TP+TN}{TP+FP+TN+FN} \quad (11)$$

SDN topology, which consists of one controller and six switches, is utilized to test the outcomes of the suggested work. The four hyperparameters of the FL model are N, B, E, and T. N denotes the number of hosts linked to each switch, B the size of the local batch, E the number of local epochs, and T the number of global round "iterations".

Three datasets-UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017-have been utilized to evaluate the suggested system's classification performance. Every switch has a separate value for N while B is set as 64, E value is set to 50, and T value is set to 10. Table 1, Table 2, Table 3, and Figure 5 (a, b, and c) shown models evaluation results of applying Multi-Model FL for UNSW-NB15, CICIDS2017, and NF-UQ-NIDS-v datasets, and Figure 6 shows an accuracy result comparison between using Multi-models FL and Single-model FL on the CICIDS2017 dataset.

| | | Predicted Label | |
|---|---|---|---|
| | | Normal | Anomaly |
| Actual Label | Normal | TP | FN |
| | Anomaly | FP | TN |

**Figure 4.** A binary confusion matrix



(a) UNSW-NB15 dataset results

(b) CICIDS2017 dataset results

(c) NF-UQ-NIDS-v2 results

**Figure 5.** The suggested system's detection performance across different datasets

**Table 1.** Models evaluation results of applying multi-model FL for the UNSW-NB15 dataset

| DL Model | 1st Data Plane (GRU) | 2nd Data Plane (LSTM) | 3rd Data Plane (RNN) | 4th Data Plane (GRU) | 5th Data Plane (LSTM) | 6th Data Plane (RNN) |
|---|---|---|---|---|---|---|
| **Accuracy** | 95.31 | 95.75 | 96 | 96.61 | 95.1 | 95.31 |

**Table 2.** Models evaluation results of applying multi-model FL for the CICIDS2017 dataset

| DL Model | 1st Data Plane (GRU) | 2nd Data Plane (LSTM) | 3rd Data Plane (RNN) | 4th Data Plane (GRU) | 5th Data Plane (LSTM) | 6th Data Plane (RNN) |
|---|---|---|---|---|---|---|
| **Accuracy** | 98.46 | 98.5 | 99.01 | 98.96 | 98.96 | 98.37 |

**Table 3.** Models evaluation results of applying multi-model FL for the NF-UQ-NIDS-v2 dataset

| DL Model | 1st Data Plane (GRU) | 2nd Data Plane (LSTM) | 3rd Data Plane (RNN) | 4th Data Plane (GRU) | 5th Data Plane (LSTM) | 6th Data Plane (RNN) |
|---|---|---|---|---|---|---|
| **Accuracy** | 90.33 | 90.78 | 91.72 | 91.37 | 90.78 | 90.42 |

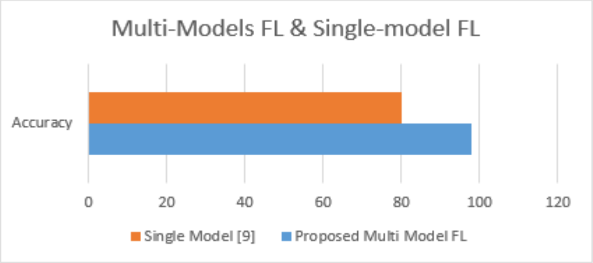


**Figure 6.** Comparison between using Multi-Models FL and Single-model FL

Applying the suggested system to datasets—of which 30% are used for testing and 70% are used for model training has allowed it to be assessed. As previously noted, the training models that are employed are RNN, LSTM, and GRU. The results of experiments show that collaborative FL training can achieve excellent accuracy in detecting SDN attacks while maintaining data privacy. The global model, which is based on FL collected on the global server, plays a crucial role in enhancing the accuracy of local models of different kinds at each data level to detect attacks, should they occur.

To simulate the proposed work, mininet version 2.3.1b1 with Ryu controller and an open virtual switch (OVS) have been used. Also, scapy tool has been used to generate packet flow in order to evaluate the system.

## 6. CONCLUSION

In this paper, an intelligent detection mechanism for SDN attack identification is introduced. The suggested solution makes use of FL in conjunction with several models that exhibit remarkable effectiveness in attack detection while maintaining data privacy. Upon predicting an attack, the controller is tasked with discarding the respective packet. The method achieved an impressive overall accuracy rate of 95.68% across diverse datasets. By successfully identifying network traffic threats and maintaining data confidentiality, decreasing communication overhead, and lowering resource consumption costs, this proposed method has the potential to improve SDN security. Utilizing training datasets tailored to the SDN environment was a limitation in examining the proposed work, one of the future works is to use a dataset tailored to the SDN to introduce a wider spectrum of SDN-relevant features.

## REFERENCES

[1] Rana, D.S., Dhondiyal, S.A., Chamoli, S.K. (2019). Software defined networking (SDN) challenges, issues and solution. International Journal of Computer Sciences and Engineering, 7(1): 884-889. https://doi.org/10.26438/ijcse/v7i1.884889

[2] Chica, J.C.C., Imbachi, J.C., Vega, J.F.B. (2020). Security in SDN: A comprehensive survey. Journal of Network and Computer Applications, 159: 102595. https://doi.org/10.1016/j.jnca.2020.102595

[3] Jimenez, M.B., Fernandez, D., Rivadeneira, J.E., Bellido, L., Cardenas, A. (2021). A survey of the main security issues and solutions for the SDN architecture. IEEE Access, 9: 122016-122038. https://doi.org/10.1109/ACCESS.2021.3109564

[4] Mathew, A., Amudha, P., Sivakumari, S. (2021). Deep learning techniques: An overview. In: Hassanien, A., Bhatnagar, R., Darwish, A. (eds) Advanced Machine Learning Technologies and Applications. AMLTA 2020. Advances in Intelligent Systems and Computing, vol 1141. Springer, Singapore. https://doi.org/10.1007/978-981-15-3383-9_54

[5] Staudemeyer, R.C., Morris, E.R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586. https://doi.org/10.48550/arXiv.1909.09586

[6] 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

[7] Mehta, N., Shukla, M., Shah, P., Patel, S., Shah, D., Makadiya, K. (2023). Distributed denial of service attack prediction over software defined networks using federated learning. Preprint. https://doi.org/10.21203/rs.3.rs-2832358/v1

[8] Ali, M.N., Imran, M., din, M.S.U., Kim, B.S. (2023). Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. Applied Sciences, 13(3): 1431. https://doi.org/10.3390/app13031431

[9] Wang, Z., Wang, P., Sun, Z. (2022). SDN traffic anomaly detection method based on convolutional autoencoder and federated learning. In GLOBECOM 2022-2022 IEEE Global Communications Conference, Rio de

Janeiro, Brazil, pp. 4154-4160. https://doi.org/10.1109/GLOBECOM48099.2022.10001438

[10] Ropout, D. (2021). Federated learning for internet of things: A federated learning framework for on-device anomaly data detection. ICLR, 1(2018): 1-12.

[11] Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A. (2015). Software-defined networking security: Pros and cons. IEEE Communications Magazine, 53(6): 73-79. https://doi.org/10.1109/MCOM.2015.7120048

[12] Cabaj, K., Wytrebowicz, J., Kuklinski, S., Radziszewski, P., Dinh, K.T. (2014). SDN architecture impact on network security. In FedCSIS (Position Papers), pp. 143-148. https://doi.org/10.15439/2014F473

[13] Abd Al-Ameer, A.A., Bhaya, W.S. (2023). Federated learning security mechanisms for protecting sensitive data. Bulletin of Electrical Engineering and Informatics, 12(4): 2421-2427. https://doi.org/10.11591/eei.v12i4.4751

[14] Man, D., Zeng, F., Yang, W., Yu, M., Lv, J., Wang, Y. (2021). Intelligent intrusion detection based on federated learning for edge-assisted internet of things. Security and Communication Networks, 2021: 1-11. https://doi.org/10.1155/2021/9361348

[15] Fatima, K., Zahoor, K., Zakaria Bawany, N. (2021). SDN control plane security: Attacks and mitigation techniques. In Proceedings of the 4th International Conference on Networking, Information Systems & Security, pp. 1-6. https://doi.org/10.1145/3454127.3456612

[16] Suzuki, K. (2013). Artificial Neural Networks: Architectures and Applications. BoD–Books on Demand.

[17] Abd Al-Ameer, A.A., Hussien, G.A., Al Ameri, H.A. (2022). Lung cancer detection using image processing and deep learning. Indonesian Journal of Electrical Engineering and Computer Science, 28(2): 987-993. http://doi.org/10.11591/ijeecs.v28.i2.pp987-993

[18] Zakaria, M., Mabrouka, A.S., Sarhan, S. (2014). Artificial neural network: A brief overview. International Journal of Engineering Research and Applications, 4(2): 7-12.

[19] Yang, Q., Liu, Y., Chen, T., Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2): 1-19. https://doi.org/10.1145/3298981

[20] Sacco, A., Esposito, F., Marchetto, G. (2020). A federated learning approach to routing in challenged sdn-enabled edge networks. In 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, pp. 150-154. https://doi.org/10.1109/NetSoft48620.2020.9165506

[21] Maas, A.L., Hannun, A.Y., Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In Proc. icml, 30(1): 3.

[22] Eliazar, I., Metzler, R., Reuveni, S. (2018). Universal max-min and min-max statistics. arXiv e-prints.

[23] Hassoun, M.H. (1995). Fundamentals of Artificial Neural Networks. MIT Press.