Ingénierie des Systèmes d'Information
Vol. 28, No. 5, October, 2023, pp. 1213-1220
Journal homepage: http://iieta.org/journals/isi

Enhanced Intrusion Detection in Software-Defined Networks Through Federated Learning and Deep Learning

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

[1] Department of Information Networks, University of Babylon, Babel 51002, Iraq
[2] Presidency of Kerbala University, University of Kerbala, Karbala 56001, Iraq
[3] Department of Information Security, University of Babylon, Babel 51002, Iraq

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

https://doi.org/10.18280/isi.280509

Received: 1 May 2023
Revised: 12 August 2023
Accepted: 23 August 2023
Available online: 31 October 2023

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

ABSTRACT

Software-defined networks (SDNs), while offering a revolutionary global view of the network, remain susceptible to a variety of attacks. This vulnerability necessitates innovative solutions for preserving data privacy and enhancing network security. The work presented herein introduces an innovative network anomaly detection methodology leveraging both federated learning (FL) and deep learning (DL) techniques. In contrast to traditional collaborative learning, where potential privacy compromises arise from the distribution of local training data to a central server, the proposed methodology enables each switch in the network to collect data from its connected hosts and independently train a local Long Short-Term Memory (LSTM) model. Subsequently, each switch encrypts and forwards its model parameters to the controller. Upon receipt, the controller decrypts the parameters from each switch, computes their average, and formulates a global LSTM model. This model is disseminated to every switch in the network, enabling each host to retrain its local model according to the global parameters. This iterative process is conducted multiple times to maintain the timeliness of the information. Evaluation of the proposed methodology using the UNSW-NB15 dataset, in conjunction with NF-UQ-NIDS-v2 and CICIDS2017 datasets, demonstrated its efficacy in anomaly detection, with performance exceeding a 96.75% accuracy rate.

1. INTRODUCTION

The field of networking technologies is experiencing an unprecedented expansion in user base, network devices, and applications. To meet escalating network demands, innovative models of network architecture are under exploration, among which a paradigm shift towards Software-Defined Networks (SDNs), pioneered by the Open Networking Foundation (ONF), has gained considerable attention [1]. This novel approach decouples the control plane from the data plane, empowering a remote controller to manage data plane network devices [2, 3]. Despite its inherent flexibility, the deployment of SDN in enterprise networks encounters significant obstacles, particularly those associated with security [4].

In the context of this study, a novel method has been proposed that employs Federated Learning (FL) to facilitate anomaly detection in SDNs while preserving data privacy. Network security threats are proliferating [5], and in SDNs, all three layers - application, control, and data - are susceptible to such threats. Thus, a security breach at any level can significantly compromise the other layers. For instance, Application Programming Interface (API) exploitation [6] can disrupt not only the application layer but also the control layer. Consequently, the implementation of detection mechanisms to counter security risks - such as information disclosure, tampering, spoofing, and repudiation - is essential for every network, including SDNs. These risks can originate from data planes, control planes, or the communication channels between them [5, 7].

The proposal in this work involves the development of a horizontal federated machine learning model [8] that seeks to enhance security in SDNs and improve their performance by detecting potential attacks. Leveraging a Long Short-Term Memory (LSTM) Deep Learning (DL) model, the federated learning approach enables local data training across multiple nodes connected to numerous switches. This methodology ensures data privacy and security and fosters effective collaboration among switches to detect attacks against any host or controller in the SDN, achieved by sharing encrypted model parameters with the controller [9, 10]. Previous works focusing on SDN security have encountered limitations in preserving network data privacy, addressing the data island problem, managing communication overhead, and optimizing resource utilization costs. The methodology proposed in this paper seeks to overcome these limitations.

2. RELATED WORK

Federated Learning (FL) operates on the principle of training a centralized model using data that remains in its local environment. This decentralized model eliminates the need to transfer data to a different location for analysis, instead bringing the computation to the data. Introduced by Google as a data leakage prevention mechanism, the focus has since shifted towards improving analytical challenges [11] and enhancing the security aspects of FL [12].

In a notable contribution, Man et al. [13] proposed the FedACNN system, an intelligent intrusion detection mechanism that enhances a Convolutional Neural Network (CNN) deep learning model using FL. The system was designed for implementation on edge devices, utilizing local datasets and computing resources to train the model before sending the model parameters to a central server for collaborative training. When tested on the NSL-KDD dataset, it was found that FedACNN significantly improved classification accuracy for attack data, achieving a detection rate of 99.76%.

Additionally, a security architecture augmentation for SDN, the Federated Distributed Integrated Clinical Environment (FedDICE), was proposed by Thapa et al. [14]. By integrating FL's privacy-preserving capabilities, it provided a tool to detect and mitigate ransomware attacks through collaborative learning. Testing on a clinical network traffic dataset demonstrated FedDICE's efficacy in detecting the spread of ransomware, with an accuracy rate of approximately 99% in the Distributed Integrated Clinical Environment (DICE).

Meanwhile, Li et al. [15] proposed an architecture named FLEAM that fuses FL and fog computing. The combination aimed to expedite mitigation, enhance detection accuracy, and foster cooperation among defenders against botnets in industrial IoT. Research findings suggested that FL could boost detection accuracy by up to around 95%.

Finally, Zhao et al. [16] introduced a two-stage learning approach named NAFT. Initially, a party seeking to develop a network anomaly detection model participates in FL to acquire foundational knowledge from other participants. Following this, they restructure the FL-trained detection model and retrain it using their private training data. Experiments conducted on the UNSW-NB15 dataset indicated that NAFT can outperform other methods in anomaly detection, especially under conditions of limited training data, with an accuracy exceeding 90%.

In summary, these studies underscore the potential of FL in enhancing security measures and improving detection accuracy in various network environments. However, they also highlight the need for further research to address remaining challenges and optimize federated learning systems.

3. BACKGROUND

The subsequent section provides a theoretical overview of the key mechanisms integral to the work under consideration, namely Federated Learning (FL), Deep Learning (DL), Homomorphic Encryption, and Software-Defined Networking (SDN). These mechanisms are combined to create an intrusion detection system based on an FL-aided LSTM.

3.1 Federated learning

Traditional machine learning (ML) pipelines involve gathering data from diverse sources and storing them centrally for model training [17]. This approach, however, exposes the data to potential privacy breaches during transfer to the central server. To safeguard data privacy, Federated Learning (FL) was proposed [18]. FL is a method wherein multiple devices collectively train a shared model by transmitting locally-computed updates to a central server [8, 13]. This technique enhances the capacity for training on larger datasets than a single device could manage, while also protecting data privacy by maintaining the data on the devices rather than transferring it to a centralized location [4].

3.2 Deep learning

Deep Learning (DL) models have been employed to construct the shared model in FL [19]. Owing to its ability to learn from data, DL technology, which originated from Artificial Neural Networks (ANNs), has found wide-ranging applications in areas such as healthcare, cyber-security, visual recognition, and more [20, 21]. DL utilizes multi-layer neural networks for computation and processing, with the term "deep" signifying multiple steps or levels of data processing to create a data-driven model [19, 22]. In the construction of a neural network, a multitude of interconnected processing elements, termed as 'neurons', are utilized [22]. It is from these neurons that a series of real-valued activations are generated, invariably leading to the attainment of the targeted result [23].

Various types of neural networks, including CNN, RNN, and LSTM [24], have been utilized with considerable success across numerous fields [20, 25]. In the proposed system, the LSTM model, which forms connections with preceding states in a sequence and addresses the vanishing gradient problem using specialized memory cells, has been incorporated to improve attack detection in SDN [25].

3.3 Homomorphic encryption

The protection of data privacy in FL necessitates the implementation of additional privacy techniques to prevent indirect leakage [8]. In the proposed system, Homomorphic Encryption has been utilized during the parameter exchange stage. This encryption method allows calculations to be performed directly on encrypted data, thereby eliminating the need for prior decryption [26, 27]. Unlike differential privacy protection, neither the data nor the model are transmitted, and they cannot be inferred from the other party's data, significantly reducing the risk of raw data leakage [8].

3.4 Software-Defined Networking

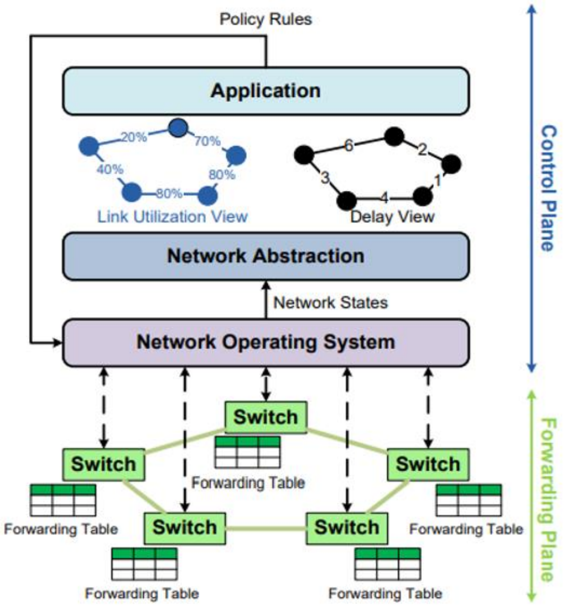

Figure 1. Software defined network architecture

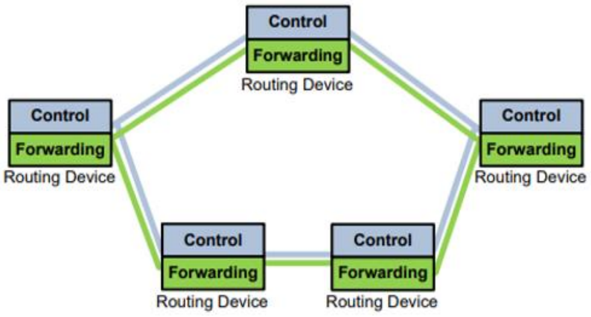

Figure 2. Traditional network architecture

Software-Defined Networking (SDN) is a novel networking paradigm predicated on the separation of the network's control plane from the data "forwarding" plane, as depicted in Figure 1 [28]. This decoupling allows simple switches to execute the policies defined by an intelligent, programmable, and logically centralized controller, in contrast to traditional networks where routing devices perform both forwarding and control functions using static protocols in a distributed manner, as shown in Figure 2 [1]. Despite the global network view provided by the SDN controller being potentially its most significant security advantage over traditional networks, it remains vulnerable to various attacks [28, 29]. The proposed work aims to address this issue by exploring the use of FL for detecting attacks in SDN.

4. PROPOSED METHOD

This research article introduces an approach for identifying anomalies in SDNs by combining FL with an LSTM model. This technique enables SDN switches to collectively acquire knowledge of a universal detection model while refraining from divulging sensitive information. At the network edge, the mechanism runs where various switches share their data to give the controller a decision. SDN is divided into two partitions: Data Plane and Control Plane. The Data Plane includes various switches to which servers and hosts are linked. A controller (Ryu has been used in our implementation) keeps tracking the weight metrics coming from the switches in the data plane partition. The weights represent the connection strength between neurons in DL model which updates based on the error between actual outputs, and predicted outputs using gradient descent as an optimization algorithm.

All the servers in all data plane partitions train LSTM local model by the switch data. Then, sending the weights parameters of the model after encrypting by homomorphic encryption to the switch, and in turn, the switch will send it to the controller. The controller decrypts it, takes the average of these weights, and uses it to train a global LSTM model.

Presuming that the controller and switches communicate for a total of t rounds, during the initial round of communication, the switches send updates to the controller, and the resulting average aggregation of the model parameter by the controller is denoted as W_G, and N = $|D_1| + |D_2| + . . . + |D_s|$ is the total number of hosts data samples for each switch. The controller updates the global model by applying the following formula, Eq. (1):

$$W_G = \sum_{k=1}^{s} \frac{|D_k|}{N} W_k \tag{1}$$

where the local model parameter of host k for each switch is W_k in the first round of updating.

This global LSTM sends in an encrypted way to the controller which in turn sends to the switches in all data plane partitions. This process returns periodically to ensure the freshness of information. According to the outputs of the global LSTM model and of the local LSTM model, the SDN controller determines if the new flow that enters the SDN network is an attack or not on the controller or on the hosts. Such a local model has as input the average weight of the local models that are built according to data traffic in the SDN data plane. Proposed work diagram is shown in Figure 3.

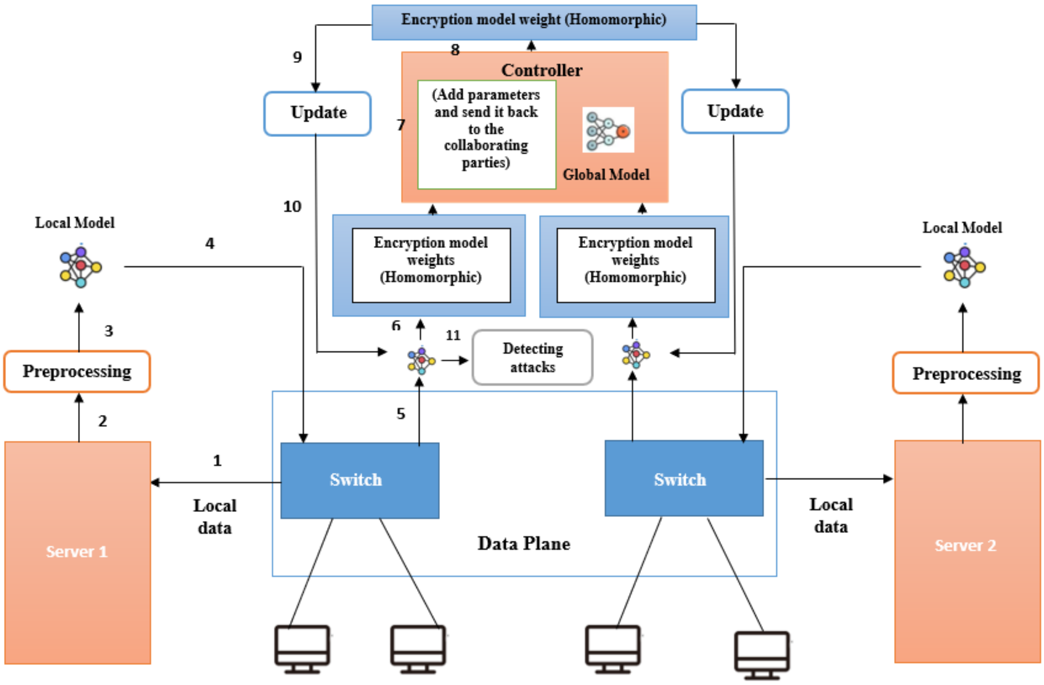
Encryption model weight (Homomorphic)
9
8
Update
Controller
(Add parameters and send it back to the collaborating parties)
7
Global Model
Update
10
Local Model
4
Encryption model weights (Homomorphic)
Encryption model weights (Homomorphic)
Local Model
3
6
11
Detecting attacks
Preprocessing
Preprocessing
2
5
1
Switch
Switch
Local data
Data Plane
Local data
Server 1
Server 2

Figure 3. Proposed federated learning intrusion detection systems

4.1 Modeling FL for detection attack in SDN

A FL model has been used in the proposed work to secure SDN therefore in this section, a description is explained for the model. N edge servers S collect the hosts-shared data D of their switch in the data plane. These servers $\{S_1, \ldots\ldots, S_N\}$ make training on their shared data using the DL model which is the LSTM model in our work. The SDN controller c; trains a centralized deep learning model by using various weighted parameters for the trained models collected from multiple data plane switches which collaboratively train the model. To denote the data D for the data owner i (edge server), let considered the matrix M_i, where each row in the representation corresponds to a specific sample, and each column represents a particular feature. X is used to refer to the feature space, while Y is the designation for the sample IDs space. To build the proposed method, a horizontal FL (HFL) [8] has been proposed where agents differ in the sample ID but share the utilized features.

4.2 LSTM for learning how to detect attack

LSTM deep learning model has been used in the proposed work. It's a kind of neural network that is publicly known as a powerful dynamic classifier [30]. A directed cycle is formed among the nodes of the neural network, which enables them to store information regarding the previous computations. This arrangement allows the nodes to leverage sequential information [31]. Therefore, there is an ability to learn the correlation between features that cause attacks and can detect it occurred in SDN in our model. In the proposed model a UNSW-NB15, NF-UQ-NIDS-v2 and CICIDS2017 dataset have been used with.

The proposed system network structure consists of a linear layer (LSTM layer), LeakyReLU (Leaky Rectified Linear Unit) layers, a Dropout and a sigmoid layer. The input samples and hidden layers output are transformed by the linear layer. To avoid the "dying ReLU" problem [32], the activation function LeakyReLU is used and allows a small, non-zero, constant gradient. Overfitting may occur and cause to degraded performance on the test dataset, so the Dropout layer is used.

The LSTM model with UNSW-NB15 dataset has as input 29 features "F" like basic, flow, content, and features for various link traffic "K" in the network while with UQ-NIDS-v2 dataset has 16 features, and with CICIDS2017 has 26 features. On each dataset, the HFL mechanisms presented by Yang et al. [8] have been applied on these links and their features, where in this matrix, each row corresponds to a specific link traffic sample, while each column represents a distinct feature. Therefore, a matrix with K × F dimension is received as input to the LSTM model. The output is a value that predicts the occurrence or nonoccurrence of an attack for any new flow traffic entering the SDN in the data plane or in the control plane so the controller can then give a notification to alert and instruct the switch to drop this traffic. In SDN, each switch functions as a conventional layer 2 switch that learns from the incoming packets. The LSTM model in the study's proposal is defined by the number of layers, neurons, and connections between the layers. Initially, these parameters should be adjusted to determine the optimal configuration that yields the highest accuracy and the lowest loss. Based on experimentation, the LSTM achieved the best performance with 5 layers and between 32 to 128 neurons. Increasing the number of hidden layers did not improve the deep learning system's performance.

5. EXPERIMENTS AND RESULTS

This section assesses the effectiveness of the suggested federated learning technique for detecting anomalies in SDNs. This approach enables switches to jointly acquire insights into a comprehensive detection model without jeopardizing privacy. UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017 datasets have been used for model evaluation. In this section, proposed work steps have been shown like dataset preprocessing, the applicability of the FL Model, and the evaluation of classification performance.

In the experiments that have been applied, the edge servers attached to the switches use local data to do training for the model and upload the parameters of the updated model to the controller for aggregation.

5.1 Dataset preprocessing

One of the datasets used in the proposed work is UNSW-NB15. UNSW-NB15 dataset includes normal traffic and all attack types records and consists of forty-five features and attributes [33]. NF-UQ-NIDS-v2 and CICIDS2017 also consist of benign and attack traffic.

Before passing it to the learning model for training each original dataset has been preprocessed. Preprocessing procedure includes transformation and normalization.

5.1.1 Transformation

From the 29 features used to build our model, UNSW-NB15 dataset has three character-based features and attributes, protocol, state, and service. We have used the one-Hot Encoding method [34] to get the numeric representation for these text features. In other words, any character-based features have to be in a numerical representation.

5.1.2 Normalization

Min-Max Scaling has been used in the proposed system to normalize the numerical features to be in the same scale i.e., between 0 and 1. Normalization formula is as Eq. (2) shown [35]:

$$x^* = \frac{x - x_m}{x_{\max} - x_{\min}} \tag{2}$$

where x^* denotes the normalized data, while x indicates the original data. Furthermore, x_{max} and x_{min} represent the highest and lowest values, respectively, of the attribute under consideration.

5.2 Applicability of FL model

After the normalization step, the FL mechanism has been applied to SDN to detect the attack in case it occurs. Every switch in the network collected the data of the hosts connected to it and sent this data to its server. In turn, the server will train a local LSTM model using the shared data. All the switches' servers will send model parameter "weights" to the controller in a secure way by encrypting it using paillier homomorphic encryption. Then, the controller will decrypt the parameter's "weights", aggregate, and get its average in order to build its global LSTM model. After that, it sends the encrypted global model parameter "weights" to every switch in the network and then every host will retain its local model according to its "decrypted parameters". As an illustration, Table 1 shows the first weight of the Global model in the first five iterations for three datasets in case 1.

This process will retrain in many iterations to save the freshness of the information. According to the global LSTM model output, the SDN controller determines if the new flow that enters the SDN network is an attack or not.

5.3 Classification performance evaluation

To describe the performance of the proposed system, a table which is a confusion matrix has been used. A binary confusion matrix is described in Figure 4.

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

Figure 4. A binary confusion matrix

The performance metric of the proposed system which used FL is accuracy. Accuracy is a metric that reflects the proportion of correctly detected instances in the overall traffic trace. To calculate accuracy, through the proportion between the number of packets correctly classified whether these packets are normal or attack over a total number of the packets classified by the proposed system correctly and incorrectly, as shown in Eq. (3).

$$Accuracy = \frac{\text{number of tru classifications}}{\text{total number of classifications}} = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

SDN topology which is shown in the Figure 5 was experienced to test the results of the proposed work. The topology of one controller and six switches. FL model hyperparameters are N, B, E, and T where N represents the number of hosts connected to each switch, B is the local batch size, E is local epochs numbers and T is the global round "iterations" number.

Two different cases (different numbers of samples) on three datasets UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017, one of 25000 samples and the other are 50000 samples have been used to test and compare the classification performance results of the proposed system. N value is different for each switch as shown in Figure 5, while B is set as 32 to the 25000 samples and set as 64 to the 50000 samples to get the best result according to the case. E and K value is set with two different values as Table 2 and Table 3 shown.

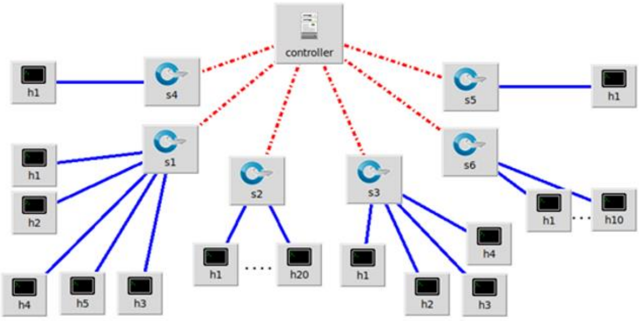

Figure 5. Proposed work SDN topology

Simulation results in the first case and on UNSW-NB15 dataset under the condition of 5 rounds of iteration and 20 epoches the proposed system has an accuracy 0.99 while when the rounds of iterations are 10 and 20 epochs the accuracy is 1. For the second case and also on UNSW-NB15 dataset, under the condition of 5 rounds of iteration and 10 epochs, the proposed system has an accuracy 0.99, which is the same accuracy result when the rounds of iterations are 10 and 20 epochs. Table 2, Table 3, and Figure 6, Figure 7 show the simulation results for the UNSW-NB15 dataset and the other datasets NF-UQ-NIDS-v2 and CICIDS2017 in the two cases.

The proposed system has been evaluated by applying it on UNSW-NB15, UQ-NIDS-v2, and CICIDS2017 datasets which 70% of them used for training the model and 30% is used for testing in the first case while 50% of them have been used for training the model and 50% is used for testing in the second case to get the best accuracy results. The training models used in it are LSTM.

This means that the higher the number of iterations, high the probability of better performance of the proposed system. Figure 8 illustrates the correlation between the number of communication iterations and the level of accuracy.

Experiments display that collaborative training of FL can achieve excellent accuracy to detect attacks in SDN while protecting data privacy. Future work includes building FL-based hybrid DL models and applying them to complex SDN topologies with huge traffic data.

Table 1. Global model weights in first five iterations

Dataset	Iterations	Layers Weights			
		Layer 1	Layer 2	Layer 3	Layer 4
UNSW-NB15	Iteration 1	-0.370038	-0.273164	0.201741	-0.241722
	Iteration 2	-0.459213	-0.322575	0.215829	-0.216713
	Iteration 3	-0.527186	-0.434171	0.220976	-0.196434
	Iteration 4	-0.463473	-0.427878	0.272371	-0.230685
	Iteration 5	-0.473676	-0.467626	0.303990	-0.225534
NF-UQ-NIDS-v2	Iteration 1	0.500880	- 0.169867	0.002923	-0.077887
	Iteration 2	0.271446	- 0.134888	0.044123	-0.107748
	Iteration 3	0.146077	-0.083110	0.030749	-0.171271
	Iteration 4	-0.096911	-0.053008	0.046418	-0.163636
	Iteration 5	-0.137259	-0.041089	0.040396	-0.167910
CICIDS2017	Iteration 1	0.596411	-0.444252	-0.285819	0.373098
	Iteration 2	0.521290	-0.490071	-0.470377	0.585126
	Iteration 3	0.554174	-0.547552	-0.414508	0.575711
	Iteration 4	0.647027	-0.594188	-0.440617	0.619177
	Iteration 5	0.681467	-0.594197	-0.359703	0.547394

Table 2. The detection performance of the proposed system in case 1 for three datasets

	Number of Iterations	Number of Epochs	Dataset	Accuracy
Case 1 (25000 Samples)	5	20	UNSW-NB15	0.99
			NF-UQ-NIDS-v2	0.94
			CICIDS2017	0.96
	10		UNSW-NB15	1
			NF-UQ-NIDS-v2	0.96
			CICIDS2017	0.95

Table 3. The detection performance of the proposed system in case 2 for three datasets

	Number of Iterations	Number of Epochs	Dataset	Accuracy
Case 2 (50000 Samples)	5	10	UNSW-NB15	0.99
			UQ-NIDS-v2,	0.95
			CICIDS2017	0.95
	10		UNSW-NB15	0.99
			UQ-NIDS-v2,	0.96
			CICIDS2017	0.97

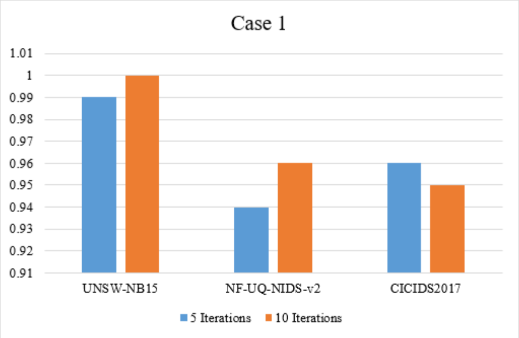

Figure 6. The detection performance of the proposed system for various datasets in case 1

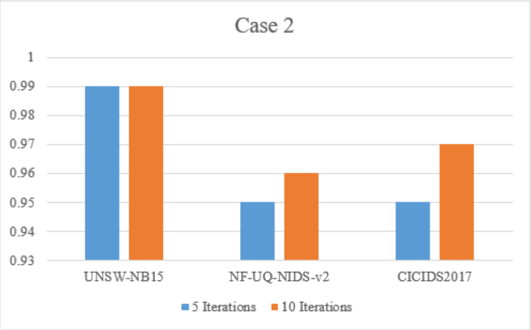

Figure 7. The detection performance of the proposed system for various datasets in case 2

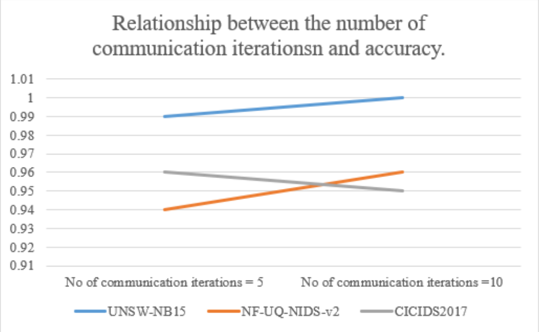

Figure 8. Relationship between the number of iterations and accuracy

6. CONCLUSIONS

The study in this paper presents an intelligent detection mechanism for identifying attacks in SDN. The proposed system is based on FL-aided LSTM which provides ideal performance in attack detections with protecting data privacy. When an attack is predicted, the controller must drop the packet. The method attained an overarching accuracy of 96.75% across various datasets. As the number of communication rounds between switches and the controller increased, accuracy also improved, illustrating the advantages of employing federated learning. The suggested technique can bolster the security of SDNs by identifying attacks in network traffic, all the while maintaining data confidentiality, minimizing communication overhead, and reducing utilization costs.

REFERENCES

[1] Haji, S.H., Zeebaree, S.R., Saeed, R.H., Ameen, S.Y., Shukur, H.M., Omar, N., Yasin, H.M. (2021). Comparison of software defined networking with traditional networking. Asian Journal of Research in Computer Science, 9(2): 1-18. https://doi.org/10.9734/ajrcos/2021/v9i230216

[2] 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

[3] 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

[4] 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, KENITRA AA Morocco, pp. 1-6. https://doi.org/10.1145/3454127.3456612

[5] 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

[6] Blial, O., Ben Mamoun, M., Benaini, R. (2016). An overview on SDN architectures with multiple controllers. Journal of Computer Networks and Communications, 2016: Article ID 9396525. https://doi.org/10.1155/2016/9396525

[7] Gao, S., Li, Z., Xiao, B., Wei, G. (2018). Security threats in the data plane of software-defined networks. IEEE Network, 32(4): 108-113. https://doi.org/10.1109/MNET.2018.1700283

[8] 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/10.48550/arXiv.1902.04885

[9] 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/10.1109/ACCESS.2020.3013541

[10] Wahab, O.A., Mourad, A., Otrok, H., Taleb, T. (2021). Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Communications Surveys & Tutorials, 23(2): 1342-1397. https://doi.org/ 10.1109/COMST.2021.3058573

[11] Konečný, J., McMahan, H.B., Ramage, D., Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527. https://doi.org/10.48550/arXiv.1610.02527

[12] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas Texas USA, pp. 1175–1191. https://doi.org/10.1145/3133956.3133982

[13] 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: 9361348. https://doi.org/10.1155/2021/9361348

[14] Thapa, C., Karmakar, K.K., Celdran, A.H., Camtepe, S., Varadharajan, V., Nepal, S. (2021). FedDICE: A ransomware spread detection in a distributed integrated clinical environment using federated learning and SDN based mitigation. In Quality, Reliability, Security and Robustness in Heterogeneous Systems: 17th EAI International Conference, QShine 2021, Virtual Event, pp. 3-24. https://doi.org/10.1007/978-3-030-91424-0_1

[15] Li, J., Lyu, L., Liu, X., Zhang, X., Lyu, X. (2021). FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT. IEEE Transactions on Industrial Informatics, 18(6): 4059-4068. https://doi.org/10.48550/10.1109/TII.2021.3088938

[16] Zhao, Y., Chen, J., Guo, Q., Teng, J., Wu, D. (2020). Network anomaly detection using federated learning and transfer learning. In International Conference on Security and Privacy in Digital Economy, Quzhou, China, pp. 219-231, https://doi.org/10.1007/978-981-15-9129-7_16

[17] Chamikara, M.A.P., Bertok, P., Khalil, I., Liu, D., Camtepe, S. (2021). Privacy preserving distributed machine learning with federated learning. Computer Communications, 171: 112-125. https://doi.org/10.1016/j.comcom.2021.02.014

[18] 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

[19] Sarker, I.H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2(6): 420. https://doi.org/10.1007/s42979-021-00815-1

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

[21] Narayanrao, P.V., Surya Kumari, P.L. (2023). Regularized CNN based model for analyzing, predicting depression and handling overfitting. Ingénierie des Systèmes d'Information, 28(1): 247-254. https://doi.org/10.18280/isi.280129

[22] Dong, S., Wang, P., Abbas, K. (2021). A survey on deep learning and its applications. Computer Science Review, 40: 100379. https://doi.org/10.1016/j.cosrev.2021.100379

[23] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, 6: 35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950

[24] Imaduddin, H., Kusumaningtias, L.A., A'la, F.Y. (2023). Application of LSTM and GloVe word embedding for hate speech detection in Indonesian twitter data. Ingénierie des Systèmes d'Information, 28(4): 1107-1112. https://doi.org/10.18280/isi.280430

[25] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, Florida, USA, pp. 1273-1282. https://doi.org/10.48550/arXiv.1602.05629

[26] Giacomelli, I., Jha, S., Joye, M., Page, C.D., Yoon, K. (2018). Privacy-preserving ridge regression with only linearly-homomorphic encryption. In Applied Cryptography and Network Security: 16th International Conference, ACNS 2018, Leuven, Belgium, pp. 243-261. https://doi.org/10.1007/978-3-319-93387-0_1

[27] Hall, R., Fienberg, S.E., Nardi, Y. (2011). Secure multiple linear regression based on homomorphic encryption. Journal of Official Statistics, 27(4): 669-691. https://doi.org/10.1109/IS3C50286.2020.00144

[28] 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

[29] Cabaj, K., Wytrebowicz, J., Kuklinski, S., Radziszewski, P., Dinh, K.T. (2014). SDN architecture impact on network security. In Position papers of the 2014 Federated Conference on Computer Science and Information Systems, pp. 143-148. https://doi.org/10.15439/2014F473

[30] 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

[31] 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

[32] Maas, A.L., Hannun, A.Y., Ng, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. Open Journal of Acoustics, 30(1): 14-24. https://doi.org/10.4236/oja.2013.31003

[33] Kanimozhi, V., Jacob, P. (2019). UNSW-NB15 dataset feature selection and network intrusion detection using deep learning. International Journal of Recent Technology and Engineering, 7(5): 443-446. https://doi.org/10.1109/ISIE.2017.800153

[34] Lyu, F., Tang, X., Guo, H., Tang, R., He, X., Zhang, R., Liu, X. (2022. Memorize, factorize, or be naive: Learning optimal feature interaction methods for CTR prediction. In 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia pp. 1450-1462. https://doi.org/10.1109/ICDE53745.2022.00113

[35] Yuan, D., Ota, K., Dong, M., Zhu, X., Wu, T., Zhang, L., Ma, J. (2020). Intrusion detection for smart home security based on data augmentation with edge computing. In ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, pp. 1-6. https://doi.org/10.1109/ICC40277.2020.9148632