

Federated deep learning intrusion detection system on software defined-network based internet of things

Heba Dhirar, Ali H. Hamad

Department of Information and Communications Engineering, Al-Khwarizmi College of Engineering, University of Baghdad, Baghdad, Iraq

Article Info

Article history:

Received Oct 4, 2024

Revised Jan 29, 2025

Accepted Mar 15, 2025

Keywords:

Federated deep learning

Internet of things

Interplanetary file system

Intrusion detection system

Software-defined network

ABSTRACT

The internet of things (IoT) and software-defined networks (SDN) play a significant role in enhancing efficiency and productivity. However, they encounter possible risks. Artificial intelligence (AI) has recently been employed in intrusion detection systems (IDSs), serving as an important instrument for improving security. Nevertheless, the necessity to store data on a centralized server poses a potential threat. Federated learning (FL) addresses this problem by training models locally. In this work, a network intrusion detection system (NIDS) is implemented on multi-controller SDN-based IoT networks. The interplanetary file system (IPFS) FL has been employed to share and train deep learning (DL) models. Several clients participated in the training process using custom generated dataset IoT-SDN by training the model locally and sharing the parameters in an encrypted format, improving the overall effectiveness, safety, and security of the network. The model has successfully identified several types of attacks, including distributed denial of service (DDoS), denial of service (DoS), botnet, brute force, exploitation, malware, probe, web-based, spoofing, recon, and achieving an accuracy of 99.89% and a loss of 0.005.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Heba Dhirar

Department of Information and Communications Engineering, Al-Khwarizmi College of Engineering

University of Baghdad

Jadriaa, Baghdad, Iraq

Email: heba.d@kecbu.uobaghdad.edu.iq

1. INTRODUCTION

The term "internet of things (IoT)" refers to connecting embedded devices to the internet. The idea behind the IoT is to enable everyday items to be connected over the network and gather vast amounts of data from devices with different powers and limited resources; hence, enforcing security and protection can be challenging [1]. Network traffic analysis and abnormal activity identification are resource-intensive tasks. Over the last several years, several lightweight approaches for improving IoT security have been created [2], [3], but these systems are unable to handle the significant security risks that have been discovered lately. Hence, it is imperative to design effective intrusion detection to efficiently defend against various forms of attacks. Intrusion detection system (IDS) performs an essential part as the primary defense mechanism [4] which employs many approaches to identify and flag abnormalities. Significant advancements have been achieved using machine learning (ML) and deep learning (DL) in recent years, resulting in widespread use across several domains. It can offer techniques to identify various forms of attack without the need for significant human involvement. While these methods have proven effective for IDS, they often need a centralized server to analyze the data gathered from all network users. Federated learning (FL) is a means to implement on-device learning while preserving data privacy [5]–[7]. FL is an iterative procedure in which the

entire model may be enhanced in each round by training the model on many devices and using their data across numerous iterations without exchanging data with a centralized server achieving privacy preservation and cost reduction, as expected in conventional centralized approaches [8]. Software-defined networking (SDN) is an innovative architecture that separates network control from forwarding functions, enabling direct programmability of network management and enhancing operational efficiency. The SDN network utilizes these attributes to create a proactive system for detecting intrusions in IoT networks, making it a superior choice for overcoming the challenges faced in the efficient operation of IoT due to its programmability and comprehensive perspective [9], [10].

The network-based IDS, referred to as network intrusion detection system (NIDS), is designed to determine whether IP traffic is compromised by threats. The process consists of a training phase utilizing an accurate representation of recognized activities, followed by an operational classification and decision phase. The training and classification phases rely on the definition and extraction of a set of statistical parameters associated with each IP flow, which constitute the statistical fingerprint of the flow, and on DL classifiers designed to differentiate between normal and malicious traffic. In this study, FL was utilized to cooperatively train DL models to implement anomaly-based IDS on a multi-controller SDN-based IoT that leverages the characteristics of SDN to establish a proactive system for detecting intrusions in IoT networks. Several clients can obtain the DL model from the interplanetary file system (IPFS) network and participate in the training process by training the model locally on their custom-generated dataset and sharing only the parameters in an encrypted form using advanced encryption standards (AES) algorithm. This process enhances the overall efficiency, safety, and security. The model has successfully identified several types of attacks achieving an accuracy of 99.89% and a loss of 0.005.

The rest of the paper is organized as follows: section 2 focuses on IDS research in context of SDN and IoT networks, and section 3 contains comprehensive background analysis with the details of our custom-generated dataset IoT-SDN. The suggested methodology is discussed in section 4. The experimental results and efficacy of proposed method are presented in section 5, whereas section 6 outlines the work's conclusion.

2. RELATED WORK

The substantial amount of data and the diversity of devices make the security of the IoT a significant problem. IDSs have been developed employing various methodologies and strategies to secure and defend IoT networks. Several prominent intrusion detection algorithms recently developed to address security challenges in SDN and IoT networks are outlined in Table 1 which provides a summary of the researchers who have concentrated on implementing IDS on the SDN network.

Table 1. Survey of the most related work of IDSs on the SDN and IoT networks

Reference	Year	Network	Dataset	Technique	Accuracy (%)
Tang <i>et al.</i> [11]	2016	SDN	NSL-KDD	Deep neural network (DNN)	75.75
Ajaeiyah <i>et al.</i> [12]	2017	SDN	Custom	Random forest (RF)	85.4
Ye <i>et al.</i> [13]	2018	SDN	Custom	Support vector machine (SVM)	95.24
Latah and Tokar [14]	2018	SDN	NSL-KDD	Decision tree	71
Tang <i>et al.</i> [15]	2019	SDN	NSL-KDD	Gate recurrent unit (GRU)- recurrent neural network (RNN)	89
Boppana <i>et al.</i> [16]	2019	SDN	NSL-KDD	RF	81.95
Hannache and Batouche [17]	2020	SDN	Custom	DNN	96.13
Lim <i>et al.</i> [18]	2020	SDN-IoT	N. A.	FL-RF with actor-critic PPO	N.A
ElSayed <i>et al.</i> [19]	2021	SDN	InSDN [20]	Convolutional neural networks (CNN)+RF	99.28
Hadem <i>et al.</i> [21]	2021	SDN	NSL-KDD	SVM	95.98
Alzahrani and Alenazi [22]	2021	SDN	NSL-KDD	XGBoost	Detection: 95.5, Classification: 95.95
Wani <i>et al.</i> [23]	2021	SDN	CSE-CIC-IDS 2018	IDS IoT-SDL	99.05
Mohsin and Hamad [24]	2022	SDN	Custom	RF KNN Naive Bayes (NB) Logistic regression (LR)	RF: 100 KNN: 99.99-100 NB: 72.11-83.5 LR: 59.44-92.74
Ravi <i>et al.</i> [25]	2022	SDN-IoT	SDN-IoT [26]	GRU feature fusion	Detection: 99 Classification: 98
Jose and Jose [27]	2023	IoT	CIC-IDS 2017	DNN; LSTM; CNN	94.61; 97.67; 98.61
Logeswari <i>et al.</i> [28]	2023	SDN	NSL-KDD	HFS-LGBM	98.72
Chaganti <i>et al.</i> [29]	2023	SDN-IoT	SDN IoT-focused	LSTM	97.1
Maddu and Rao [30]	2023	SDN	InSDN edge IoT	DL	99.65
Elsayed <i>et al.</i> [31]	2023	SDN-IoT	ToN-IoT InSDN	LSTM	96.35; 99.73
Vidhya and Nagarajan [32]	2024	SDN-IoT	CSE-CIC-IDS2018; SDN-IoT	BiLSTM-based WNIDS	99.97-99.96 95.13-92.90
Niknami and Wu [33]	2024	SDN	NSL-KDD; KDD99	DeepIDPS (CNN-LSTM+AM)	92.2-95.4; 95.26-97.42
Our work	2024	SDN-IoT	IoT-SDN [34]	FL-DL	99.89

Ajaeiya *et al.* [12] introduced RF based IDS for identifying network threats in SDN. The network features used to train the model and predict network attacks consisted of tuple-5, packet count, byte count, and packet interarrival time. The detection method was tested against many types of attacks, including brute forcing, port scanning, and flooding attacks. While the results indicated a high level of accuracy in detecting attacks using the RF method, there is a lack of detailed information on the selection of the dataset for attack traffic. Therefore, these results may be solely relevant to non-IoT traffic.

Ye *et al.* [13] introduced a distributed denial of service (DDoS) attack detection system in SDN that utilized SVM. The feature set used for predicting flooding attacks consisted of the 6-tuple network flow characteristics. The authors state that they achieved an average detection accuracy rate of 95.24% in detecting user datagram protocol (UDP) flooding attacks. However, the attack traffic created with the hping3 tool is not suitable for generating IoT traffic.

Latah and Toker [14] conducted a comparison of several supervised ML methods for anomaly-based intrusion detection in SDNs. The authors stated that the decision tree algorithm obtained a higher accuracy of 99.7% when the network security laboratory (NSL)-KDD dataset characteristics were utilized as input for comparing ML detection models. However, the distinctive characteristics of SDN for detecting anomalies should be taken into account. Nevertheless, the NSL-KDD dataset was specifically created to assess and identify traditional network traffic, rather than focusing on the capabilities of SDN.

Boppana *et al.* [16] conducted a comparison of ML algorithms using various feature selection methods in the SDN anomaly detection module. The NSL-KDD dataset was utilized to assess the effectiveness of various feature and ML model combinations in the context of SDN. However, the authors acknowledge that conducting tests on a real-time SDN testbed is a potential future goal to verify the validity of their findings.

Hadem *et al.* [21] utilized an SVM and selective logging with IP traceback to accurately identify attacks in SDN using an IDS which also helped conserve memory resources. The NSL-KDD dataset utilized yielded a detection accuracy of 87.74%. However, the dataset is not sourced from non-IoT networks, and there is still potential for enhancing accuracy.

Alzaharani and Alenazi [22] presented a NIDS for SDNs that uses the extreme gradient boosting (XGBoost) model to accurately categorize network intrusions. Five features were chosen from 41 in the NSL-KDD dataset. The given findings indicate a classification accuracy of 95.5% for XGBoost. Additionally, the authors emphasize that their approach may be used for SDN.

Mohsin and Hamad [24] investigated the effectiveness of various supervised ML algorithms for detecting DDoS attacks across different SDN network topologies. They applied RF, k-nearest neighbors (KNN), NB, and LR to single, linear, and multi-controller architectures. Their results showed that while RF and KNN achieved strong detection performance, NB and LR suffered from low accuracy and a high rate of false predictions, limiting their suitability for practical deployment.

Jose and Jose [27] investigated the efficacy of DNN, convolutional neural networks (CNN), and long short-term memory networks (LSTM) in IoT environments for the deployment of IDS utilizing the CIC-IDS 2017 dataset. The results indicated that DL models outperformed previous methods used in IoT-based IDS. Specifically, LSTM and CNN achieved accuracies of 97.67% and 98.61%, respectively, while the overall DL approach reached 94.61% accuracy.

The aforementioned studies together either imitate the behavior of conventional network traffic or employ the attributes of previous network traffic data to perform testing. The experiments confirm that it is possible to integrate such enhancements into the module that is in charge of detecting attacks in the SDN controller. However, IoT network traffic should be considered, as it is produced through the utilization of IoT devices inside the SDN framework, or by combining the flow of IoT traffic with conventional network traffic to evaluate the detection effectiveness of ML models. Furthermore, the performance of detecting or classifying in supervised or unsupervised ML models still needs enhancement in the SDN network. Nevertheless, some researchers investigated further the utilization of neural network models for the detection and categorization of network attacks in SDN such as:

Chaganti *et al.* [29] an LSTM-based architecture for intrusion detection in SDN-enabled IoT networks. Their model effectively identified and classified various network attacks, including port scanning, operating system fingerprinting, denial of service (DoS), and DDoS. The results highlight the model's suitability for capturing temporal patterns and enhancing detection accuracy in complex SDN-IoT environments.

Elsayed *et al.* [31] conducted a secured automatic two-level intrusion detection system (SATIDS) that employed an enhanced LSTM network and utilized ToN-IoT and InSDN datasets. The author stated that the proposed system effectively distinguished between malicious and harmless network traffic, accurately categorized the type of attack, and precisely identified the specific sub-attack. The research results demonstrated that the suggested system surpasses others in identifying a wide range of attacks. However, LSTM-based models need substantial memory capacity throughout the training process. The substantial

memory resource consumption might restrict the utilization of LSTM for IDS in SDN and IoT networks. Also, in a complex IoT network, the suggested architecture requires significant time to train the model due to the process of self-learning the features and adjusting the model weights.

3. TECHNOLOGY BACKGROUND

Before examining the proposed model, it is essential to get an understanding of the main technique and method utilized in this study. Which was selected through an evaluation of the prior studies that consider the development of an efficient IDS system and analyze the used tools. This section provides an overview of the technologies and methodologies utilized to implement NIDS on an SDN network as follows.

3.1. Software defined network

Switches and routers were utilized in traditional networks to establish network connections and facilitate the transmission of data throughout the network. This networking technique may be vulnerable to a lack of confidentiality and susceptible to third-party attacks. SDN is a networking strategy that enhances the efficiency of a centralized environment by separating data transfer from dedicated devices [35]. This paradigm is structured around distinct planes, each with its own designated functions, i) data plane responsible for the forwarding of packets; ii) the control plane determines routing by leveraging a flow table that provides rules for efficiently managing incoming packets; and The application plan contains a range of services that are offered to users.

However, new vulnerabilities may also be introduced from this separation. For example, the controller can be illustrated by exhausting the communication bandwidth between infrastructure layers such as the OpenFlow switch and SDN controller. Nevertheless, SDN can improve network security due to its programming capabilities that enable the creation of security applications such as IDS that detect network threats. Also, it is important to mention that flow rules may be modified based on requirements [36] by leveraging the ability to program and control offered by SDN in comparison to traditional networking systems [37]. Figure 1 illustrates the typical SDN architecture.

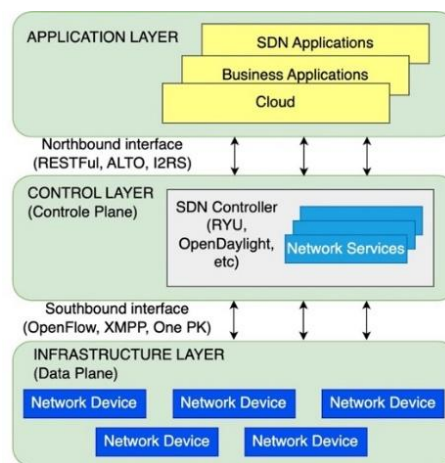


Figure 1. SDN network architecture

3.2. Intrusion detection systems

IDS is a crucial element in safeguarding systems by detecting and analyzing network traffic to identify security breaches and threats using one of the following techniques: signature-based or anomaly-based. The first method relies on predetermined network patterns and is therefore unable to identify new attacks. In contrast, the latter method analyzes particular characteristics of network traffic, allowing any divergence from normal network activity to be recognized as a potential attack; a simple comparison between them is presented in Table 2, [38]. Nevertheless, some drawbacks were also introduced, such as the lack of identification of encrypted packets and, the incidence of false alarms may be elevated, leading to the need for human intervention to adjust the anomaly indicators and ultimately resulting in an inefficient security solution [39].

Table 2. Detection technique comparison

Factor	Detection technique	
	Signature	Anomaly
Alarm rate	Low	High
Speed	High	Low
Flexibility	Low	High
Reliability	High	Moderate
Scalability	Low	High
Robustness	Low	High

Recently, several ML approaches have been introduced to identify intrusion in SDN and IoT [15], [16], [19]-[21]. Also, the DL model was proposed in the context of SDN and the IoT to enhance intrusion attack detection [17], [22], [25]. Previous research on intrusion detection has demonstrated that the DL model provides superior performance when applied to large-scale network datasets [22], [25]. Despite the substantial impact of ML and DL on practical problem-solving, they are subject to many limitations, including: i) users must provide their data to a centralized server to train the model; ii) when network size increases, the performance diminishes and there is a risk of a single point of failure that might undermine the integrity and quality of services (QoS); iii) IDS needs rapid analysis, however, centralized processing is a time-consuming process; and iv) IoT devices frequently gather data from end-users, potentially exposing their sensitive information. To tackle these problems, it's necessary to use methods that involve on-device learning.

3.3. Federated learning

Google introduced the concept of FL to preserve data privacy on devices [5]–[7] by allowing nodes to learn collaboratively without sharing data with a centralized server. FL is an iterative procedure in which the entire model is enhanced in each round until a specific number has been reached or the required level of performance is attained. In the beginning, the FL server selects a distinct group of clients to participate in the training process and distributes its global model to them [7]. Once the global model is obtained, each client employs its data for local training and transmits their acquired parameters back to the server, as illustrated in Figure 2. It offers a privacy protection technique that efficiently utilizes the processing resources of the parity device for model training, thereby preventing the leakage of private information during data transfer. Considering the enormous number of devices, there are a large number of relevant dataset resources that can be effectively utilized.

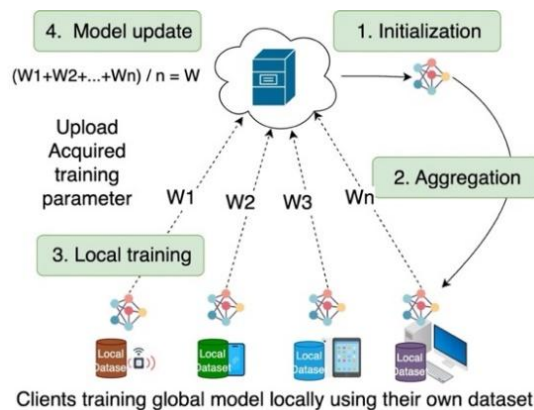


Figure 2. FL overview

Generally, FL may be categorized into three types based on the distribution of clients' data: vertical, horizontal, and transfer FL. Herts for learning (HFL) is an FL technique in which the datasets on the clients share the same feature but have separate observations. Vertical federated learning (VFL), often referred to as features-based FL, in which data from several domains is utilized to train a global model. In this context, the client dataset may contain identical observations but with varying characteristics. Aside from HFL and VFL, there is also the federated transfer learning (FTL) architecture presented in [40], which is applicable when the datasets on the devices differ not only in occurrences but also in characteristics. However, to ensure privacy, some problems must be addressed in the implementation of FL: i) it is imperative to guarantee that the

training model used does not disclose users' confidential information; ii) since the training process proceeds locally at each entity using its dataset. Therefore, it is crucial to guarantee that only allowed entities participate in the training process and the received model updates have been transmitted by them; iii) traditional ML models require a substantial amount of data to achieve outstanding performance. However, in a dispersed context, the accessible data on each device is minimal. Conversely, consolidating all data in a centralized way might lead to significant costs; and iv) the data stored on such devices may not exhibit distinct and symmetrical distribution (non-IID) characteristics; training these data sets poses a substantial challenge.

3.4. IoT-SDN dataset

There is a lack of publicly accessible datasets that are explicitly designed for intrusion detection in SDN-based IoT. For this work, a custom-generated dataset was utilized. The dataset comprises eighty-six attributes within a size of (2.7 GB) collected from simulated SDN-based IoT networks within two flow profiles: normal and attack traffic such as botnet, brute force, DoS, DDoS, exploitation, malware, MIRAI, probe, R2L, UR2, web-based, spoofing, and recon, employed using Metasploit. Table 3 presents the collected traffic categories together with their corresponding record numbers. The network topology is implemented using Mininet WiFi on the Ubuntu 20.04 LTS operating system consent of two Ryu controllers who were responsible for managing the operation of the four OpenFlow switches that connected to four subdomains. Each subdomain comprises a pair of hosts a single access point, and three wireless stations. The first two subdomains encompass a variety of services, such as HTTP and FTP servers. In contrast, the last two comprise many wireless sensor devices. The network traffic is captured using Wireshark and classified according to its features extracted using CICFlowMeter.

Table 3. Data records number for each traffic group

Group	Traffic type	Records
Normal	HTTPS, HTTP, FTP, DNS, mail, browsing, and YouTube	367,396
Attack	DoS, DDoS, R2L, Brute-Force, Exploitation, Web-Based, Botnet Probe, Recon, Spoofing, Malware	5,878,336 (367,396 for each)

4. PROPOSED METHODOLOGY

The technique employs a systematic approach that starts with the precise definition of the research issue. The combination of FL with DL techniques for anomaly incursion detection in SDN-based IoT networks is emerging as a potentially unique approach. This section delineates the projected architecture illustrated in Figure 3 which has been executed in two principal phases as follows.

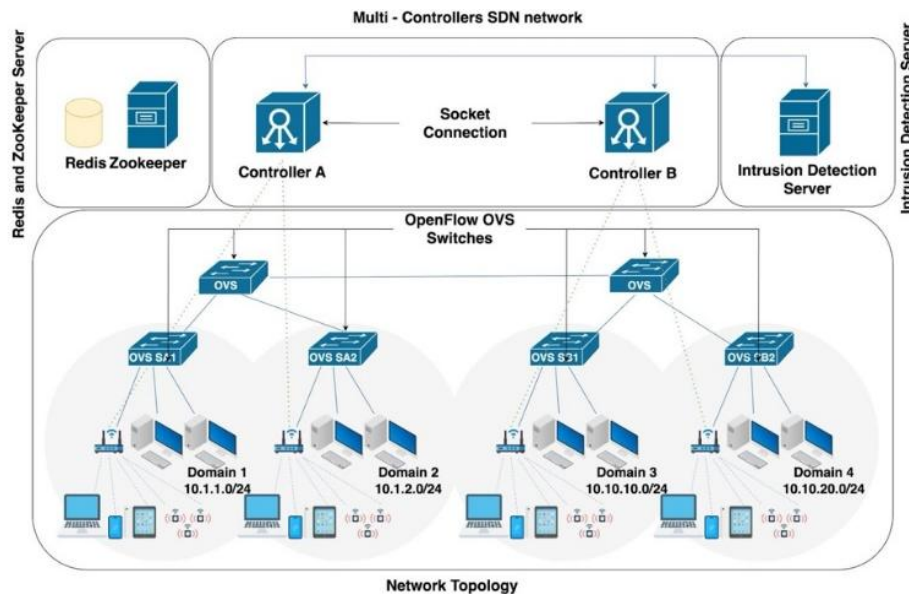


Figure 3. SDN-based IoT proposed system architecture

4.1. Deploying an SDN infrastructure for an IoT network

One of the significant vulnerabilities is the SDN network when the controller is exploited by overwhelming the communication capacity with excessive and undesired traffic, leading to a DDoS attack. However, network security can be enhanced by its programming capabilities, which allow for the development of security applications such as IDS that can identify network threats. Our suggested architecture comprises numerous objectives, which are i) a multi-controller SDN network was established utilizing Zookeeper and Redis. ZooKeeper will promptly organize and coordinate the change of controller role, whereas a backup copy of the flow table will be stored in Redis storage. This setup serves as a robust framework to prevent network failure. If the master controller becomes inactive, the other controller retrieves the flow entries from Redis storage and smoothly continues network operations, Figure 4 demonstrates the multi-controller implementation steps; ii) ingress and egress policies were employed to manage and control network traffic; iii) all packets received by the controller will be initially sent to the IDS server to predict whether the traffic received is an attack or normal traffic. However, by flooding the controller, the IDS server will also be flooded. To prevent this, DoS and DDoS attacks are mitigated once a threshold is reached; and iv) the slave controller is utilized to efficiently manage the huge amount of data received on the master controller by enabling Pushback police. Mininet WiFi used to construct a tree topology illustrated in Figure 4 consists of four domains. Each domain is composed of two hosts, an access point, three stations, and three wireless sensors. IoT devices may experience communication resource limits that prevent them from interacting with a central base station due to the limitations in communication resources.

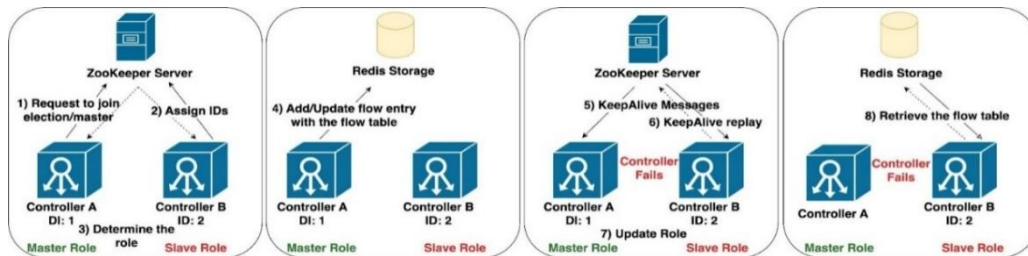


Figure 4. ZooKeeper and Redis coordination system

4.2. Deploying anomaly-based network intrusion detection system

Initially, the coordinator server uploads the global model presented in Table 4, to the IPFS network which is utilized to enhance secure model aggregation, ensuring that only authorized clients involved in the training process can access and download the global model based on a specific hash identifier. IPFS is a decentralized framework involving protocols, packages, and composable specifically designed to handle, direct, and transmit content-addressed data. The system is both resource-efficient and reliably converges to centralized FL frameworks with a drop of less than 1% [41]. The model consists of four dense layers with 256, 128, 64, and 32 neurons, respectively in addition to the input and output layers. Each layer is followed by a batch normalization layer and a dropout rate of 0.5. The reason for adding these layers is to improve the performance and generalization of the network. The batch normalization stabilizes the learning process by normalizing the activations of the preceding layer. The dropout layer is utilized to mitigate overfitting in the model by enhancing its ability to generalize to new data and increasing its overall resilience. The rectified linear unit (ReLU) activation function was used in all the Dense layers due to its simplicity, efficiency, and ability to address the vanishing gradient problem. SoftMax activation function was used in the output layer for multi-class classification tasks to generate a probability distribution across different classes.

Table 4. DL training model

Algorithm	Layers	Neuron
DNN	4 Dense, in addition to input and output layer 4 Batch Normalization layer 4 Dropout layer	256, 128, 64, 32
Activation function	ReLU, SoftMax	
Loss function	Categorical cross-entropy	
Optimizer	Adam	
Batch-size	256	

Table 5 illustrates the classification report of detecting each attack type after training the model. All of the metrics demonstrate a superior degree of effectiveness in all types of traffic, with a notable exception

being the user to root (U2R) category which can be explained by the fact that other categories frequently display more dissimilarity in comparison to normal traffic patterns. In contrast, the U2R attacking class has a notable similarity to the standard data traffic.

The IDS model is deployed on a dedicated server to perform intrusion detection to the complete network, which offers substantial advantages in terms of performance, scalability, security, and maintenance. It guarantees that the controller can concentrate on its primary operations, while the IDS server is fine-tuned and expanded expressly for efficient intrusion detection. In an SDN, the process of packet forwarding is handled differently, when a host sends a request to another host, it is first forwarded to the Open vSwitch (OVS) switch. The switch checks if there is any instruction to proceed with. If not, the packet is forwarded to the controller to identify the optimal path. In our work, the master controller sends the packet to the NIDS server, which contains the trained model. This model predicts whether the packet is normal or intrusion. In the case of normal traffic, the controller identifies the optimal path to forward it to the destination, sends the instruction back to the switch, and adds a new flow entry in Redis storage. If it is an intrusion traffic the controller adds a flow entry to block the source host.

Table 5. DNN training model

Traffic Type	Precision	Recall	F1-score
Botnet	0.9984	0.9936	0.9960
Brute-Force	0.9993	0.9815	0.9903
DDoS-ICMP	1.0	1.0	1.0
DDoS-UDP	0.9991	0.9994	0.9992
DoS-SYN	1.0	1.0	1.0
DoS-UDP	1.0	0.9998	0.9999
Exploitation	0.9936	0.9975	0.9956
Malware	0.9987	0.9944	0.9966
Mirai	0.9999	0.9989	0.9994
Normal	0.9994	0.9975	0.9985
Probe	1.0	0.9991	0.9995
R2L-IMAP	1.0	1.0	1.0
Recon-PingSweep	0.9936	0.9984	0.9960
Recon-Sniffing	0.9983	1.0	0.9991
Spoofing	1.0	1.0	1.0
U2R	0.9893	1.0	0.9946
Web-Attack	0.9908	1.0	0.9953

5. RESULTS AND DISCUSSION

The systematic review presents a thorough study of many research sources to assess and synthesize information about federated DL anomaly intrusion detection in SDN-based IoT. The findings collected indicate a variety of techniques and practices in the execution of the proposed methodology. This section explains the research findings, which offer a summary of the present study.

5.1. Statistical metrics

Several performance indicators have been defined for the multi-class confusion matrix to evaluate the effectiveness of the model. The multi-class confusion matrix is an $N \times N$ matrix, where N represents the number of unique class labels (C_0, C_1, \dots, C_N). Matrix cells are determined by the output consisting of the predicted label, which may be either positive or negative, that comes out of comparing the predicted label with the actual class label, which can be either normal or attack [42]. As a result, the traditional classification of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) cases becomes irrelevant. Alternatively, a more appropriate approach entails focusing on certain classes. This technique allows for the formulation of class-specific metrics. By adeptly merging these measurements that are distinct to each class, a comprehensive collection of metrics for the whole confusion matrix can be obtained, as exemplified in (1) to (5) [43].

$$Accuracy = \frac{\sum_{i=1}^N TP(C_i)}{\sum_{i=1}^N \sum_{j=1}^N C_{i,j}} \quad (1)$$

$$F1(C_i) = \frac{2TPR(C_i)PPV(C_i)}{TPR(C_i)+PPV(C_i)} \quad (2)$$

$$TPR(C_i) = \frac{TP(C_i)}{TP(C_i)+FN(C_i)} \quad (3)$$

$$F\beta(C_i) = (1 + \beta^2) \frac{TPR(C_i)PPV(C_i)}{\beta^2TPR(C_i)+PPV(C_i)} \tag{4}$$

$$PPV(C_i) = \frac{TP(C_i)}{TP(C_i)+FN(C_i)} \tag{5}$$

5.2. Experimental results

The implementation of the DL model utilized TensorFlow, Keras, and Scikit-Learn as the underlying technology and was executed under the graphics processing unit (GPU) T4 x2 environment. The process of FL was examined for 20 rounds the evaluation parameters for each round are illustrated in Table 6. In the first round, only one client was used which observed a high detection loss, reaching 2.8321 and an accuracy of 0.0841. This can be attributed to the limited diversity and insufficient training data. Comparatively, running the model using three clients for just one round, decreased the loss to 0.0987, and increased the accuracy to 0.9749. For both scenarios, the tests were conducted for 50 epochs, with a batch size of 250 and Adam optimizer due to its adaptive learning rate features and durability. After completing the 20 training rounds, there is a significant enhancement in accuracy, rising from 99.76 to 99.89%. In addition, a notable reduction demonstrated in loss decreased from 0.01 in the standard centralized training procedure to 0.005 in the federated DL scenario. Figure 5 illustrates the results of each cycle, with Figure 5(a) showing an improvement in enhancement and Figure 5(b) indicating a loss reduction.

Table 6. Federated DL training results

Round	Client 1		Client 2		Client 3		FedAvg	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
1	0.9745	0.0986	0.9751	0.0984	0.9750	0.0984	0.9749	0.0987
2	0.9964	0.0131	0.9955	0.0142	0.9959	0.0136	0.9959	0.0139
3	0.9980	0.0093	0.9979	0.0093	0.9980	0.0093	0.9978	0.0094
4	0.9982	0.0090	0.9981	0.0089	0.9981	0.0090	0.9980	0.0091
5	0.9981	0.0089	0.9981	0.0086	0.9981	0.0088	0.9980	0.0088
6	0.9984	0.0078	0.9984	0.0077	0.9984	0.0078	0.9983	0.0079
7	0.9983	0.0079	0.9984	0.0077	0.9983	0.0079	0.9982	0.0079
8	0.9987	0.0068	0.9987	0.0068	0.9987	0.0069	0.9986	0.0069
9	0.9984	0.0073	0.9984	0.0071	0.9984	0.0072	0.9983	0.0073
10	0.9987	0.0069	0.9987	0.0069	0.9987	0.0070	0.9986	0.0070
11	0.9988	0.0064	0.9988	0.0064	0.9988	0.0064	0.9987	0.0065
12	0.9987	0.0066	0.9987	0.0066	0.9987	0.0066	0.9986	0.0067
13	0.9988	0.0060	0.9988	0.0060	0.9988	0.0060	0.9987	0.0061
14	0.9987	0.0067	0.9987	0.0068	0.9987	0.0068	0.9986	0.0069
15	0.9988	0.0060	0.9989	0.0061	0.9988	0.0061	0.9988	0.0061
16	0.9989	0.0057	0.9989	0.0057	0.9989	0.0058	0.9988	0.0058
17	0.9989	0.0056	0.9989	0.0056	0.9989	0.0057	0.9988	0.0057
18	0.9989	0.0059	0.9989	0.0059	0.9989	0.0059	0.9988	0.0060
19	0.9989	0.0059	0.9989	0.0059	0.9989	0.0059	0.9988	0.0060
20	0.9989	0.0057	0.9989	0.0057	0.9989	0.0057	0.9988	0.0057

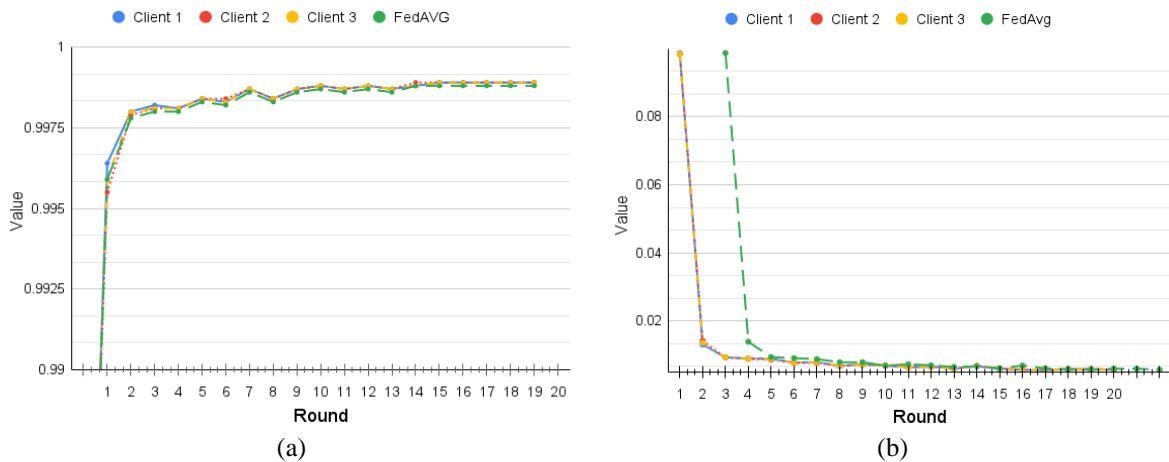


Figure 5. Client training metrics results for (a) model accuracy metric for the clients and FedAvg and (b) model loss metric for the clients and FedAvg

6. CONCLUSION

This work presents a federated DL for NIDS in an IoT context in a multi-controller SDN network, using IPFS as the underlying technology and the AES encryption algorithm to help improve the security of the aggregation and training. A custom-generated dataset of intra- and inter-attacks was utilized to extract internal feature representations to detect and classify attacks. The proposed architecture successfully mitigates DoS and DDoS attacks once the attack threshold is reached on the controller to avoid flooding the IDS server, where the suggested model possesses an accuracy of 99.89% in identifying several attack types demonstrated superior performance in both the detection and classification of attacks using FL, surpassing conventional DL for the same model. The result shows a drop in loss from 0.01 in the standard centralized training procedure that utilizes the DL model to 0.005 in the federated DL scenario. In addition, there is a significant enhancement in accuracy, rising from 99.76 to 99.89%. The suggested technique can apply to a wide range of situations and may be included as a component in a real-time SDN-IoT environment. Its purpose is to detect any attacks and classify them into certain types, causing an alarm. The current work is suboptimal. Instead of using a method that selects all the characteristics, it would be more effective to apply kernel-based methods to choose the ideal features. This may significantly enhance the effectiveness of the SDN-IoT IDS. In addition, doing a thorough examination and evaluation of the model and another model inside the environment is crucial, as the majority of ML and DL models are susceptible to adversarial attacks.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Heba Dhirar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Ali H. Hamad		✓			✓	✓	✓			✓		✓	✓	

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : **O**riting - **O**riginal Draft

E : **E**riting - **R**eview & **E**ditting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest related to this work.

DATA AVAILABILITY

The data that support the findings of this study are openly available on Kaggle at <https://www.kaggle.com/datasets/hebadhirar/sdn-iot>, under the title SDN-IoT Intrusion Detection Dataset.

REFERENCES




- [1] A. Alrawais, A. Althothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, Mar. 2017, doi: 10.1109/MIC.2017.37.
- [2] L. Liu, B. Xu, X. Zhang, and X. Wu, "An intrusion detection method for internet of things based on suppressed fuzzy clustering," *Eurasip Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018, doi: 10.1186/s13638-018-1128-z.
- [3] H. Qushtom and K. Rabaya'h, "Enhancing the QoS of IoT networks with lightweight security protocol using Contiki OS," *International Journal of Computer Network and Information Security*, vol. 9, no. 11, pp. 27–35, 2017, doi: 10.5815/ijenis.2017.11.03.
- [4] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 205–216, 2015, doi: 10.14257/ijisa.2015.9.5.21.
- [5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," *arXiv-Computer Science*, pp. 1–10, 2016.
- [6] P. Boobalan *et al.*, "Fusion of federated learning and industrial internet of things: a survey," *Computer Networks*, vol. 212, 2022, doi: 10.1016/j.comnet.2022.109048.

- [7] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021, doi: 10.1016/j.future.2020.10.007.
- [8] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, 2021, doi: 10.1016/j.knsys.2021.106775.
- [9] Á. L. V. Caraguay, A. B. Peral, L. I. B. López, and L. J. G. Villalba, "SDN: evolution and opportunities in the development IoT applications," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014, doi: 10.1155/2014/735142.
- [10] F. Kiani, "A survey on management frameworks and open challenges in IoT," *Wireless Communications and Mobile Computing*, vol. 2018, 2018, doi: 10.1155/2018/9857026.
- [11] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," *2016 International Conference on Wireless Networks and Mobile Communications, WINCOM 2016: Green Communications and Networking*, pp. 258–263, 2016, doi: 10.1109/WINCOM.2016.7777224.
- [12] G. A. Ajaeiy, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based Intrusion detection system for SDN," *IEEE Symposium on Computers and Communications*, pp. 787–793, 2017, doi: 10.1109/ISCC.2017.8024623.
- [13] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, 2018, doi: 10.1155/2018/9804061.
- [14] M. Latah and L. Toker, "Towards an efficient anomaly-based intrusion detection for software-defined networks," *IET Networks*, vol. 7, no. 6, pp. 453–459, 2018, doi: 10.1049/iet-net.2018.5080.
- [15] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in sdn-based networks: Deep recurrent neural network approach," *Advanced Sciences and Technologies for Security Applications*, pp. 175–195, 2019, doi: 10.1007/978-3-030-13057-2_8.
- [16] R. V. Boppana, R. Chaganti, and V. Vedula, "Analyzing the vulnerabilities introduced by ddos mitigation techniques for software-defined networks," *Advances in Intelligent Systems and Computing*, vol. 1055, pp. 169–184, 2020, doi: 10.1007/978-3-030-31239-8_14.
- [17] O. Hannache and M. C. Batouche, "Neural network-based approach for detection and mitigation of DDoS attacks in SDN environments," *International Journal of Information Security and Privacy*, vol. 14, no. 3, pp. 50–71, 2020, doi: 10.4018/IJISP.2020070104.
- [18] H. K. Lim, J. B. Kim, S. Y. Kim, and Y. H. Han, "Federated reinforcement learning for automatic control in SDN-based IoT environments," *International Conference on ICT Convergence*, pp. 1868–1873, 2020, doi: 10.1109/ICTC49870.2020.9289245.
- [19] M. S. ElSayed, N. A. Le-Khac, M. A. Albahar, and A. Jurcut, "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique," *Journal of Network and Computer Applications*, vol. 191, 2021, doi: 10.1016/j.jnca.2021.103160.
- [20] M. S. Elsayed, N. A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263–165284, 2020, doi: 10.1109/ACCESS.2020.3022633.
- [21] P. Hadem, D. K. Saikia, and S. Moulik, "An SDN-based intrusion detection system using SVM with selective logging for IP traceback," *Computer Networks*, vol. 191, 2021, doi: 10.1016/j.comnet.2021.108015.
- [22] A. O. Alzahrani and M. J. F. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, 2021, doi: 10.3390/fi13050111.
- [23] A. Wani, S. Revathi, and R. Khaliq, "SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL)," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 3, pp. 281–290, 2021, doi: 10.1049/cit2.12003.
- [24] M. A. Mohsin and A. H. Hamad, "Performance evaluation of SDN DDoS attack detection and mitigation based random forest and K-nearest neighbors machine learning algorithms," *Revue d'Intelligence Artificielle*, vol. 36, no. 2, pp. 233–240, 2022, doi: 10.18280/ria.360207.
- [25] V. Ravi, R. Chaganti, and M. Alazab, "Deep learning feature fusion approach for an intrusion detection system in SDN-based IoT networks," *IEEE Internet of Things Magazine*, vol. 5, no. 2, pp. 24–29, 2022, doi: 10.1109/IOTM.003.2200001.
- [26] A. K. Sarica and P. Angin, "A novel SDN dataset for intrusion detection in IoT networks," *2020 16th International Conference on Network and Service Management (CNSM)*, Izmir, Turkey, 2020, pp. 1–5, doi: 10.23919/CNSM50824.2020.9269042.
- [27] J. Jose and D. V. Jose, "Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 1, pp. 1134–1141, 2023, doi: 10.11591/ijece.v13i1.pp1134-1141.
- [28] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for SDN using machine learning," *Intelligent Automation and Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023, doi: 10.32604/iasc.2023.026769.
- [29] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep learning approach for SDN-enabled intrusion detection system in IoT networks," *Information*, vol. 14, no. 1, 2023, doi: 10.3390/info14010041.
- [30] M. Maddu and Y. N. Rao, "Network intrusion detection and mitigation in SDN using deep learning models," *International Journal of Information Security*, vol. 23, no. 2, pp. 849–862, 2024, doi: 10.1007/s10207-023-00771-2.
- [31] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, 2023, doi: 10.1016/j.asej.2023.102211.
- [32] G. S. Vidhya and R. Nagarajan, "A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network," *Computing*, vol. 106, no. 8, pp. 2613–2642, Aug. 2024, doi: 10.1007/s00607-024-01295-w.
- [33] N. Niknami and J. Wu, "DeepIDPS: an adaptive DRL-based intrusion detection and prevention system for SDN," in *ICC 2024 - IEEE International Conference on Communications*, Denver, USA, 2024, pp. 2040–2046, doi: 10.1109/ICC51166.2024.10622849.
- [34] H. Dhirar, "IoT-SDN IDS dataset 2024," *Kaggle*, 2024. Accessed: Sep. 07, 2024. [Online]. Available: <https://www.kaggle.com/datasets/hebadhirar/iot-sdn-ids-dataset>
- [35] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015, doi: 10.1109/JPROC.2014.2371999.
- [36] S. T. Ali, V. Sivaraman, A. Radford, and S. Jha, "A survey of securing networks using software defined networking," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 1086–1097, 2015, doi: 10.1109/TR.2015.2421391.
- [37] P. Megyesi, A. Botta, G. Aceto, A. Pescapé, and S. Molnár, "Challenges and solution for measuring available bandwidth in software defined networks," *Computer Communications*, vol. 99, pp. 48–61, 2017, doi: 10.1016/j.comcom.2016.12.004.
- [38] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 493–501, Jan. 2019, doi: 10.1007/s12083-017-0630-0.
- [39] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, no. 1–2, pp. 18–28, 2009, doi: 10.1016/j.cose.2008.08.003.




- [40] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020, doi: 10.1109/MIS.2020.2988525.
- [41] C. Pappas, Di. Chatzopoulos, S. Lalis, and M. Vavalis, "IPLS: a framework for decentralized federated learning," *2021 IFIP Networking Conference, IFIP Networking 2021*, 2021, doi: 10.23919/IFIPNetworking52078.2021.9472790.
- [42] Padmanayana and D. A. Anoop, "Binary classification of DR-diabetic retinopathy using CNN with fundus colour images," *Materials Today: Proceedings*, vol. 58, pp. 212–216, 2022, doi: 10.1016/j.matpr.2022.01.466.
- [43] J. Opitz, "A closer look at classification evaluation metrics and a critical reflection of common evaluation practice," *Sustainability*, vol. 11, no. 1, pp. 1–14, Apr. 2024, doi: 10.1162/tacl_a_00675.

BIOGRAPHIES OF AUTHORS



Heba Dhirar    received her B.S. in information and communications engineering, from Al Khwarizmi College of Engineering, University of Baghdad, Iraq in 2019. Currently, she is a master's student in information and communication engineering at the University of Baghdad, Iraq. Her research interests include computer engineering, machine learning, and information security. She can be contacted at email: heba.d@kecbu.uobaghdad.edu.iq.



Ali H. Hamad    has been a faculty member at the Department of Information and Communication Engineering, University of Baghdad, since 2003. He received his Ph.D. in control and systems engineering from the University of Basrah, Iraq, in 2015. He obtained his M.Sc. and B.Sc. degrees from the University of Technology, Iraq, in 2000 and 1997, respectively. His research interests include the internet of things (IoT), machine learning, information security, wireless sensor networks, and blockchain technology. He can be contacted at email: ahamad@kecbu.uobaghdad.edu.iq.