Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices

Segun I. Popoola, Ruth Ande, Bamidele Adebisi, *Senior Member, IEEE,* Guan Gui, *Senior Member, IEEE,*
Mohammad Hammoudeh, *Senior Member, IEEE,* Olamide Jogunola, *Graduate Student Member, IEEE*

***Abstract*—Deep Learning (DL) has been widely proposed for botnet attack detection in Internet of Things (IoT) networks. However, the traditional Centralized DL (CDL) method cannot be used to detect previously unknown (zero-day) botnet attack without breaching the data privacy rights of the users. In this paper, we propose Federated Deep Learning (FDL) method for zero-day botnet attack detection to avoid data privacy leakage in IoT edge devices. In this method, an optimal Deep Neural Network (DNN) architecture is employed for network traffic classification. A model parameter server remotely coordinates the independent training of the DNN models in multiple IoT edge devices, while Federated Averaging (FedAvg) algorithm is used to aggregate local model updates. A global DNN model is produced after a number of communication rounds between the model parameter server and the IoT edge devices. Zero-day botnet attack scenarios in IoT edge devices is simulated with the Bot-IoT and N-BaIoT data sets. Experiment results show that FDL model: (a) detects zero-day botnet attacks with high classification performance; (b) guarantees data privacy and security; (c) has low communication overhead (d) requires low memory space for the storage of training data; and (e) has low network latency. Therefore, FDL method outperformed CDL, Localized DL, and Distributed DL methods in this application scenario.**

***Index Terms*—Cybersecurity, botnet detection, federated learning, deep learning, deep neural network, Internet of Things.**

I. INTRODUCTION

BOTNET attack is a serious cyber security challenge facing the Internet of Things (IoT) [1]–[3]. In our context, a botnet is a network of compromised devices that is used to launch cyber attack against critical infrastructures [4]. This cyber attack may be in form of Denial of Service (DoS), Distributed DoS (DDoS), reconnaissance, or data theft [5]. For example, *Mirai* is a popular IoT botnet, which can automatically scan a network for vulnerable devices (Scan attack) as well as launch acknowledgement (ACK), synchronization (SYN), and User Datagram Protocol (UDP) flooding attacks [6]. Meanwhile, more than 25.4 billion IoT devices will be connected to the Internet in 2030 [7], and IoT market is expected to worth about 1.6 trillion US dollars by 2025 [8]. There is an increasing attention of cybercriminals to IoT due to its fast-growing adoption in smart applications, distributed nature, market size, and security vulnerabilities.

IoT networks generate high volume of data, and this is expected to reach 79.4 zettabytes (ZB) by 2025 [9]. With the advent of cloud computing, each IoT device transmits its data to a central server on the Cloud, where different pre-processes and analyses can be performed on the aggregated data. In view of this, Centralized Deep Learning (CDL) method has been extensively proposed for network-based botnet attack detection in large IoT network traffic data with good classification performance [10]–[14]. For example, Apruzzese et al [15] proposed a method that can prevent adversarial attacks using Deep Reinforcement Learning (DRL). Also, Zhao et al [16] proposed a Lightweight Dynamic Autoencoder Network (LDAN) method for network intrusion detection in resource-constrained devices of Wireless Sensor Network (WSN). In previous works [10], [17]–[20], we proposed different Deep Learning (DL) methods, which can process a large volume of network traffic data to protect communication networks against cyber attacks. However, modern IoT networks are fast becoming highly scalable. Therefore, due to network constraints, it may be difficult to offload massive distributed IoT network traffic data to a remote central cloud server for data processing in real-life use cases. Also, CDL method takes longer time to train, it has high communication overhead, and its memory space requirement for data storage is high. Furthermore, Cloud data centers are often located far away from where IoT devices are deployed. This causes high latency in CDL-based botnet detection method.

Recently, strict laws such as the General Data Protection Regulation (GDPR)[1] and the Consumer Privacy Bill of Rights (CPBR)[2] were enacted to address data privacy concerns. Unfortunately, CDL method does not guarantee the privacy and security of IoT devices because it involves the transmission of network traffic features from all participating IoT devices to a central cloud server. Specifically, the use of a third-party cloud

Manuscript received February xx, 2021; revised June xx, 2021.

This work is supported in part by Cyraatek Ltd UK, the Faculty of Science & Engineering, Manchester Metropolitan University, and in part by ENERGY-IQ project, a UK-Canada Power Forward Smart Grid Demonstrator project funded by The Department for Business, Energy and Industrial Strategy (BEIS) under Grant 7454460; and the NICE (Nigerian Intelligent Clean Energy) Marketplace project funded by the Department for International Development (DFID).

S. I. Popoola, B. Adebisi and O. Jogunola are with the Department of Engineering, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester M1 5GD, United Kingdom. E-mail: s.popoola@mmu.ac.uk; b.adebisi@mmu.ac.uk; o.jogunola@mmu.ac.uk

R. Ande is with the Artificial Intelligence for Cybersecurity Research Team, Cyraatek Ltd., Manchester M5 3EZ, United Kingdom. E-mail: ruth@raait.com

G. Gui is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, 210003 China. E-mail: guiguan@njupt.edu.cn

M. Hammoudeh is with the Department of Computing and Mathematics, Manchester Metropolitan University, Manchester M1 5GD, United Kingdom. E-mail: m.hammoudeh@mmu.ac.uk

[1] https://gdpr.eu/data-privacy/

[2] https://www.congress.gov/bill/116th-congress/senate-bill/2968/text

server for CDL will introduce a high risk of privacy leakage in IoT systems because the network traffic features may contain sensitive information about the owners of the IoT devices [21]–[24]. The violation of data privacy protection regulations can lead to a serious penalty. In the case of GDPR, the fine of data privacy breach could be as high as €10 million[3].

Edge computing can be combined with DL to bring intelligence closer to where data are being generated, thereby addressing the issues of data privacy, high communication cost, large memory space requirement, short training time, and high latency [25]. Localized DL (LDL) and Distributed DL (DDL) methods achieve edge intelligence without data aggregation [26]. However, the classification performance of these methods is usually low in zero-day (previously unknown) botnet attack scenarios because a single IoT edge device has limited training samples. Meanwhile, in zero-day botnet attacks, hackers use a network of compromised computing devices to exploit previously unknown vulnerabilities in IoT systems. Detecting zero-day cyber attacks is a very difficult task because there is no prior knowledge of such incidence [27]. Therefore, an efficient intrusion detection system designed for IoT edge network must be able to detect zero-day botnet attacks with high detection rate and very low false alarm rate.

Federated Learning (FL) is a collaborative method for privacy-preserving DL based on the private data in distributed multiple devices [28]. This method enables collaborative DL in distributed IoT devices without sharing private network traffic data with the central cloud server. In this paper, we propose Federated DL (FDL) method for zero-day botnet attack detection in IoT edge devices. Previous studies have shown that DNN models can process network traffic data to detect botnet attacks in IoT networks [29]–[31]. Therefore, Deep Neural Network (DNN) architecture is employed for network traffic classification. Local DNN models are trained independently in multiple IoT edge devices, while Federated Averaging (FedAvg) algorithm is used to aggregate local model updates. A global DNN model is produced after a number of communication rounds between the model parameter server and the IoT edge devices. The main contributions of this paper are as follows:

1) We propose FDL method for zero-day botnet attack detection in IoT edge devices without any data privacy concern;
2) DNN is designed for network traffic classification while FedAvg algorithm is used for the aggregation of local DNN model updates;
3) FDL method is simulated with Bot-IoT dataset, and its classification performance in five IoT edge devices is evaluated based on accuracy, precision, recall, and F1 score;
4) The effectiveness of FDL method is compared with state-of-the-art DL methods i.e. CDL, LDL, and Distributed DL (DDL).

The remaining parts of this paper are organised as follows: in Section II, we present the review of related works and the main contributions of this paper; in Section III, we describe the proposed method for zero-day botnet attack detection in IoT edge devices; in Section IV, we develop and simulate DL models; in Section V, we analyse and compare the effectiveness of the DL models; and in Section VI, we summarise our findings.

II. Review of Related Works

Network Intrusion Detection System (NIDS) is often designed for specific use cases. In the literature, Federated Learning (FL) method has been proposed for intrusion detection in Wireless Edge Network (WEN) [32], [33], IoT [21]–[23], [34]–[39], Industrial IoT (IIoT) [24], [40]–[42], industrial Cyber-Physical System (CPS) [43], Medical CPS [44], Wireless Fidelity (Wi-Fi) network [45], large-scale distributed Local Area Network (LAN) [46], [47], satellite-terrestrial integrated networks [48], Cloud [49], edge computing [50], vehicular network [26], [51], [52]. We acknowledge that FL methods have been proposed for intrusion detection in IoT networks [21]–[23], [34]–[39]. However, the authors did not consider zero-day cyber-attack vulnerabilities in edge IoT devices. In real-life scenario, the training data on each IoT edge device is expected to have a unique statistical distribution depending on the usage pattern [53]. Therefore, the sizes of local training data in IoT edge devices should vary. In previous studies, the local training data in an IoT edge device is a representative of the overall data distribution. That is, the local training data are balanced, independent and identically distributed across the classes of network traffic investigated.

Different ML/DL model architectures have been proposed for network traffic classification in FL method. Chen *et al* [32] combined Gated Recurrent Unit (GRU) with Support Vector Machine (SVM). Rahman *et al* [21], Al-Athba *et al* [36], and Kim *et al* [23] used Artificial Neural Network (ANN). Li *et al* [43] combined Convolutional Neural Network (CNN) with GRU. Cetin *et al* [45] employed Stacked Autoencoder (SAE). Sun *et al* [46], [47] and Li *et al* [48] adopted CNN. Qin *et al* [33] proposed Binarised Neural Network (BNN). Nguyen *et al* [22] proposed GRU. Hei *et al* [49] compared the effectiveness of ANN, Decision Tree (DT), Random Forest (RF), and SVM. Zhao *et al* [29] proposed Deep Neural Network (DNN). Chen *et al* [54] proposed Deep Autoencoding Gaussian Mixture Model (DAGMM). Li *et al* [50] recommended a hybrid of CNN and Long Short-Term Memory (LSTM).

FL methods are simulated with relevant network intrusion dataset(s) to evaluate their performance. So far, FL methods have been simulated with WSN-DS [32], KDDCup99 [32], [49], [54], CICIDS2017 [32], [33], NSL-KDD [21], [23], [36], GPWST [43], AWID [45], ISCX2014 [33], UNSW-NB15 [29], [50], and private datasets [22], [46]–[48]. However, these network intrusion datasets did not contain samples of IoT network traffic. Also, different botnet attack scenarios were not included in the datasets. Table I summarises the review of related works on the application of FL methods to intrusion detection. In this paper, we addressed the aforementioned research gaps in the literature.

[3] https://gdpr-info.eu/issues/fines-penalties/

TABLE I
COMPARISON WITH RELATED WORKS

Ref.	Year	Model	Dataset	IoT traffic	Botnet attacks	Zero-day scenario	Main contribution
[32]	2020	GRU-SVM	WSN-DS, KDDCup99, CICIDS2017	✗	✗	✗	Chen *et al.* proposed the use of attention mechanism in FedAvg algorithm to reduce the communication overhead of FL in WEN while ensuring learning convergence.
[21]	2020	ANN	NSL-KDD	✗	✗	✗	Rahman *et al.* proposed FedAvg algorithm for intrusion detection in IoT networks.
[43]	2020	CNN-GRU	GPWST	✗	✗	✗	Li *et al.* proposed the use of Paillier cryptosystem-based secure communication protocol in FL-based intrusion detection system to preserve the security and the privacy of model parameters.
[45]	2019	SAE	AWID	✗	✗	✗	Cetin *et al.* investigated the effectiveness of FL approach for wireless intrusion detection in a Wi-Fi network.
[46], [47]	2020	CNN	Private	✗	✗	✗	Sun *et al.* proposed a segmented FL method for intrusion detection in a large-scale distributed LAN setting.
[36]	2020	ANN	NSL-KDD	✗	✗	✗	Al-Athba *et al.* proposed the use of mimic learning in FL method to prevent reverse engineering ML attacks.
[48]	2020	CNN	Private	✗	✗	✗	Li *et al.* proposed FL method for distributed network intrusion detection in satellite-terrestrial integrated networks.
[33]	2020	BNN	CICIDS2017, ISCX2014	✗	✗	✗	Qin *et al.* proposed FL method for line-speed and scalable intrusion detection at network edge
[22]	2019	GRU	Private	✗	✗	✗	Nguyen *et al.* proposed FL approach for device-type-specific anomaly detection in IoT networks.
[49]	2020	ANN, DT, RF, SVM	KDDCup99	✗	✗	✗	Hei *et al.* proposed a blockchained-FL method for cloud-based intrusion detection
[23]	2020	ANN	NSL-KDD	✗	✗	✗	Kim *et al.* proposed FL method for collaborative anomaly detection in IoT networks.
[29]	2020	DNN	UNSW-NB15	✗	✗	✗	Zhao *et al.* combined FL with transfer learning to address the problem of data scarcity in anomaly detection.
[54]	2020	DAGMM	KDDCup99	✗	✗	✗	Chen *et al.* proposed FL method for network anomaly detection.
[50]	2020	CNN-LSTM	UNSW-NB15	✗	✗	✗	Li *et al.* proposed FL method for APT detection in edge computing.
Ours	2021	DNN	Bot-IoT, N-BaIoT	✓	✓	✓	We propose FL method for zero-day botnet attack detection in edge IoT devices.

III. THE PROPOSED FDL METHOD

In this section, we describe the network traffic features, the DNN architecture, and the FDL algorithm that are proposed for zero-day botnet attack detection in IoT edge devices.

A. Deep Neural Network

For accurate classification of network traffic in IoT edge devices, we propose DNN architecture to obtain hierarchical representation using multiple layers of abstraction. This DL model architecture comprised an input layer, densely-connected hidden layers, and an output layer. The number of neurons at the input layer, d, is equal to the number of features that faithfully represents a single network traffic packet in the training data. The number of the densely-connected hidden layers and the number of neurons in each of them are mostly determined based on experimentation with different sets of values to obtain the optimal DNN optimal architecture that achieves the best classification performance. For the first hidden layer, each of the network traffic samples in the training data, x, is transformed into h_1 as:

$$h_1 = \sigma_h(W_1 x + b_1), \tag{1}$$

where σ_h is the activation function at the hidden layer, W_1 is the weight matrix of the first hidden layer, and b_1 is the bias vector of the first hidden layer. For any successive hidden layer, h_{i+1}, the output of the current hidden layer, h_i, is transformed as:

$$h_{i+1} = \sigma_h(W_i h_i + b_i), \tag{2}$$

where W_i is the weight matrix of the current hidden layer, and b_i is the bias vector of the current hidden layer. For each of the hidden layers, a Rectified Linear Unit (ReLU) activation function is used to transform the summed input because it is easier to train and it helps to achieve good classification performance [55]. The initial random weight matrices of the hidden layers are set based on the He uniform initialization method that was proposed for ReLU activation functions in [56]. Finally, the predicted class label, $\tilde{y}$, is obtained by transforming the output of the last hidden layer, h_j:

$$\tilde{y} = \sigma_y(h_j), \tag{3}$$

where σ_y is the activation function at the output layer. The number of neurons at the output layer, m, is equal to the number of classes that network traffic in the IoT network can be categorised into. A softmax activation function is used to normalize the output of h_j to a probability distribution over the predicted output classes as:

$$\sigma_y(h)_\alpha = \frac{e^{h_\alpha}}{\sum_{\beta=1}^{m} e^{h_\beta}}, \qquad (4)$$

where $\sigma_y(h)$ is the softmax function, h_α is the input hidden vector, e^{h_α} is the standard exponential function for the input hidden vector, and e^{h_β} is the standard exponential function for the output hidden vector. *Adam* algorithm that was proposed in [57] is used with categorical cross-entropy loss function for first-order gradient-based stochastic optimization.

B. Federated Deep Learning

FDL method is proposed to detect zero-day botnet attacks in IoT edge devices based on **Algorithm 1**. The FDL framework comprised of a model parameter server and K edge IoT devices. The model parameter server coordinates the training of DNN models in the edge IoT devices. It determines the number of training iterations/epochs (E), the batch size of training data (B), and the number of communication rounds (R). In this method, K DNN models are trained separately with local training data that are privately held in K edge IoT devices. After each training of E epochs, all the edge IoT devices send their local model updates to the model parameter server for aggregation using FedAvg algorithm [28]. Model aggregation is performed by model parameter server in R communication rounds.

Algorithm 1: FDL algorithm

```
Input: $R$, $E$, $N$, $B$, $K$
Initialization: $W = W_0$
Output: $W_r$
function localUpdate($W, k$):
    for $e = 1$ to $E$ do
        for $b = 1$ to $\frac{N}{B}$ do
            $W_{k,b} = W_{k,b-1} - \gamma \Delta L(b, W_k)$
        end
    end
    return $W_k$
end function
for $r = 1$ to $R$ do
    for $k = 1$ to $K$ do
        $W_{r,k}$ = localUpdate($W_{r-1}, k$)
    end
    $W_r = \sum_{k=1}^{K} \frac{n_k}{N} W_{r,k}$
end
```

IV. Model Development and Experiments

In this section, we simulate the CDL, LDL, DDL, and FDL methods with the Bot-IoT and N-BaIoT data sets. Experiments were performed to determine the effectiveness of these methods for zero-day botnet attack detection in five IoT edge devices, as shown in Fig. 1.

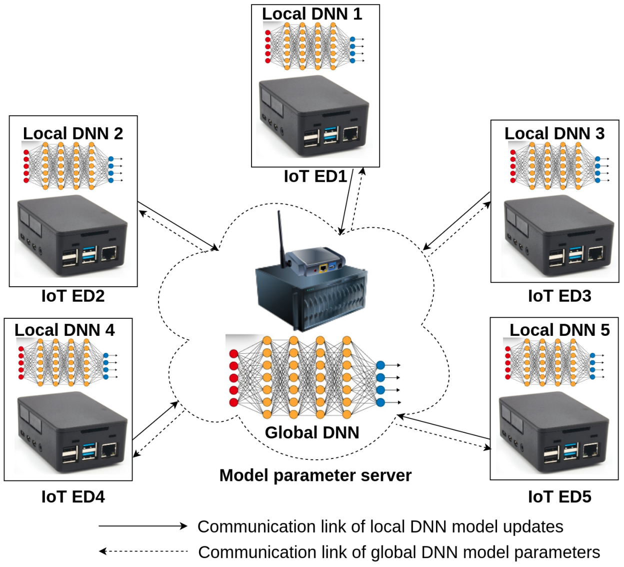

Fig. 1. FDL architecture for zero-day botnet attack detection in IoT edge devices.

A. Bot-IoT Data Set

Bot-IoT data set [5] is publicly and freely available for cyber security research. It contains benign IoT network traffic and four botnet attack scenarios including DoS, DDoS, reconnaissance, and data theft. The testbed that generated the benign IoT network traffic data comprised a weather station, a smart fridge, motion-activated lights, a remote-controlled garage door, and a smart thermostat. Koroniotis *et al.* [5] proposed a method for network packet capturing and feature extraction. In this method, network packets were captured using Tshark[4] while network traffic features were extracted using Argus[5]. Also, new features were generated based on transaction flows of network connections over a sliding window of 100. Previous study [10] confirmed that this method of feature extraction is effective for multi-class network traffic classification. Forty-three features were extracted from a network packet to describe the behaviour of a network traffic sample. The list and description of these features are available in [5]. This data set has 477 benign IoT network traffic samples and 3,668,045 botnet attack samples.

In this study, we identified and removed six redundant features from the dataset, namely: *pkSeqID*, *flgs*, *proto*, *state*, *saddr*, and *daddr*. Specifically, *pkSeqID* is the sequence identification number assigned to the network packet; *flgs*, *proto*, and *state* are duplication of *flgs_number*, *proto_number*, and *state_number*, respectively; while *saddr* is the source Internet Protocol (IP) address and *daddr* is the destination IP address. Therefore, we used only 37 features to represent a network traffic sample in this paper. For effective training of a neural network model, the values of these features were scaled to numbers between 0 and 1 using min-max normalisation given by:

$$\mathbf{x}_{norm} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}, \qquad (5)$$

[4] https://www.wireshark.org/docs/man-pages/tshark.html
[5] https://openargus.org/

where $\mathbf{x}$ is a network traffic feature vector; while $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are the minimum and maximum values of $\mathbf{x}$ respectively.

B. N-BaIoT Data Set

N-BaIoT data set [6] is also publicly and freely available for cyber security research. The IoT testbed that generated this data set comprised two doorbells, a thermostat, a baby monitor, four security cameras, and a webcam. These commercial IoT devices were infected with *Mirai* and BASHLITE botnets, and 115 statistical features that represent the behaviour snapshots of the network traffic were extracted from the network packets over several temporal windows. The details of the data collection and the feature extraction can be found in [6]. This data set contains benign IoT network traffic and IoT botnet scenarios including ACK, Scan, SYN, and UDPP flooding attacks. In this study, we used 363,979 benign IoT network traffic samples and 1,483,658 IoT botnet attack samples. The network traffic features were normalized based on Eq. (5) to eliminate any form of bias in favour of a particular feature.

C. Zero-Day Botnet Attack Detection in IoT Edge Devices

In this subsection, we model zero-day botnet attack scenarios in five IoT edge devices using the Bot-IoT and N-BaIoT data sets.

IoT devices have low computational resources and a limited memory space for data storage. Therefore, the private network traffic data generated by IoT devices within the same network are stored in an IoT edge device for ease of processing. In this study, we have five different IoT edge devices, namely ED1, ED2, ED3, ED4, and ED5.

TABLE II
DISTRIBUTION OF TRAINING DATA IN BOT-IOT DATA SET

Class	ED1	ED2	ED3	ED4	ED5
DDoS	0	337162	337163	337163	337163
DoS	288752	288752	288752	0	288752
Normal	84	84	84	84	0
Reconn	15979	0	15979	15979	15979
Theft	13	14	0	14	14
Total	304828	626012	641978	353240	641908

TABLE III
DISTRIBUTION OF TESTING DATA IN BOT-IOT DATA SET

Class	ED1	ED2	ED3	ED4	ED5
DDoS	115413	115382	115459	115570	115406
DoS	99090	99303	99193	99016	99354
Normal	31	26	34	32	26
Reconn	5572	5398	5420	5490	5318
Theft	5	2	5	3	7
Total	220111	220111	220111	220111	220111

Tables II and III show the distribution of training data and testing data, respectively among the IoT edge devices using the Bot-IoT data set. One class of network traffic was not included in each of the IoT edge devices to determine the ability of the FDL model to detect zero-day botnet attacks without any data privacy concern. Specifically, no sample of DDoS attack, reconnaissance attack, theft attack, DoS attack, and normal traffic was included in ED1, ED2, ED3, ED4, and ED5, respectively. In order to depict real-life scenario, sample distribution in the training data was unbalanced and non-identically distributed across the five classes and across the five IoT edge devices. The generalization performance of the FDL model was evaluated with a unique set of testing data in each of the IoT edge devices as shown in Table III.

TABLE IV
DISTRIBUTION OF TRAINING DATA IN N-BAIOT DATA SET

Class	ED1	ED2	ED3	ED4	ED5
Normal	0	9224	122682	43648	68979
ACK	71529	0	63632	42486	40590
Scan	75239	30258	0	67706	67813
SYN	85900	81834	82895	0	43438
UDPP	57436	61020	56500	39479	0
Total	290104	182336	325709	193319	220820

TABLE V
DISTRIBUTION OF TESTING DATA IN N-BAIOT DATA SET

Class	ED1	ED2	ED3	ED4	ED5
Normal	14877	3893	52419	18693	29564
ACK	30557	33907	27361	18193	17211
Scan	32359	12895	31058	29055	29266
SYN	36621	35013	35577	19778	18647
UDPP	24781	26422	24261	16856	16085
Total	139195	112130	170676	102575	110773

Tables IV and V show the distribution of training data and testing data, respectively among the IoT edge devices using the N-BaIoT data set. One class of network traffic was not included in each of the IoT edge devices to determine the ability of the FDL model to detect zero-day botnet attacks without any data privacy concern. Specifically, no sample of normal traffic, ACK attack, Scan attack, SYN attack, and UDPP attack was included in ED1, ED2, ED3, ED4, and ED5, respectively. The generalization performance of the FDL model was evaluated with a unique set of testing data in each of the IoT edge devices as shown in Table V.

D. Experiments

First, sixteen DNN models were trained and tested with the Bot-IoT and N-BaIoT data sets to determine the optimal neural network architecture for efficient network traffic classification. Then, CDL, LDL, DDL, and FDL models were developed for zero-day botnet attack detection in five IoT edge devices. We used TensorFlow[6] and Keras[7] frameworks for the DNN local models in the CDL, LDL, DDL, and FDL methods, while IBM[8] framework was used for FL in the FDL method. The models were trained using the Spyder[9] Integrated Development Environment (IDE) running on Ubuntu 16.04 LTS workstation with the following specifications: Random Access Memory (32 GB), Processor (Intel Core i7-9700K CPU @ 3.60GHz × 8), and 64-bit Operating System (OS). The deployment of the FDL model in IoT edge devices was

[6] https://www.tensorflow.org/

[7] https://keras.io/

[8] https://ibmfl.mybluemix.net/

[9] https://www.spyder-ide.org/

simulated using five Linux terminals. Finally, the classification performance of the models was evaluated based on the accuracy, precision, recall, and F1 score.

The architecture of a DNN model comprises the input layer, the hidden layers, and the output layer. Each of these layers is made up of neurons. For the input layer, the number of neurons is the same as the number of network traffic features in the training data. In this study, there were 37 and 115 network traffic features in the Bot-IoT and N-BaIoT data sets, respectively. Therefore, the number of input layer neurons was set to 37 and 115 when the DNN models were trained with the Bot-IoT and N-BaIoT data sets, respectively. The numbers of the hidden layers and their hidden neurons are usually determined by experimentation. So, we varied the number of the hidden layers between 1 and 4, while the number of the hidden neurons in each layer was varied between 25 and 100 at an interval of 25. For the output layer, the number of neurons is the same of the number of classes of network traffic in the training data. There are five classes of network traffic in each of the Bot-IoT and N-BaIoT data sets. Therefore, the number of output layer neurons was set to 5. The DNN models were trained with a moderate batch size (i.e., 128) and a small number of epochs (i.e., 5) to minimize the time spent during the training process as well as to avoid model over-fitting.

For the CDL method, each of the IoT edge devices (ED1-ED5) transmitted its training data to a central server for aggregation. Therefore, the CDL model was trained with an aggregated data in the cloud. A copy of the CDL model was sent back to all the IoT edge devices for network traffic classification on the testing data. For the LDL method, model training was performed with the local training data in the edge IoT devices. Therefore, a unique LDL model was developed for each of the IoT edge devices. The DDL method is similar to LDL method. Unlike the LDL method, the parameters of the local models in the DDL method were sent to a model parameter server for aggregation. The communication between the model parameter server and the five IoT edge devices was established using the Flask[10] web framework. A global DDL model was developed and a copy of this model was transmitted to all the IoT edge devices for network traffic classification on their testing data. The FDL method is similar to the DDL method. However, in the FDL method, the model parameter server receives further updates from the local models in the IoT edge devices to improve the classification performance of the global FDL model.

The classification performance of the CDL, LDL, DDL, and FDL models was evaluated with the testing data in the IoT edge devices based on accuracy, precision, recall, and F1 score:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{6}$$

$$Precision = \frac{TP}{TP + FP}, \tag{7}$$

$$Recall = \frac{TP}{TP + FN}, \tag{8}$$

$$F1 = \frac{2 \times TP}{(2 \times TP) + FP + FN}, \tag{9}$$

where True Positive (TP) is the number of network traffic samples in the positive class that are correctly classified as positive; False Positive (FP) is the number of network traffic samples in the negative class that are misclassified as positive; True Negative (TN) is the number of network traffic samples in the negative class that are correctly classified as negative; and False Negative (FN) is the number of network traffic samples in the negative class that are misclassified as positive.

V. Results and Discussion

In this section, we analyze and compare the effectiveness of the FDL model with that of the CDL, LDL, and DDL models based on their: (a) classification performance; (b) training time; (c) data privacy preservation; (d) communication cost; (e) memory space requirement; and (e) network latency.

A. Optimal Architecture for DNN Model

In this subsection, we evaluate the performance of sixteen CDL models to determine the optimal neural network architecture for botnet attack detection in IoT edge devices.

TABLE VI
Recall (%) of CDL model for different DNN architectures based on Bot-IoT data set

Hidden neurons	ED1	ED2	ED3	ED4	ED5	Avg.
25	57.99	57.97	57.97	39.99	47.78	52.34
50	61.83	61.18	61.53	47.11	56.33	57.60
75	71.03	70.41	72.87	58.06	69.47	68.37
100	73.77	72.46	73.02	64.05	76.66	71.99
100, 25	74.34	73.05	74.78	56.04	69.55	69.55
100, 50	75.01	71.79	74.81	55.90	69.38	69.38
100, 75	75.73	75.10	74.90	56.93	70.67	70.67
100, 100	75.69	74.39	74.84	56.73	70.41	70.41
100, 100, 25	75.95	74.67	76.88	57.63	71.28	71.28
100, 100, 50	75.36	75.37	75.17	57.22	70.78	70.78
100, 100, 75	75.35	75.35	75.74	57.36	70.95	70.95
100, 100, 100	80.00	89.99	81.93	63.73	78.91	78.91
100, 100, 100, 25	85.31	90.00	81.94	65.31	80.64	80.64
100, 100, 100, 50	83.28	92.63	89.33	67.31	83.13	83.14
100, 100, 100, 75	86.60	92.58	93.71	96.27	96.93	93.22
100, 100, 100, 100	95.34	93.32	98.81	99.21	98.50	**97.04**

Tables VI and VII show that the DNN model with four hidden layers and 100 hidden neurons per layer gave the best overall classification performance. The DNN model was able to perform well because the increase in the numbers of the hidden layers and the neurons per hidden layer helped the model to produce a more accurate hierarchical representation of the network traffic features using additional parameters. The model achieved an average recall of 97.04% and 97.88% when it was simulated with Bot-IoT and N-BaIoT data sets, respectively. This implies that nearly all of the network traffic samples in each of the five IoT edge devices were correctly classified. Therefore, this particular DNN architecture is suitable for efficient botnet attack detection in IoT networks. Subsequently, we used this optimal DNN model architecture to develop CDL, LDL, DDL, and FDL models.

[10] https://palletsprojects.com/p/flask/

TABLE VII
AVERAGE RECALL (%) OF CDL MODEL FOR DIFFERENT DNN ARCHITECTURES BASED ON N-BAIOT DATA SET

Hidden neurons	ED1	ED2	ED3	ED4	ED5	Avg.
25	93.01	94.62	99.92	88.67	90.03	93.25
50	93.03	94.77	99.99	90.57	91.06	93.88
75	98.11	94.86	99.97	90.71	91.86	95.10
100	93.25	93.91	99.99	92.16	96.30	95.12
100, 25	93.03	94.53	99.99	95.25	92.59	95.08
100, 50	95.93	94.82	99.99	95.04	97.67	96.69
100, 75	99.97	94.87	99.99	94.62	96.28	97.15
100, 100	94.62	94.89	99.99	94.39	94.23	95.62
100, 100, 25	99.96	94.89	99.99	95.61	97.53	97.60
100, 100, 50	99.98	94.86	99.99	97.03	99.60	98.29
100, 100, 75	95.25	91.84	99.99	96.88	99.91	96.77
100, 100, 100	99.98	94.66	99.99	95.65	98.22	97.70
100, 100, 100, 25	95.34	94.92	99.99	93.45	97.58	96.26
100, 100, 100, 50	99.96	94.91	100.00	94.50	97.95	97.46
100, 100, 100, 75	99.96	94.90	99.99	96.12	97.69	97.73
100, 100, 100, 100	99.94	94.80	99.99	96.35	98.32	**97.88**

TABLE VIII
CLASSIFICATION PERFORMANCE OF CDL MODEL BASED ON BOT-IOT DATA SET

	Metric (%)	DDoS	DoS	Normal	Recon.	Theft
ED1	Accuracy	99.96	99.96	100.00	100.00	100.00
	Precision	99.98	99.95	100.00	99.95	100.00
	Recall	99.95	99.97	96.77	100.00	80.00
	F1 score	99.97	99.96	98.36	99.97	88.89
ED2	Accuracy	99.97	99.97	100.00	100.00	100.00
	Precision	99.98	99.95	100.00	99.98	100.00
	Recall	99.96	99.98	100.00	100.00	66.67
	F1 score	99.97	99.96	100.00	99.99	80.00
ED3	Accuracy	99.97	99.97	100.00	100.00	100.00
	Precision	99.98	99.95	96.97	99.96	100.00
	Recall	99.96	99.97	94.12	99.98	100.00
	F1 score	99.97	99.96	95.52	99.97	100.00
ED4	Accuracy	99.96	99.96	100.00	100.00	100.00
	Precision	99.97	99.95	100.00	99.98	100.00
	Recall	99.95	99.97	96.15	100.00	100.00
	F1 score	99.96	99.96	98.04	99.99	100.00
ED5	Accuracy	99.97	99.97	100.00	100.00	100.00
	Precision	99.98	99.95	96.15	99.96	100.00
	Recall	99.96	99.98	92.59	99.98	100.00
	F1 score	99.97	99.97	94.34	99.97	100.00

B. *Classification Performance of CDL Model*

In this subsection, we evaluate the class-wise classification performance of the CDL in the five IoT edge devices.

Table VIII shows that the CDL model achieved an excellent performance in detecting DDoS, DoS, and reconnaissance attacks when it was trained and tested with Bot-IoT data set. Although the CDL model also achieved a high accuracy for the Normal and Theft classes, its precision, recall, and the F1 score were relatively low in some cases. Table II shows that the number of samples in each of the Normal and Theft classes is far less than the number of samples in each of the DDoS, DoS, and reconnaissance classes. Therefore, the class imbalance adversely affected the ability of CDL model to correctly classify the samples in the minority classes. In other words, class imbalance in the training data reduced the rates of precision, recall, and F1 score of the CDL model in the Normal and Theft classes. For instance, in ED1, the recall and F1 score of CDL model for the Theft class were 80% and 88.89%, respectively. Also, in ED2, the recall of CDL model for the Theft class was 66.67% and its F1 score was 80%. Furthermore, in ED5, the precision, recall, and F1 score of CDL model for the Normal class were 96.15%, 92.59%, and 94.34%, respectively.

Table IX shows the class-wise performance of the CDL model in the five edge devices when it was trained and tested with the N-BaIoT data set. The CDL model achieved an excellent classification performance in ED1, ED3, and ED5 with accuracy, precision, recall, and F1 score of more than 95%. On the other hand, the CDL model had a relatively lower classification performance in ED2 and ED4. Specifically, some of the ACK attack samples in ED2 were misclassified as UDPP attack, while some of the SYN attack samples in ED4 were misclassified as Scan attack.

TABLE IX
CLASSIFICATION PERFORMANCE OF CDL MODEL BASED ON N-BAIOT DATA SET

	Metric (%)	Normal	ACK	Scan	SYN	UDPP
ED1	Accuracy	99.98	99.97	99.99	100.00	99.96
	Precision	99.99	100.00	99.98	99.99	99.75
	Recall	99.85	99.86	99.98	99.99	100.00
	F1 score	99.92	99.93	99.98	99.99	99.88
ED2	Accuracy	99.99	92.16	99.99	100.00	92.16
	Precision	99.90	100.00	99.98	100.00	75.03
	Recall	99.95	74.07	99.97	100.00	100.00
	F1 score	99.92	85.10	99.98	100.00	85.73
ED3	Accuracy	99.99	100.00	100.00	100.00	100.00
	Precision	99.98	99.98	100.00	100.00	100.00
	Recall	100.00	100.00	99.99	99.99	99.98
	F1 score	99.99	99.99	100.00	99.99	99.99
ED4	Accuracy	99.99	99.99	96.50	96.50	99.99
	Precision	99.94	99.99	89.02	100.00	99.93
	Recall	100.00	99.92	99.99	81.86	99.99
	F1 score	99.97	99.96	94.19	90.03	99.96
ED5	Accuracy	99.99	99.30	99.34	99.34	99.30
	Precision	99.97	100.00	97.59	100.00	95.43
	Recall	99.99	95.53	99.98	96.11	99.99
	F1 score	99.98	97.71	98.77	98.01	97.66

C. *Classification Performance of LDL Model*

In this subsection, we evaluate the class-wise classification performance of the LDL model in the five IoT edge devices.

Table X shows that the LDL model could not detect zero-day botnet attacks when it was trained and tested with the Bot-IoT data set. In ED1, the LDL model achieved an excellent performance in detecting benign network traffic as well as DoS and reconnaissance attacks. However, all the samples in the DDoS and Theft classes were incorrectly classified as DoS attacks. This is partly because there was no DDoS attack sample in the training data of ED1. Also, the number of samples in the Theft class was relatively low, compared to the number of samples in each of the DoS and Reconnaissance classes. In ED2, the LDL model achieved an excellent performance in detecting benign network traffic as well as DDoS and DoS attacks. However, all the samples in the Reconnaissance class and some of the samples in Theft class were incorrectly classified as benign network traffic. This is partly because there was no reconnaissance attack sample in the training data of ED2. Also, the number of samples in the Theft class was relatively low, compared to the number of samples in each of the DDoS and DoS classes. In ED3, the LDL model achieved an excellent performance in detecting DDoS, DoS, and reconnaissance attacks. However, all the samples in the Theft class were incorrectly classified as benign network traffic. This is partly because there was no Theft attack sample in the training data of ED3. Also, the number of samples in the Normal class was relatively low, compared to the number of samples in each of the DDoS, DoS, and reconnaissance classes. In ED4, the LDL model achieved an excellent performance in detecting benign network traffic as well as DDoS, reconnaissance, and theft attacks. However, each of the samples in the DoS class was incorrectly classified as either DDoS or Theft attack. This happened because there was no DoS attack sample in the training data of ED4. In ED5, the LDL model achieved an excellent performance in detecting DDoS, DoS, reconnaissance, and theft attacks. However, each of the samples in the Normal class was incorrectly classified as either DDoS, DoS, Reconnaissance, or Theft attack. This happened because there was no benign network traffic sample in the training data of ED5.

TABLE X
CLASSIFICATION PERFORMANCE OF LDL MODELS BASED ON BOT-IOT DATA SET

	Metric (%)	DDoS	DoS	Normal	Recon.	Theft
ED1	Accuracy	47.57	47.56	100.00	100.00	100.00
	Precision	0.00	46.19	100.00	99.98	0.00
	Recall	0.00	100.00	100.00	99.98	0.00
	F1 score	0.00	63.19	100.00	99.98	0.00
ED2	Accuracy	99.57	98.13	99.66	97.60	100.00
	Precision	99.38	96.04	3.75	0.00	100.00
	Recall	99.80	99.98	93.55	0.00	66.67
	F1 score	99.59	97.97	7.20	0.00	80.00
ED3	Accuracy	99.86	99.86	100.00	100.00	100.00
	Precision	99.78	99.95	100.00	99.89	0.00
	Recall	99.96	99.75	82.35	100.00	0.00
	F1 score	99.87	99.85	90.32	99.95	0.00
ED4	Accuracy	55.05	55.00	100.00	100.00	99.95
	Precision	53.89	0.00	96.30	99.98	5.69
	Recall	100.00	0.00	100.00	99.98	100.00
	F1 score	70.04	0.00	98.11	99.98	10.77
ED5	Accuracy	99.85	99.85	99.99	99.99	100.00
	Precision	99.82	99.86	0.00	99.48	100.00
	Recall	99.88	99.79	0.00	99.98	100.00
	F1 score	99.85	99.83	0.00	99.73	100.00

TABLE XI
CLASSIFICATION PERFORMANCE OF LDL MODELS BASED ON N-BAIOT DATA SET

	Metric (%)	Normal	ACK	Scan	SYN	UDPP
ED1	Accuracy	89.31	98.55	89.34	99.99	98.52
	Precision	0.00	99.99	68.56	99.99	92.33
	Recall	0.00	93.40	100.00	99.97	100.00
	F1 score	0.00	96.58	81.35	99.98	96.01
ED2	Accuracy	99.99	69.76	100.00	100.00	69.76
	Precision	99.85	0.00	99.98	100.00	43.80
	Recall	99.92	0.00	99.99	99.99	100.00
	F1 score	99.88	0.00	99.98	99.99	60.91
ED3	Accuracy	97.23	100.00	81.80	84.57	100.00
	Precision	91.73	99.99	0.00	57.46	100.00
	Recall	100.00	100.00	0.00	99.99	99.98
	F1 score	95.69	99.99	0.00	72.98	99.99
ED4	Accuracy	99.99	99.99	80.72	80.72	99.99
	Precision	99.94	100.00	59.50	0.00	99.96
	Recall	100.00	99.95	99.99	0.00	99.99
	F1 score	99.97	99.98	74.61	0.00	99.98
ED5	Accuracy	99.99	85.48	100.00	100.00	85.48
	Precision	99.96	51.69	100.00	100.00	0.00
	Recall	100.00	99.99	99.98	99.99	0.00
	F1 score	99.98	68.15	99.99	99.99	0.00

Table XI shows that the LDL model could not detect zero-day botnet attacks when it was trained and tested with the N-BaIoT data set. In ED1, the LDL model achieved an excellent performance in detecting ACK, Scan, SYN, and UDPP attacks. However, all the samples in the Normal class were misclassified as Scan attack because there was no benign sample in the training data of ED1. In ED2, the LDL model achieved achieved an excellent performance in detecting benign network traffic as well as Scan and SYN attacks. However, all the samples in the ACK class were misclassified as UDPP attack. This happened because there was no ACK attack sample in the training data of ED2. In ED3, the LDL model achieved an excellent performance in detecting benign network traffic as well as ACK and UDPP attacks. However, all the samples in Scan class were misclassified as SYN attack. This happened because there was no Scan attack sample in the training data of ED3. In ED4, the LDL model achieved an excellent performance in detecting benign network traffic as well as the ACK and UDPP attacks. However, all of the samples in the SYN class were misclassified as Scan attack. This happened because there was no SYN attack sample in the training data of ED4. In ED5, the LDL model achieved achieved an excellent performance in detecting benign network traffic as well as Scan and SYN attacks. However, all of the samples in the UDPP class were misclassified as ACK attack. This happened because there was no UDPP attack sample in the training data of ED5.

D. Classification Performance of DDL Model

In this subsection, we evaluate the class-wise classification performance of the DDL model in the five IoT edge devices.

Table XII shows that the DDL model could only detect DDoS attacks efficiently when it was trained and tested with the Bot-IoT data set. The DDL model achieved a high recall of more than 98% in detecting DDoS attacks. However, benign network traffic as well as DoS, reconnaissance, and theft attacks were largely misclassified. The recall of the DDL model was 32.28±0.08% for the DoS class, 69.96±5.29% for the Normal class, and 0.29 ± 0.04% for the Reconnaissance class. A recall of 0% for the Theft class implies that none of the samples in this attack category was correctly classified.

Table XIII shows that the DDL model could not detect all the categories of botnet attacks in the five IoT edge devices efficiently when it was trained and tested with N-BaIoT data set. The DDL model achieved an excellent performance in detecting benign network traffic as well as ACK and SYN attacks with a recall of 99.87 ± 0.19%, 92.99 ± 6.64%, and 96.82 ± 4.41%, respectively. However, the DDL model had a relatively low recall of 43.09% for the Scan class in ED1, 52.66% for the UDPP class in ED4, and 53.27% for the UDPP class in ED5. Also, the DDL model largely misclassified the UDPP attack samples in ED1, ED2, and ED3.

TABLE XII
CLASSIFICATION PERFORMANCE OF DDL MODEL BASED ON BOT-IOT DATA SET

	Metric (%)	DDoS	DoS	Normal	Reconn.	Theft
ED1	Accuracy	66.83	68.76	99.64	97.48	100.00
	Precision	61.44	94.78	2.72	100.00	0.00
	Recall	98.71	32.38	70.97	0.31	0.00
	F1 score	75.73	48.27	5.24	0.61	0.00
ED2	Accuracy	66.84	68.68	99.67	97.60	100.00
	Precision	61.52	94.70	2.95	100.00	0.00
	Recall	98.69	32.17	70.97	0.23	0.00
	F1 score	75.79	48.03	5.66	0.45	0.00
ED3	Accuracy	66.86	68.78	99.64	97.49	100.00
	Precision	61.50	94.74	2.76	100.00	0.00
	Recall	98.70	32.31	64.71	0.31	0.00
	F1 score	75.78	48.18	5.30	0.61	0.00
ED4	Accuracy	66.87	68.74	99.66	97.56	100.00
	Precision	61.50	94.89	2.27	100.00	0.00
	Recall	98.74	32.29	65.38	0.32	0.00
	F1 score	75.79	48.18	4.39	0.63	0.00
ED5	Accuracy	66.85	68.70	99.67	97.58	100.00
	Precision	61.51	94.67	2.83	100.00	0.00
	Recall	98.67	32.26	77.78	0.26	0.00
	F1 score	75.78	48.12	5.46	0.52	0.00

TABLE XIII
CLASSIFICATION PERFORMANCE OF DDL MODEL BASED ON N-BAIOT DATA SET

	Metric (%)	Normal	ACK	Scan	SYN	UDPP
ED1	Accuracy	86.77	82.20	86.77	99.99	82.19
	Precision	44.68	55.22	100.00	99.97	27.27
	Recall	99.93	100.00	43.09	99.99	0.01
	F1 score	61.75	71.15	60.23	99.98	0.02
ED2	Accuracy	95.34	77.74	96.92	99.89	76.44
	Precision	42.67	57.63	100.00	100.00	100.00
	Recall	100.00	99.72	73.19	99.65	0.01
	F1 score	59.82	73.04	84.52	99.83	0.02
ED3	Accuracy	93.04	89.47	96.58	94.74	85.79
	Precision	81.76	62.67	100.00	79.86	61.70
	Recall	99.54	84.86	81.21	99.99	0.12
	F1 score	89.78	72.10	89.63	88.80	0.24
ED4	Accuracy	96.70	91.77	99.24	98.11	90.98
	Precision	84.71	71.01	98.24	99.89	87.47
	Recall	99.92	90.54	99.08	90.29	52.66
	F1 score	91.69	79.59	98.66	94.85	65.74
ED5	Accuracy	96.53	93.46	99.64	99.01	92.04
	Precision	88.51	73.79	99.80	99.93	86.84
	Recall	99.96	89.84	98.85	94.19	53.27
	F1 score	93.89	81.03	99.32	96.97	66.03

E. Classification Performance of FDL Model

In this subsection, we evaluate the classification performance of the FDL model for different number of communication rounds in the five IoT edge devices.

Figs. 2-6 show that the accuracy, precision, recall, and F1 score of the FDL model increased significantly as the number of communication rounds increased from 1 to 5 when it was trained with the Bot-IoT data set. At the end of the fifth communication round, the FDL model achieved an accuracy of 99.79 ± 0.01%, a precision of 99.51 ± 0.38%, a recall of 96.27 ± 2.45%, and a F1 score of 97.68 ± 1.52%. Table XV shows that the FDL model achieved an excellent performance in detecting benign network traffic as well as DoS, DDoS, reconnaissance, and theft attacks in the five IoT edge devices. Further local model updates from the IoT edge devices beyond the sixth communication round did not lead to any significant classification performance gain.

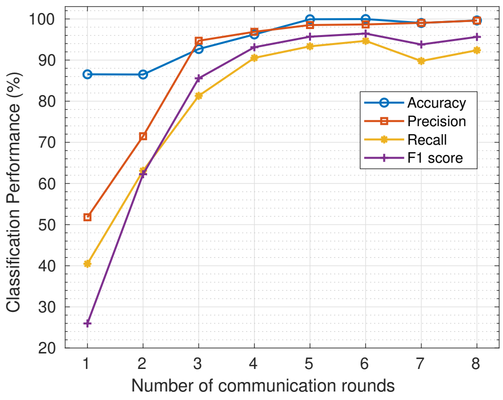

Fig. 2. Classification performance of FDL model in ED1 based on Bot-IoT data set.

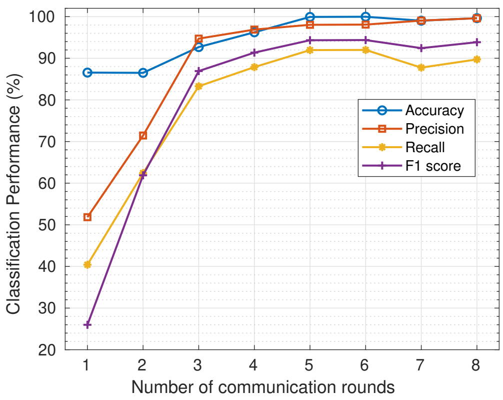

Fig. 3. Classification performance of FDL model in ED2 based on Bot-IoT data set.

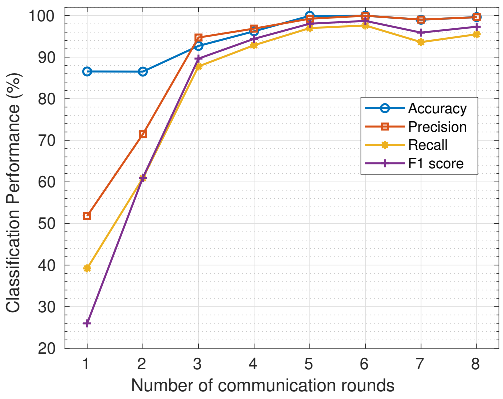

Fig. 4. Classification performance of FDL model in ED3 based on Bot-IoT data set.

Figs. 7-11 show that the accuracy, precision, recall, and F1 score of the FDL model increased significantly as the number of communication rounds increased from 1 to 6 when it was trained with the N-BaIoT data set. At the end of the sixth communication round, the FDL model achieved an accuracy

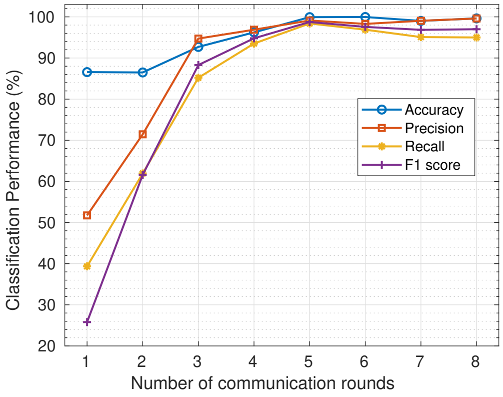

Fig. 5. Classification performance of FDL model in ED4 based on Bot-IoT data set.

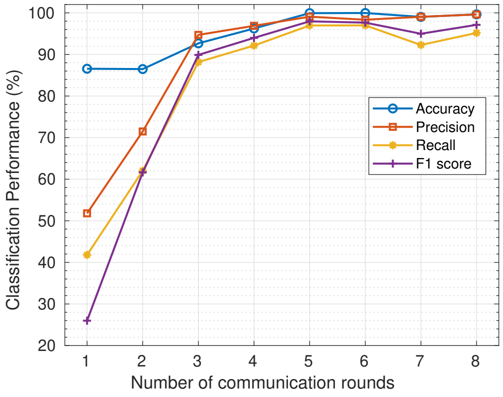

Fig. 6. Classification performance of FDL model in ED5 based on Bot-IoT data set.

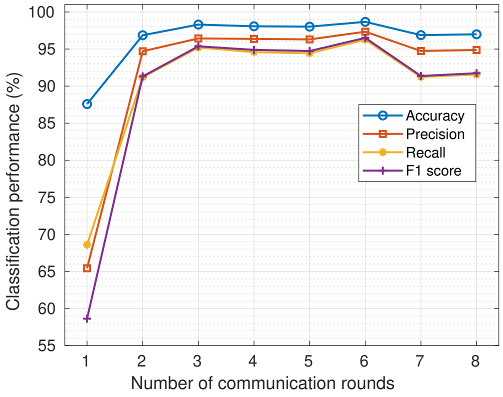

Fig. 7. Classification performance of FDL model in ED1 based on N-BaIoT data set.

of 99.00 ± 0.60%, a precision of 96.87 ± 1.86%, a recall of 97.24 ± 1.59%, and a F1 score of 96.88 ± 1.67%. Table XIV shows that the FDL model achieved an excellent performance in detecting benign network traffic as well as ACK, Scan, SYN, and UDPP attacks in the five IoT edge devices. Further local model updates from the IoT edge devices beyond the sixth communication round did not improve the classification performance of the FDL model.

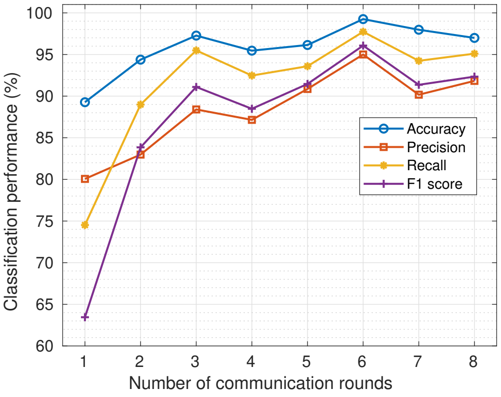

Fig. 8. Classification performance of FDL model in ED2 based on N-BaIoT data set.

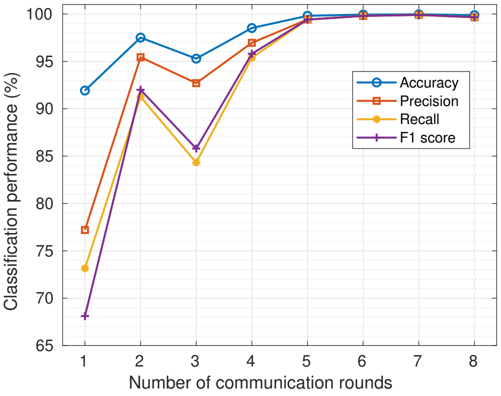

Fig. 9. Classification performance of FDL model in ED3 based on N-BaIoT data set.

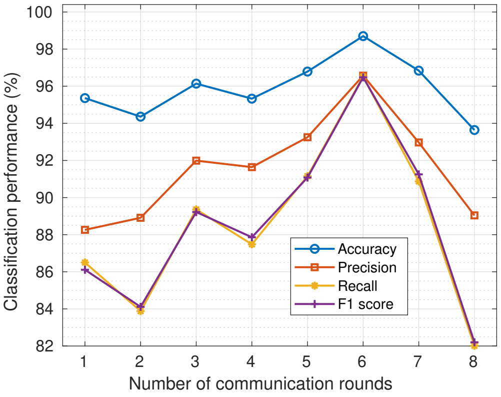

Fig. 10. Classification performance of FDL model in ED4 based on N-BaIoT data set.

F. Comparison of the FDL Method with the State-of-the-art Methods

In this subsection, we compare the effectiveness of the FDL model with that of the CDL, LDL, and DDL models in the five IoT edge devices.

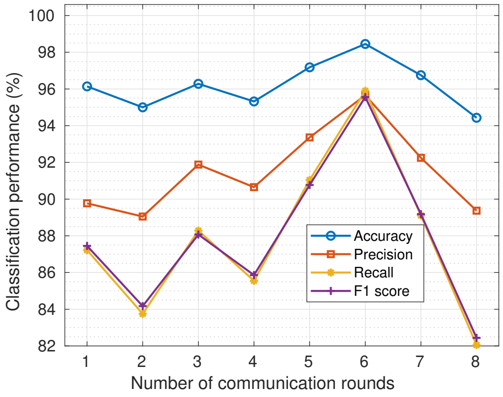

Fig. 11. Classification performance of FDL model in ED5 based on N-BaIoT data set.

TABLE XIV
CLASSIFICATION PERFORMANCE OF FDL MODEL BASED ON BOT-IOT DATA SET

	Metric (%)	DDoS	DoS	Normal	Recon.	Theft
ED1	Accuracy	99.81	99.81	100.00	100.00	100.00
	Precision	99.97	99.61	93.10	99.89	100.00
	Recall	99.67	99.97	87.10	99.96	80.00
	F1 score	99.82	99.79	90.00	99.93	88.89
ED2	Accuracy	99.82	99.82	100.00	100.00	100.00
	Precision	99.97	99.64	90.63	99.94	100.00
	Recall	99.69	99.97	93.55	99.94	66.67
	F1 score	99.83	99.80	92.06	99.94	80.00
ED3	Accuracy	99.83	99.83	100.00	100.00	100.00
	Precision	99.97	99.66	96.67	99.91	100.00
	Recall	99.71	99.97	85.29	99.98	100.00
	F1 score	99.84	99.81	90.63	99.95	100.00
ED4	Accuracy	99.83	99.83	100.00	100.00	100.00
	Precision	99.99	99.63	96.00	99.96	100.00
	Recall	99.68	99.98	92.31	99.98	100.00
	F1 score	99.84	99.81	94.12	99.97	100.00
ED5	Accuracy	99.83	99.83	100.00	100.00	100.00
	Precision	99.98	99.65	95.83	99.93	100.00
	Recall	99.70	99.98	85.19	99.98	100.00
	F1 score	99.84	99.81	90.20	99.95	100.00

TABLE XV
CLASSIFICATION PERFORMANCE OF FDL MODEL BASED ON N-BAIOT DATA SET

	Metric (%)	Normal	ACK	Scan	SYN	UDPP
ED1	Accuracy	99.98	96.69	99.99	99.99	96.68
	Precision	99.81	86.91	100.00	99.97	100.00
	Recall	99.99	100.00	99.95	99.99	81.35
	F1 score	99.90	93.00	99.98	99.98	89.72
ED2	Accuracy	99.06	99.06	99.08	99.99	99.06
	Precision	78.71	99.34	100.00	100.00	96.93
	Recall	100.00	97.54	91.99	99.98	99.14
	F1 score	88.09	98.43	95.83	99.99	98.02
ED3	Accuracy	99.98	99.86	100.00	100.00	99.86
	Precision	99.94	99.98	100.00	99.99	99.08
	Recall	100.00	99.12	99.99	99.99	99.96
	F1 score	99.97	99.55	99.99	99.99	99.52
ED4	Accuracy	99.32	97.99	99.07	99.07	98.05
	Precision	96.39	98.88	98.29	97.69	91.59
	Recall	100.00	89.67	98.42	97.46	97.04
	F1 score	98.16	94.05	98.36	97.58	94.24
ED5	Accuracy	99.68	97.91	98.31	98.31	98.06
	Precision	98.82	99.88	99.86	91.05	88.67
	Recall	99.99	86.62	93.73	99.79	99.36
	F1 score	99.40	92.78	96.70	95.22	93.71

Tables XVI and XVII show that the CDL and FDL models achieved better classification performance than the LDL and DDL models when they were trained and tested with the Bot-IoT and N-BaIoT data set, respectively. The CDL model achieved an excellent classification performance because it was trained with a large and diverse data which covered all the benign network traffic patterns and all the categories of botnet attacks that were generated and transmitted from the five edge IoT devices to a central cloud server. However, this method will leak the privacy of the IoT edge devices. Also, the CDL method requires long training time (as shown in Figs. 12 and 13), high communication overhead, and large memory space for data storage (as shown in Figs. 14 and 15). On the other hand, in LDL, DDL, and FDL, the network traffic features of the IoT edge devices were not shared with a third-party central cloud server to preserve the data privacy of users. LDL and DDL models required a shorter training time and a lower memory space for data storage, and they incurred lower communication overhead. However, the classification performance of the LDL and DDL models is relatively low, compared to the CDL and FDL models. In other words, these models have low detection rate and high false alarm rate. The LDL models had a lower classification performance because each of them was trained with insufficient private network traffic and fewer botnet attack scenarios in a single IoT edge device. The DDL model achieved a lower classification performance because the communication of local model updates from the IoT edge devices to a central parameter server was limited to a single round. Therefore, the LDL and DDL models are not suitable for zero-day botnet attack detection in IoT edge devices.

TABLE XVI
CLASSIFICATION PERFORMANCE OF CDL, LDL, AND FDL MODELS BASED ON BOT-IOT DATA SET

	Metric (%)	CDL	LDL	DDL	FDL
ED1	Accuracy	99.98	79.03	86.54	99.92
	Precision	99.97	49.24	51.79	98.52
	Recall	95.34	60.00	40.47	93.34
	F1 score	97.43	52.64	25.97	95.69
ED2	Accuracy	99.99	98.99	86.56	99.93
	Precision	99.98	59.83	51.83	98.04
	Recall	93.32	72.00	40.41	91.96
	F1 score	95.98	56.95	25.99	94.33
ED3	Accuracy	99.99	99.94	86.55	99.93
	Precision	99.37	79.92	51.80	99.24
	Recall	98.81	76.41	39.20	96.99
	F1 score	99.09	78.00	25.98	98.04
ED4	Accuracy	99.98	82.00	86.57	99.93
	Precision	99.98	51.17	51.73	99.12
	Recall	99.22	80.00	39.34	98.39
	F1 score	99.59	55.78	25.80	98.75
ED5	Accuracy	99.99	99.93	86.56	99.93
	Precision	99.21	79.83	51.80	99.08
	Recall	98.50	79.93	41.79	96.97
	F1 score	98.85	79.88	25.98	97.96

The summary of our findings in this paper is presented in Table XVIII. FDL method detects zero-day botnet attacks with high classification performance; it guarantees data privacy and security; it has low communication overhead; it requires low memory space for the data storage; and it has low latency. The FDL model achieved a better classification performance

TABLE XVII
CLASSIFICATION PERFORMANCE OF CDL, LDL, AND FDL MODELS BASED ON N-BAIOT DATA SET

	Metric (%)	CDL	LDL	DDL	FDL
ED1	Accuracy	99.98	95.14	87.58	98.67
	Precision	99.94	72.17	65.43	97.34
	Recall	99.94	78.67	68.60	96.26
	F1 score	99.94	74.78	58.63	96.51
ED2	Accuracy	96.86	87.90	89.27	99.25
	Precision	94.98	68.72	80.06	95.00
	Recall	94.80	79.98	74.52	97.73
	F1 score	94.15	72.16	63.45	96.07
ED3	Accuracy	100.00	92.72	91.92	99.94
	Precision	99.99	69.83	77.20	99.80
	Recall	99.99	79.99	73.14	99.81
	F1 score	99.99	73.73	68.11	99.80
ED4	Accuracy	98.59	92.28	95.36	98.70
	Precision	97.78	71.88	88.26	96.57
	Recall	96.35	79.99	86.50	96.57
	F1 score	96.82	74.91	86.11	96.48
ED5	Accuracy	99.46	94.19	96.14	98.45
	Precision	98.60	70.33	89.77	95.66
	Recall	98.32	79.99	87.22	95.90
	F1 score	98.43	73.62	87.45	95.56

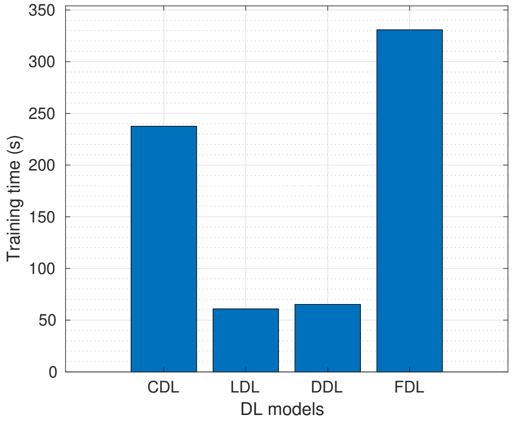

Fig. 12. Training time of CDL, LDL, DDL, and FDL models based on Bot-IoT data set.

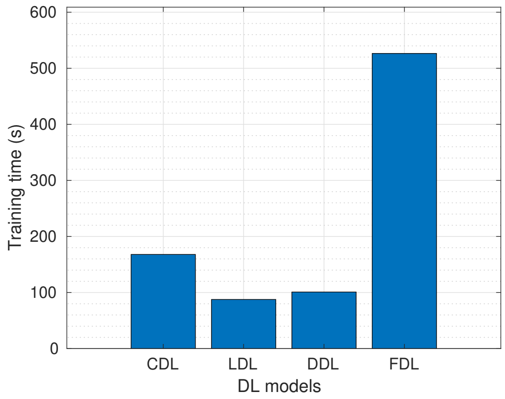

Fig. 13. Training time of CDL, LDL, DDL, and FDL models based on N-BaIoT data set.

than the LDL and DDL models because the central parameter server receives multiple local model updates from all the IoT edge devices. The only trade-off in FDL method is the time required to train its model. Therefore, FDL method is efficient for zero-day botnet attack detection in IoT edge devices.

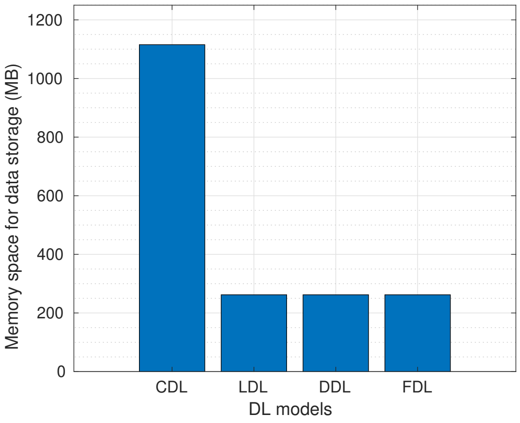

Fig. 14. Memory space required for data storage based on Bot-IoT data set.

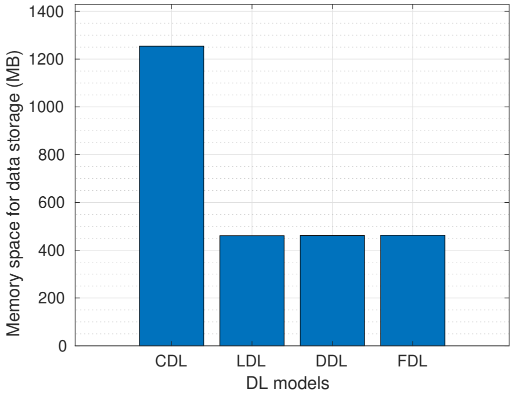

Fig. 15. Memory space required for data storage based on N-BaIoT data set.

TABLE XVIII
SUMMARY OF RESEARCH FINDINGS

	CDL	LDL	DDL	FDL
Data aggregation	✓	✗	✗	✗
Model parameter aggregation	✗	✗	✓	✓
Classification performance	high	low	low	high
Training time	long	short	short	long
Data privacy	✗	✓	✓	✓
Communication overhead	high	low	low	low
Memory requirement	high	low	low	low
Latency	high	low	low	low

VI. CONCLUSION

In this paper, we proposed FDL method for zero-day attack detection in IoT edge devices. The FDL model was developed with the Bot-IoT and N-BaIoT data sets, and its effectiveness was compared with the CDL, LDL, and DDL models. The CDL model involves data aggregation, and it achieved high classification performance. However, it did not preserve the privacy and security of network traffic data in IoT edge devices. Also, the CDL model had high communication overhead, large memory space requirement for data storage, high network latency, and it took long time to train the model. Although the LDL and DDL models overcame the limitations of the CDL model, their classification performance was very low. Interestingly, the FDL model outperformed the CDL, LDL, and DDL models, except for the long training time. In the future, we plan to optimise the FL algorithm such that the FDL model will converge with fewer number of communication rounds. This will significantly reduce the training time of the FDL model. Hence, the FDL method is most efficient for zero-day botnet attack detection in the IoT edge devices. In the future, we hope to further explore how advanced FL algorithms can improve the classification performance of attack detection models.

References

[1] N. Koroniotis, N. Moustafa, and E. Sitnikova, "Forensics and deep learning mechanisms for botnets in internet of things: A survey of challenges and solutions," *IEEE Access*, vol. 7, pp. 61 764–61 785, 2019.

[2] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[3] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "Ddos in the iot: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] G. Vormayr, T. Zseby, and J. Fabini, "Botnet communication patterns," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2768–2796, 2017.

[5] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[6] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot: Network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[7] A. Holst, "Number of iot connected devices worldwide 2019-2030," January 20, 2021, accessed: 2021-02-20. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[8] Statista, "Global iot end-user spending worldwide 2017-2025," January 22, 2021, accessed: 2021-02-20. [Online]. Available: https://www.statista.com/statistics/976313/global-iot-market-size/

[9] E. Estopace, "Idc forecasts connected iot devices to generate 79.4zb of data in 2025," June 22, 2019, accessed: 2021-02-20. [Online]. Available: https://futureiot.tech/idc-forecasts-connected-iot-devices-to-generate-79-4zb-of-data-in-2025/

[10] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet of things networks," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4944–4956, 2021.

[11] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting iot and cloud networks," *IEEE Internet of Things Journal*, 2020.

[12] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, "Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[13] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for internet of things," *Computer Networks*, vol. 186, p. 107784, 2021.

[14] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1285–1297, 2020.

[15] G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, and M. Colajanni, "Deep reinforcement adversarial learning against botnet evasion attacks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 1975–1987, 2020.

[16] R. Zhao, J. Yin, Z. Xue, G. Gui, B. Adebisi, T. Ohtsuki, H. Gacanin, and H. Sari, "An efficient intrusion detection method based on dynamic autoencoder," *IEEE Wireless Communications Letters*, 2021.

[17] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, and A. A. Atayero, "Memory-efficient deep learning for botnet attack detection in iot networks," *Electronics*, vol. 10, no. 9, p. 1104, 2021.

[18] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, "Smote-drnn: A deep learning algorithm for botnet detection in the internet-of-things networks," *Sensors*, vol. 21, no. 9, p. 2985, 2021.

[19] S. I. Popoola, B. Adebisi, M. Hammoudeh, H. Gacanin, and G. Gui, "Stacked recurrent neural network for botnet detection in smart homes," *Computers & Electrical Engineering*, vol. 92, p. 107039, 2021.

[20] S. I. Popoola, R. Ande, K. B. Fatai, and B. Adebisi, "Deep bidirectional gated recurrent unit for botnet detection in smart homes," *Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics: Theories and Applications*, p. 29, 2021.

[21] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, 2020.

[22] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "Dïot: A federated self-learning anomaly detection system for iot," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 756–767.

[23] S. Kim, H. Cai, C. Hua, P. Gu, W. Xu, and J. Park, "Collaborative anomaly detection for internet of things based on federated learning," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2020, pp. 623–628.

[24] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. J. Piran, and M. S. Hossain, "Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning," *IEEE Internet of Things Journal*, 2021.

[25] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[26] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for vanets: A deep learning-based distributed sdn approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. early access, doi: 10.1109/TITS.2020.3027390, 2020.

[27] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 1071–1086, 2016.

[28] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[29] Y. Zhao, J. Chen, Q. Guo, J. Teng, and D. Wu, "Network anomaly detection using federated learning and transfer learning," in *International Conference on Security and Privacy in Digital Economy*. Springer, 2020, pp. 219–231.

[30] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[31] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for internet of things networks: A particle deep framework," *Future Generation Computer Systems*, vol. 110, pp. 91–106, 2020.

[32] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," *IEEE Access*, vol. 8, pp. 217 463–217 472, 2020.

[33] Q. Qin, K. Poularakis, K. K. Leung, and L. Tassiulas, "Line-speed and scalable intrusion detection at the network edge via federated learning," in *2020 IFIP Networking Conference (Networking)*. IEEE, 2020, pp. 352–360.

[34] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, 2021.

[35] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, and K. J. Wu, "Enhancing iot anomaly detection performance for federated learning," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2020, pp. 206–213.

[36] N. A. A.-A. Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated mimic learning for privacy preserving intrusion detection," in *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2020, pp. 1–6.

[37] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi, "Fedmccs: multi-criteria client selection model for optimal iot federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4723–4735, 2020.

[38] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot," in *2020 IEEE*

14th International Conference on Big Data Science and Engineering (BigDataSE). IEEE, 2020, pp. 88–95.

[39] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning attacks on federated learning-based iot intrusion detection system," in *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, 2020, pp. 1–7.

[40] T. V. Khoa, Y. M. Saputra, D. T. Hoang, N. L. Trung, D. Nguyen, N. V. Ha, and E. Dutkiewicz, "Collaborative learning model for cyberattack detection systems in iot industry 4.0," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[41] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-iiot: A robust federated malware detection architecture in industrial iot," *IEEE Transactions on Industrial Informatics*, 2020.

[42] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lv, "Fleam: A federated learning empowered architecture to mitigate ddos in industrial iot," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[43] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, 2020.

[44] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyber-physical systems," in *28th International Conference on Computer Communications and Networks (ICCCN)*, 2019, pp. 1–8.

[45] B. Cetin, A. Lazar, J. Kim, A. Sim, and K. Wu, "Federated wireless network intrusion detection," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 6004–6006.

[46] Y. Sun, H. Esaki, and H. Ochiai, "Adaptive intrusion detection in the networking of large-scale lans with segmented federated learning," *IEEE Open Journal of the Communications Society*, 2020.

[47] Y. Sun, H. Ochiai, and H. Esaki, "Intrusion detection with segmented federated learning for large-scale multiple lans," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[48] K. Li, H. Zhou, Z. Tu, W. Wang, and H. Zhang, "Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning," *IEEE Access*, vol. 8, pp. 214 852–214 865, 2020.

[49] X. Hei, X. Yin, Y. Wang, J. Ren, and L. Zhu, "A trusted feature aggregator federated learning for distributed malicious attack detection," *Computers & Security*, vol. 99, p. 102033, 2020.

[50] Z. Li, J. Chen, J. Zhang, X. Cheng, and B. Chen, "Detecting advanced persistent threat in edge computing via federated learning," in *International Conference on Security and Privacy in Digital Economy*. Springer, 2020, pp. 518–532.

[51] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021.

[52] I. Aliyu, M. C. Feliciano, S. Van Engelenburg, D. O. Kim, and C. G. Lim, "Blockchain-based federated forest for sdn-enable in-vehicle network intrusion detection system," *IEEE Access*, 2021.

[53] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys Tutorials*, pp. early access, 10.1109/COMST.2021.3 058 573, 2021.

[54] Y. Chen, J. Zhang, and C. K. Yeo, "Network anomaly detection using federated deep autoencoding gaussian mixture model," in *International Conference on Machine Learning for Networking*. Springer, 2019, pp. 1–14.

[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 807–814.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Segun I. Popoola received the B.Tech. degree in electronic and electrical engineering from the Ladoke Akintola University of Technology, Ogbomoso, Nigeria in 2014, and the M.Eng. degree in information and communication engineering from the Department of Electrical and Information Engineering, Covenant University, Ota, Nigeria in 2018. He is currently pursuing the Ph.D. degree with the Department of Engineering, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, U.K. He was a Lecturer with the Department of Electrical and Information Engineering, Covenant University. His research interests include wireless communications, machine/deep learning, cybersecurity, and the Internet of Things. He is a Registered Engineer with the Council for the Regulation of Engineering in Nigeria (COREN).

Ruth Ande received the B.Eng and M.Eng degrees from the department of electrical and electronic engineering at the University of Manchester Institute of Science and Technology, UK, and the PhD degree from the department of engineering at Manchester Metropolitan University, UK. Following the completion of her B.Eng and M.Eng, she started her engineering career as a Hardware Design Engineer, specialising in asynchronous system on chip design. From there, she moved to lead the QoS team. In 2011, she joined RAATek as CTO specialising in the application of IoT within buildings. She has led a number of projects in the areas of IoT, AI and cybersecurity.

Bamidele Adebisi (Senior Member, IEEE) received the B.S. degree in electrical engineering from Ahmadu Bello University, Zaria, Nigeria, in 1999, the M.S. degree in advanced mobile communication engineering, and the Ph.D. degree in communication systems from Lancaster University, Lancaster, U.K., in 2003 and 2009, respectively. He was a Senior Research Associate with the School of Computing and Communication, Lancaster University, from 2005 to 2012. He joined Manchester Metropolitan University, Manchester, U.K., in 2012, where he is currently a Professor in electrical and electronic engineering. He has been involving in several commercial and government projects focusing on various aspects of wireline and wireless communications. He is particularly interested in the research and development of communication technologies for electrical energy monitoring/management, transport, water, critical infrastructures protection, home automation, the IoTs, and cyber physical systems. He has several publications and a patent in the research area of data communications over power line networks and smart grid. He is a member of the IET.

Mohammad Hammoudeh (Senior Member, IEEE) received his BSc in Computer Communications in 2004 (Arts Sciences & Technology University, Lebanon), his MSc in Advanced Distributed Systems in 2006 (University of Leicester, UK) and his PhD in Computer Science in 2008 (University of Wolverhampton, UK). He is a Professor (Chair) of Cyber Security in the Department of Computing and Mathematics at the Manchester Metropolitan University. Mohammad heads the CfACS Internet of Things Lab he founded in 2016 where he leads a multi-disciplinary group of research associates and PhD students. From this he established the Lab as a leading research hub with a broad portfolio of successful, industry-sponsored projects. He has been awarded above £2.5M in competitive research funding as Principal/Co-Investigator for 16 research projects. He has a global collaborative research network spanning the academic community, industry, policy makers and wider technology stakeholders in the field of cybersecurity, the Internet of Things and complex highly decentralised systems. He published over 80 refereed conference papers, over 65 peer reviewed journal articles, and is a successful editor of 4 books and many journal special issues. He is a Senior Member of IEEE and Fellow of the Higher Education Academy UK.

Guan Gui (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012.,From 2009 to 2014, he joined Tohoku University as a Research Assistant and a Post-Doctoral Research Fellow, respectively. From 2014 to 2015, he was an Assistant Professor with Akita Prefectural University. Since 2015, he has been a Professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 200 IEEE journal/conference papers and won several best paper awards, such as ICC 2017, ICC 2014, and VTC 2014-Spring. His recent research interests include artificial intelligence, deep learning, non-orthogonal multiple access, wireless power transfer, and physical layer security. He received the IEEE Communications Society Heinrich Hertz Award in 2021, the Elsevier Highly Cited Chinese Researchers in 2020, the Member and Global Activities Contributions Award in 2018, the Top Editor Award of IEEE Transactions on Vehicular Technology in 2019, the Outstanding Journal Service Award of KSII Transactions on Internet and Information System in 2020, and the Exemplary Reviewer Award of IEEE Communications Letters in 2017. He was selected as the Jiangsu Specially-Appointed Professor in 2016, the Jiangsu High-level Innovation and Entrepreneurial Talent in 2016, the Jiangsu Six Top Talent in 2018, and the Nanjing Youth Award in 2018. He is serving or served on the editorial boards of several journals, including IEEE Transactions on Vehicular Technology, IEICE Transactions on Communications, Physical Communication, Wireless Networks, IEEE Access, Journal of Circuits, Systems and Computers, Security and Communication Networks, IEICE Communications Express, KSII Transactions on Internet and Information Systems, and Journal of Communication. In addition, he served as the IEEE VTS Ad Hoc Committee Member in AI Wireless, the Executive Chair for IEEE VTC 2021-Fall, the Vice Chair for IEEE WCNC 2021, the TPC Chair for PHM 2021, the General Co-Chair for Mobimedia 2020, the TPC Chair for WiMob 2020, the Track Chair for IEEE VTC 2020-Spring, ISNCC 2020, and ICCC 2020, and the Award Chair for IEEE PIMRC 2019. He served as a TPC member for many IEEE international conferences, including GLOBECOM, ICC, WCNC, PIRMC, VTC, and SPAWC.

Olamide Jogunola (Graduate Student Member, IEEE) received the M.Sc. degree in networking and data communication from Kingston University, London, U.K., in 2015, and the Ph.D. degree in energy transactions in smart grid from Manchester Metropolitan University, Manchester, U.K., in 2019. She is currently a Research Associate with the Department of Engineering, Manchester Metropolitan University, working on Energy-IQ, a U.K.-Canada power forward smart grid demonstrator project. Her research interests include SG, P2P communication technology, the IoT, peer-to-peer energy trading, network optimization, and artificial intelligence for energy market.,Dr. Jogunola was a recipient of the School of Engineering, Manchester Metropolitan University Ph.D. Studentship on an EPSRC U.K.-Korea-funded project: P2P-ETS system. She was previously involved in an Horizon 2020 Smart Cities and Communities programme; Triangulum, funded by the European Commission.