Federated Deep Learning for Collaborative Intrusion Detection in Heterogeneous Networks

Segun I. Popoola[†], Guan Gui[‡], Bamidele Adebisi[†], Mohammad Hammoudeh[§], Haris Gacanin[¶]

[†]Department of Engineering, Manchester Metropolitan University, Manchester, M1 5GD, UK.
[‡]College of Telecommunications and Information Engineering, NJUPT, Nanjing, China.
[§]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, M1 5GD, UK.
[¶]Institute for Communication Technologies & Embedded Systems, RWTH Aachen University, Aachen, Germany.
E-mail: segun.i.popoola@stu.mmu.ac.uk, guiguan@njupt.edu.cn, b.adebisi@mmu.ac.uk,
m.hammoudeh@mmu.ac.uk, harisg@ice.rwth-aachen.de

***Abstract*—In this paper, we propose Federated Deep Learning (FDL) for intrusion detection in heterogeneous networks. Local Deep Neural Network (DNN) models are used to learn the hierarchical representations of the private network traffic data in multiple edge nodes. A dedicated central server receives the parameters of the local DNN models from the edge nodes, and it aggregates them to produce an FDL model using the Fed+ fusion algorithm. Simulation results show that the FDL model achieved an accuracy of $99.27 \pm 0.79\%$, a precision of $97.03 \pm 4.22\%$, a recall of $98.06 \pm 1.72\%$, an F1 score of $97.50 \pm 2.55\%$, and a False Positive Rate (FPR) of $2.40 \pm 2.47\%$. The classification performance and the generalisation ability of the FDL model are better than those of the local DNN models. The Fed+ algorithm outperformed two state-of-the-art fusion algorithms, namely federated averaging (FedAvg) and Coordinate Median (CM). Therefore, the DNN-Fed+ model is preferable for intrusion detection in heterogeneous wireless networks.**

***Index Terms*—intrusion detection, federated learning, deep learning, smart city, heterogeneous wireless networks**

I. Introduction

The communication system in a smart city comprises a number of heterogeneous networks including mobile communication, wireless sensor, Internet of Things (IoT), Industrial IoT (IIoT), and vehicular networks. The distributed and heterogeneous nature of the communication system has made critical infrastructure and services vulnerable to cyber attacks of different forms [1]. In previous works [2]–[6], 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, these centralised DL approach does not preserve the privacy of the network device owners. Also, the classification performance of local DL approach is limited to the network traffic data within a specific application.

Federated Learning (FL) is an advanced Artificial Intelligence (AI) technique which seeks to protect the privacy of participating nodes without a significant compromise in the classification performance and the generalisation ability of the DL models [7]–[10]. Basically, this concept involves the training of local DL models with private data sets, and the aggregation of their parameters in a central cloud server. In recent literature, FL methods have been proposed for intrusion detection in different application scenarios including large-scale local area network [11], [12], vehicular edge network [13], wireless edge network [14], [15], IoT network [16]–[21], Wi-Fi network [22], satellite-terrestrial integrated network [23], industrial cyber-physical system [24], Industrial IoT (IIoT) [25], [26]. Most of these FL methods were developed for homogeneous networks, and they used either federated averaging (FedAvg) [7] or Coordinate Median (CM) [27] fusion algorithm to aggregate the parameters of their local models. However, these state-of-the-art fusion algorithms force heterogeneous networks to find an average solution across all the participating edge nodes, which has adverse effects on the classification performance of the global DL model [28]. They are most suitable for scenarios where the data in the participating nodes are independent and identically distributed (IID). Meanwhile, the network traffic data in heterogeneous networks are non-IID.

Fed+ algorithm can efficiently handle the non-IID network traffic data in heterogeneous wireless networks because it does not require all edge nodes to converge to a single central point [28]. Therefore, in this paper, we propose Fed+ fusion algorithm for Federated Deep Learning (FDL)-based intrusion detection in heterogeneous networks. A Deep Neural Network (DNN) architecture was used for local DL at the edge nodes of the wireless networks. The parameters of the local DNN models are transmitted to a cloud server for aggregation based on Fed+ fusion algorithm. Then, a single global DNN model is sent to all participating edge nodes, where it will be deployed for intrusion detection. We investigated the classification performance and the generalisation ability of the local DNN models, and the global DNN models. Furthermore, we compared the effectiveness of the Fed+ algorithms with two state-of-the-art fusion algorithms, namely FedAvg and CM.

The remaining parts of this paper is organised as follows: in Section II, we describe the system model; in Section III, we provide the details of the proposed FDL model; in Section IV, we present and discuss the simulation results; and finally summarise our findings in Section V.

II. System Model

In this section, we describe the system model of the heterogeneous wireless networks. Fig. 1 shows the architecture of the heterogeneous wireless networks with edge nodes for local DL, and a central cloud server for the aggregation of local DNN models' parameters.

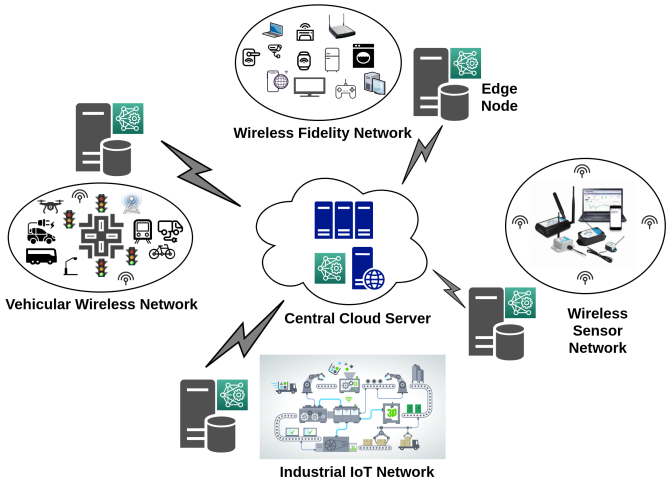

Fig. 1. FDL architecture for intrusion detection in heterogeneous wireless networks.

Four distinct network intrusion detection data sets (NF-ToN-IoT-v2, NF-UNSW-NB15-v2, NF-BoT-IoT-v2, and NF-CSE-CIC-IDS2018-v2)[1] were used to model a typical heterogeneous network in a smart city. These data sets were chosen because they: (a) were derived from NetFlow features, which can be easily extracted from the headers of the network traffic packets; (b) have the same set of network traffic features; (c) were collected from different network configurations; (d) have varieties of complex and popular attack scenarios; and (e) contain sufficient number of benign and malicious network traffic samples. A detailed information about the datasets can be found in [29].

TABLE I
NETWORK TRAFFIC SAMPLES IN WN1

Class	Training	Validation	Testing
Benign	26500	11357	16167
Backdoor	187	80	85
DDoS	19572	8388	12040
DoS	6259	2682	3724
Injection	6726	2882	4078
MITM	58	25	41
Password	10982	4706	6869
Ransomware	33	14	18
Scanning	10508	4503	6324
XSS	23772	10188	14692
Total	104595	44827	64038

In this study, we assume that the network traffic data in NF-ToN-IoT-v2, NF-UNSW-NB15-v2, NF-BoT-IoT-v2, and NF-CSE-CIC-IDS2018-v2 were collected from four wireless networks named WN1, WN2, WN3, and WN4, respectively. WN1 contains nine attack scenarios, namely backdoor, Denial of Service (DoS), Distributed DoS (DDoS), injection, Man-in-the-Middle (MITM), password, ransomware, scanning, and cross-site scripting (XSS). WN2 also contains nine attack scenarios, namely analysis, backdoor, DoS, exploits, fuzzers, generic, reconnaissance, shellcode, and worms. WN3 has four attack classes, namely DoS, DDoS, reconnaissance, and data theft. WN4 data set has six major attack types, namely bot, brute-force, DoS, DDoS, infiltration, and Structured Query Language (SQL) injection. Each of the wireless networks has a training set, a validation set, and a testing set. The number of benign and malicious network traffic samples in the training set, the validation set, and the testing set of WN1-WN4 are presented in Tables I-IV, respectively.

[1]https://staff.itee.uq.edu.au/marius/NIDS_datasets/

TABLE II
NETWORK TRAFFIC SAMPLES IN WN2

Class	Training	Validation	Testing
Analysis	36	15	20
Backdoor	32	14	19
Benign	223364	95727	136660
DoS	328	140	193
Exploits	2884	1236	1782
Fuzzers	1481	635	923
Generic	268	115	193
Reconnaissance	631	271	407
Shellcode	57	24	46
Worms	13	6	19
Total	229094	98183	140262

TABLE III
NETWORK TRAFFIC SAMPLES IN WN3

Class	Training	Validation	Testing
Benign	638	274	373
DDoS	41383	17736	25433
DoS	81876	35090	50038
Reconnaissance	12687	5437	7783
Theft	18	8	7
Total	136602	58544	83634

TABLE IV
NETWORK TRAFFIC SAMPLES IN WN4

Class	Training	Validation	Testing
Benign	75765	32471	46413
Bot	750	321	439
Brute Force-Web	8	4	12
Brute Force-XSS	1	1	2
DDOS attack-HOIC	5347	2292	3298
DDoS attacks-LOIC-HTTP	1412	605	874
DoS attacks-GoldenEye	144	62	86
DoS attacks-Hulk	2120	909	1332
DoS attacks-SlowHTTPTest	59	25	49
DoS attacks-Slowloris	34	14	21
FTP-BruteForce	135	58	74
Infiltration	550	236	301
SQL Injection	5	2	1
SSH-Bruteforce	467	200	240
Total	86797	37199	53142

III. THE PROPOSED DNN-FED+ MODEL

In this section, we provide the details of the proposed DNN-Fed+ model, which comprise of local DL at the edge and the aggregation of the local DNN models' parameters in the cloud.

1) Local Deep Learning in Edge Nodes: Each of the wireless networks is equipped with an edge node for efficient data storage and computation. Four local DNN models were trained and validated with the private data sets in the edge nodes to correctly distinguish between benign and malicious network traffic. The model architecture comprised of an input layer, two 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 packet in the training data. Each of the two hidden layers has u neurons. 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 the successive hidden layer, the output of the preceding hidden layer is transformed as:

$$h_2 = \sigma_h(W_2 h_1 + b_2), \tag{2}$$

Finally, the predicted class label, $\tilde{y}$, is obtained by transforming the output of the last hidden layer:

$$\tilde{y} = \sigma_y(h_2), \tag{3}$$

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, and $m = 2$ for binary classification. The hyperparameters of the local DNN models are presented in Table V.

TABLE V
HYPERPARAMETERS OF DNN MODELS

Hyperparameter	
Input neurons	39
Hidden layers	2
Neurons in each hidden layer	128
Hidden layer activation function	ReLU
Output neuron	1
Output layer activation function	Sigmoid
Optimisation algorithm	Adam
Learning rate	0.0001
Batch size	512
Epochs	10
Loss function	Binary cross-entropy

2) Model Parameter Aggregation in Cloud Server: Fed+ algorithm [28] was used to aggregate the parameters of local DNN models in a central cloud server, as presented in Algorithm 1. The loss minimisation objective of the FDL model is given as:

$$\min_{X} \quad F(X, \alpha) = \frac{1}{N} \sum_{n=1}^{N} f_n(x_n) + \alpha_n B(x_n, C(X)), \tag{4}$$

where $X = x_1, x_2, ..., x_n$ is the local DNN models' parameters, N is the number of participating edge nodes, f_n is the local loss function, B is a distance function, and C is an aggregating function which finds the central point of X. Each participating edge node performs mini-batch stochastic gradient descent locally using Adam optimisation algorithm [30]. The updates of the optimisation process is given as:

$$x_n^{k+1} = x_n^k - \gamma^k [\nabla_n(x_n^k) + \alpha_n^k \nabla B(x_n^k, C(X^k))], \tag{5}$$

where γ is the learning rate, and $k = 1, ..., K$ is the number of communication rounds. The edge nodes initialise and transmit the parameters of their local DNN models, x_n^0, to the central cloud server, which calculates the central value $\tilde{x} \leftarrow C(X^0)$.

Algorithm 1: Fed+ algorithm

Initialization: $K, N, \alpha_n^k, \gamma_n^k, p_1 \in (0,1), p_2 \in Z$
function `localDNNUpdate`$(n, k, \tilde{x})$**:**
  Start with $x_n^k \leftarrow x_n^{k-1}$
  for *epoch* $= 1$ *to* 10 **do**
    for *batch* $= 1$ *to* $\frac{total\_samples}{512}$ **do**
      $x_n^k \leftarrow x_n^k - \gamma_n^k(\nabla f_n(x_n^k) + \alpha_n^k \nabla B(x_n^k, \tilde{x}))$
    end
  end
end function
for $k = 2$ *to* 10 **do**
  for *each edge node* $n \in N$ *in parallel* **do**
    Cloud server transmits $\tilde{x}^{k-1}$ to the edge node.
    $x_n^k \leftarrow$ localDNNUpdate$(n, k, \tilde{x}^{k-1})$
    The edge nodes sends x_n^k to the cloud server.
  end
  $\tilde{x}^k \leftarrow C(X^k)$
end

IV. RESULTS AND DISCUSSION

In this section, we present and analyse the classification performance and the generalisation ability of the FDL models.

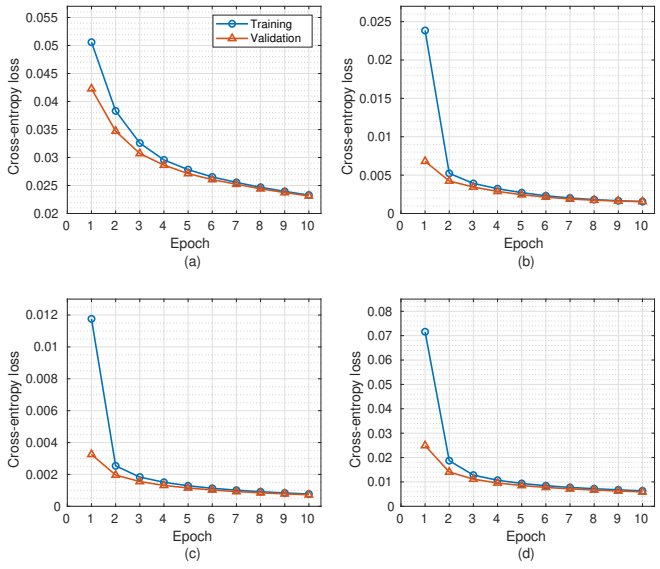

Fig. 2. Cross-entropy loss of local DNN models in (a) WN1, (b) WN2, (c) WN3, and (d) WN4 during training and validation.

First, four local DNN models were trained and validated with the private training and validation data sets in WN1-WN4, respectively. During the training and validation, the cross-entropy loss of the local DNN models were monitored to determine the suitability of the model hyperparameters in Table V. Fig. 2 shows that the cross-entropy loss of local DNN models reduced significantly as the number of epochs increased from 1 to 10. At the end of the 10th iteration of Adam optimisation algorithm, the cross-entropy loss reduced by $83.01 \pm 19.40\%$ and $69.15 \pm 15.94\%$ when the local DNN

models were evaluated with the network traffic samples in the training sets and the validation sets, respectively. A significant loss minimisation during training and validation implies that the local DNN models neither under-fitted the samples in the training sets nor over-fitted the samples in the validation sets.

TABLE VI
CLASSIFICATION PERFORMANCE OF LOCAL DNN MODELS

Training data	Metric (%)	Testing data			
		WN1	WN2	WN3	WN4
WN1	Accuracy	92.47	78.96	29.19	62.83
	Precision	95.96	5.80	99.54	16.83
	Recall	93.87	47.22	29.01	49.12
	F1 score	94.91	10.34	44.92	25.07
	FPR	11.69	20.20	30.03	35.18
WN2	Accuracy	25.23	99.25	0.45	71.04
	Precision	0.00	92.62	-	25.95
	Recall	0.00	77.01	0.00	69.43
	F1 score	0.00	84.10	0.00	37.78
	FPR	0.05	0.16	0.00	28.73
WN3	Accuracy	76.49	65.94	99.82	53.44
	Precision	77.78	2.61	99.92	11.91
	Recall	95.97	33.73	99.90	41.88
	F1 score	85.92	4.84	99.91	18.55
	FPR	81.20	33.21	17.43	44.89
WN4	Accuracy	44.58	97.39	35.92	98.52
	Precision	96.79	16.84	99.38	98.99
	Recall	26.76	0.44	35.86	89.23
	F1 score	41.92	0.87	52.70	93.86
	FPR	2.63	0.06	50.13	0.13

Also, the classification performance of each of the local DNN models was evaluated with the testing data sets in WN1-WN4. The accuracy, precision, recall, F1 score, and FPR of the local DNN models were analysed to determine their generalisation ability. Table VI shows that the local DNN models had a good classification performance when the training data and the testing data were taken from the same network. In this scenario, the local DNN models achieved an accuracy of $97.51 \pm 3.41\%$, a precision of $96.88 \pm 3.30\%$, a recall of $90.00 \pm 9.70\%$, a F1 score of $72.57 \pm 6.61\%$, and a FPR of $25.80 \pm 8.64\%$. On the other hand, when the training data and the testing data were taken from different networks, the local DNN models had a poor classification performance. In this case, the local DNN models achieved an accuracy of $53.45 \pm 27.38\%$, a precision of $37.79 \pm 42.09\%$, a recall of $35.78 \pm 28.67\%$, a F1 score of $26.91 \pm 26.59\%$, and a FPR of $27.19 \pm 24.74\%$. Meanwhile, a significantly high accuracy, precision, recall, and F1 score (the closer to 100%, the better) implies that a larger percentage of the malicious traffic samples were correctly classified. On the other hand, a significantly low FPR (the closer to 0%, the better) means that a larger percentage of the benign traffic samples were correctly classified. We observed that the local DNN models had poor generalisation ability. Therefore, they are not suitable for intrusion detection in heterogeneous wireless networks.

On the other hand, four FDL models were collaboratively developed with the private training data sets in WN1-WN4 using FedAvg, CM, FedAvg+, and CM+ fusion algorithms, respectively. The classification performance of the FDL models was monitored as the number of communication rounds increased from 2 to 10. Fig. 3 shows that the MCC values of the DNN-FedAvg+ and DNN-CM+ models were higher

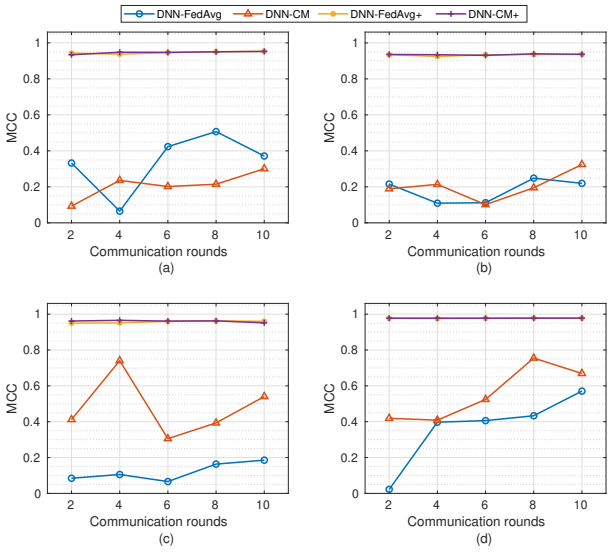

Fig. 3. MCC of FDL models in (a) WN1, (b) WN2, (c) WN3, and (d) WN4.

MCC than those of the DNN-FedAvg and DNN-CM models. Generally, the MCC values of the DNN-FedAvg and DNN-CM models were less than 0.80 but those of the DNN-FedAvg+ and DNN-CM+ models were more than 0.92. Unlike the DNN-FedAvg and DNN-CM models, the MCC of the DNN-FedAvg+ and DNN-CM+ models was relatively stable as the number of communication rounds increased from 2 to 10. When the MCC of a binary classifier is greater than 0.9, it means that there is a high correlation between the actual and the predicted classes of network traffic samples in the heterogeneous wireless networks. Therefore, the FedAvg+ and CM+ fusion algorithms for more suitable for FDL-based intrusion detection in heterogeneous wireless networks compared to the FedAvg and CM fusion algorithms.

Furthermore, the classification performance of the FDL models was evaluated with the testing data sets in WN1-WN4. The accuracy, precision, recall, F1 score, and FPR of the FDL models were analysed to assess their generalisation ability. Table VII shows that the classification performance of the DNN-FedAvg+ and DNN-CM+ models was consistently better than that of the DNN-FedAvg and DNN-CM models in WN1-WN4. DNN-FedAvg+ model achieved an accuracy of $99.27 \pm 0.79\%$, a precision of $97.03 \pm 4.22\%$, a recall of $98.06 \pm 1.72\%$, a F1 score of $97.50 \pm 2.55\%$, and a FPR of $2.40 \pm 2.47\%$. Also, DNN-CM+ model achieved an accuracy of $99.28 \pm 0.79\%$, a precision of $97.15 \pm 3.97\%$, a recall of $98.08 \pm 1.78\%$, a F1 score of $97.57 \pm 2.45\%$, and a FPR of $2.27 \pm 2.50\%$. Higher values of accuracy, precision, recall, and F1 score as well as a lower FPR in WN1-WN4 implies that the DNN-FedAvg+ and DNN-CM+ models has a better generalisation ability than the DNN-FedAvg and DNN-CM models. This further confirmed that the FedAvg+ and CM+ fusion algorithms are efficient for FDL-based intrusion detection in heterogeneous wireless networks.

TABLE VII
CLASSIFICATION PERFORMANCE OF FDL MODELS

Testing data	Metric	FDL models			
		DNN-FedAvg	DNN-CM	DNN-FedAvg+	DNN-CM+
WN1	Accuracy	68.49	73.16	97.98	97.99
	Precision	88.03	78.64	98.20	98.09
	Recall	67.67	87.96	99.11	99.24
	F1 score	76.14	83.00	98.65	98.66
	FPR	29.09	70.64	5.38	5.73
WN2	Accuracy	78.47	85.12	99.65	99.66
	Precision	8.50	10.86	90.20	90.66
	Recall	65.99	58.85	96.89	96.85
	F1 score	14.92	17.96	93.40	93.63
	FPR	21.20	14.19	0.28	0.26
WN3	Accuracy	74.03	98.11	99.96	99.96
	Precision	99.97	99.98	99.98	99.99
	Recall	73.93	98.12	99.98	99.98
	F1 score	84.27	99.04	99.98	99.98
	FPR	4.08	3.75	3.91	3.06
WN4	Accuracy	70.36	79.72	99.50	99.51
	Precision	29.15	42.55	99.72	99.86
	Recall	83.95	96.89	96.28	96.24
	F1 score	43.02	57.74	97.97	98.02
	FPR	31.61	22.77	0.04	0.02

V. CONCLUSION

In this paper, we proposed FDL for intrusion detection in heterogeneous wireless networks. First, local DNN models were trained and validated with private data sets in multiple edge nodes to correctly distinguish between benign and malicious network traffic. Then, the parameters of the local DNN models were transmitted from the edge nodes to a central server. Global DNN models, DNN-FedAvg+ and DNN-CM+, were developed by aggregating the parameters of the local DNN models using Fed+ fusion algorithms i.e., FedAvg+ and CM+, respectively. Also, we investigated the classification performance and the generalisation ability of the local DNN models, and the global DNN models. Furthermore, we compared the effectiveness of the Fed+ algorithms with two state-of-the-art fusion algorithms, namely FedAvg and CM. Simulation results showed that the local DNN models had a high attack detection rate. However, their false alarm rate was high and their generalisation ability was poor. On the other hand, the DNN-FedAvg+ and DNN-CM+ models achieved a higher classification performance and a better generalisation ability.

ACKNOWLEDGMENT

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).

REFERENCES

[1] Y. Wei, S. Zhou, S. Leng, S. Maharjan, and Y. Zhang, "Federated learning empowered end-edge-cloud cooperation for 5g hetnet security," *IEEE Network*, vol. 35, no. 2, pp. 88–94, 2021.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021.

[9] 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*, 2021.

[10] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[11] 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*, vol. 2, pp. 102–112, 2021.

[12] 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.

[13] 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*, 2021.

[14] 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.

[15] 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.

[16] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] T. T. Huong, T. P. Bac, D. M. Long, B. D. Thang, N. T. Binh, T. D. Luong, and T. K. Phuc, "Lockedge: Low-complexity cyberattack detection in iot edge computing," *IEEE Access*, vol. 9, pp. 29 696–29 710, 2021.

[22] 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.

[23] 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.

[24] 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*, vol. 17, no. 8, pp. 5615–5624, 2021.

[25] P. Zhou, "Federated deep payload classification for industrial internet with cloud-edge architecture," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2020, pp. 228–235.

[26] 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.

[27] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[28] P. Yu, L. Wynter, and S. H. Lim, "Fed+: A family of fusion algorithms for federated learning," *arXiv preprint arXiv:2009.06303*, 2020.

[29] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," *arXiv preprint arXiv:2011.09144*, 2020.

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