Building a Robust Federated learning based Intrusion Detection System in Internet of Things

Afrooz Rahmati, Afra Mashhadi, and Geethapriya Thamilarasu

Computing and Software Systems, University of Washington Bothell, Bothell, WA USA

Abstract. The Internet of Things (IoT) has emerged as the next big technological revolution in recent years with the potential to transform every sphere of human life. As devices, applications, and communication networks become increasingly connected and integrated, security and privacy concerns in IoT are growing at an alarming rate as well. While existing research has largely focused on centralized systems to detect security attacks, these systems do not scale well with the rapid growth of IoT devices and pose a single-point of failure risk. Furthermore, since data is extensively dispersed across huge networks of connected devices, decentralized computing is critical. Federated learning (FL) systems in the recent times has gained popularity as the distributed machine learning model that enables IoT edge devices to collaboratively train models in a decentralized manner while ensuring that data on a user's device stays private without the contents or details of that data ever leaving that device. In this paper, we propose a federated learning based intrusion detection system using LSTM Autoencoder. The proposed technique allows IoT devices to train a global model without revealing their private data, enabling the training model to grow in size while protecting each participants local data. We conduct extensive experiments using the BoT-IoT data set and demonstrate that our solution can not only effectively improve IoT security against unknown attacks but also ensure users data privacy.

Keywords: Internet of Things, security, Intrusion Detection system, Federated learning, Deep embedded clustering

1 Introduction

The rapid proliferation of Internet of Things (IoT) devices has ushered in a new era of connectivity and convenience [1]. However, this interconnected ecosystem also introduces a myriad of security challenges that pose significant threats to the integrity and privacy of IoT-enabled systems. The inherent characteristics of IoT, such as the diverse nature of devices, resource constraints, and the vast scale of deployments, create a fertile ground for potential security vulnerabilities. Traditional security measures, designed for conventional computing environments, often fall short in addressing the unique challenges presented by the IoT paradigm.

David C. Wyld et al. (Eds): IOTBS, NLTM, AIMLA, DBDM -2024
pp. 01-13, 2024. CS & IT - CSCP 2024
DOI: 10.5121/csit.2024.140201

In response to these challenges, intrusion detection emerges as a critical component in fortifying the security posture of IoT ecosystems. By enhancing the detection and mitigation of security breaches, organizations can not only protect sensitive data but also ensure the uninterrupted functionality of IoT applications. Traditional intrusion detection methods often struggle to adapt to the dynamic and heterogeneous nature of IoT environments. Machine learning, with its ability to discern patterns and anomalies, presents a promising approach to augment the security posture of IoT deployments. ML models, trained on diverse datasets, can learn to distinguish normal from malicious behavior, enabling more accurate and proactive threat detection [2].

Deep learning (DL), a subset of ML characterized by neural networks with multiple layers, offers enhanced capabilities for feature extraction and representation learning. Existing research discusses how deep learning algorithms can be tailored to the intricacies of IoT traffic patterns, enabling the identification of subtle and complex security anomalies. The integration of ML and DL in intrusion detection for IoT is not without challenges, including the scarcity of labeled datasets and the resource constraints of IoT devices. Moreover, the usage of third-party servers for machine learning training may compromise data privacy, as the training data contain sensitive personal information such as the patient's demographic info, and might lead to data breaches [1]. It is thus critical to create novel methodologies to implement efficient and privacy-enriched IoT networks and applications.

Federated learning (FL) is emerging as a valuable approach in addressing the challenges associated with intrusion detection in Internet of Things (IoT) environments. Federated learning allows models to be trained collaboratively without exposing raw data, as only model updates, in the form of gradients, are shared between devices and the central server. This preserves the confidentiality of individual device data while still allowing the model to learn global patterns and trends. Federated learning provides resilience to heterogeneity in IoT by allowing devices to contribute to the learning process based on their capabilities. Since federated learning only transmits model updates, the amount of data exchanged is reduced. This is particularly advantageous in IoT environments where bandwidth constraints and latency issues may be prevalent.

In this paper, we propose a federated learning-based intrusion detection system that relies on the deep embedded clustering model based on LSTM autoencoder and clustering layer. The suggested method enables IoT devices to train a global model without disclosing their private data, allowing the training model to expand in size while protecting each participant's local data.

The contributions of this paper are as follows. First, we propose an LSTM-Autoencoder-clustering model that can identify attacks in IoT attacks under the centralized environment. Second, we prove that our clustering model can run in

federated learning setting and achieve high performance. Thirdly, we compare our result with other models and demonstrate the effectiveness of our solution.

2 Related Work

In the last few years, research on developing intrusion detection systems (IDS) for securing Internet of Things is gaining increased attention. In this section, we will discuss some of the existing approaches in this field of research.

2.1 Centralized Intrusion Detection

Centralized IDS assume the presence of a central server to analyze data and identify threats. In [2], authors presented a new method based on the Long Short Term Memory (LSTM) autoencoder and the One-class Support Vector Machine (OC-SVM) to detect anomaly-based intrusions in an unbalanced dataset by training the models with only samples of normal classes. They evaluated their method on InSDN dataset and achieved accuracy of 0.90% with precision and recall value close to 93%. Zong et al. [3], developed a deep Autoencoding Gaussian Mixture Model (DAGMM) for unsupervised anomaly detection. DAGMM optimizes the parameters of the deep autoencoder and the mixture model concurrently, using a separate estimate network to speed up the mixture model parameter learning. koroniotis et al. [4] observed SVM, RNN and LSTM-RNN model on the Bot-IoT dataset with all 46 features as well as the top 10 features.Their results indicate that the SVM classifier required the most training time when all features were used, but had the highest accuracy and recall rates. In [5] a C5 classifier and a One Class Support Vector Machine classifier are combined to form a Hybrid Intrusion Detection System (HIDS). This method integrates signature-based IDS for detection of well-known attacks with a behavioral IDS for detection of zero-day attacks. They evaluate their model on 13 features of Bot-Iot dataset with the highest accuracy of 99.97% from the combination of both signature-based and behavioral IDS.

Even though these models have been used successfully for IDS, they usually require a central server to process the data collected from all network users. However, a single-point IDS server may compromise data privacy. Due to the dispersal of data across various sources and the high cost of obtaining data at a central node, a centralized IDS may not always be practical in an IoT environment.

2.2 Federated Learning based Intrusion Detection

Federated learning, first proposed by Google, as a collaborative machine learning method that train models locally on each client device and aggregate only model parameters on a central server. This method ensures that the data stays on client device and remain private. Recent research in federated learning-based intrusion

detection systems demonstrates promising advancements in addressing the unique challenges posed by IoT environments [6].

In [7], the authors propose DIoT, an autonomous self-learning system based on FL for identifying IoT devices compromised with Mirai malware in the IoT of SOHO network. DIoT is composed of a security gateway and IoT security services. A security gateway, which monitors the network for suspicious activity, is integrated with the anomaly detection component. IoT security services keep track of device-specific anomaly detection models and aggregate model weights updates from IoT devices in a central repository. The device-specific anomaly detection gets existing anomaly detection models from the repository and allows network traffic monitoring when new devices are added to the IoT network. According to the evaluation findings, false alarms are minimized in the detection of attacks. However, the method was confined to Mirai attack types and did not include the construction of a deep learning framework in FL domain.

Liu et al. [8] integrate federated learning with deep anomaly detection, in which a convolutional neural network model with long short term memory is developed to improve detection accuracy. The gradient compression is also used in this FL to reduce communication costs and improve communication quality. Zhao et al. [9] developed MT-DNN-FL, which is a multi-task deep neural network in federated learning that can handle network anomaly detection, VPN (Tor) traffic recognition, and traffic classification tasks all at the same time.

FedAGRU, a FL-based Attention Gated Recurrent Unit, is proposed in [10]. FedAGRU is a federated averaging method that has been improved to detect poisoning attacks and minimize contributing updates for a highly efficient global model with low communication costs. Chen et al. [10] suggest a FL-based method specifically for wireless intrusion detection (WID) using the awid dataset.

In order to discover malicious devices in industrial control systems, Wang et al. [11] offer an anomaly detection approach based on a composite auto encoder model, in which anomalies are identified based on error distribution. In [12], they present a Smart Manufacturing architecture that makes use of FL for anomaly detection. Their proposed architecture has three main components: Factory sites, which are controlled by edge servers for collecting anomalies, edge device that receives sensor data and performs anomaly detection and global unit processing, cloud server, to aggregate data from various edge devices. Saharkhizan et al. [13] proposed a deep neural network intrusion detection system using six layers of LSTM. The model's efficiency is confirmed by the evaluation results, however, it is confined to a centralized form of ML. The Federated Averaging (FedAVG) method was developed in [14], which combines local SGD (stochastic gradient descent) on each client with model averaging on a server. Model averaging is similar to dropout training in that it creates an average model based on the common parameters of various clients. The major restriction in the federated learning scenario is communication cost.

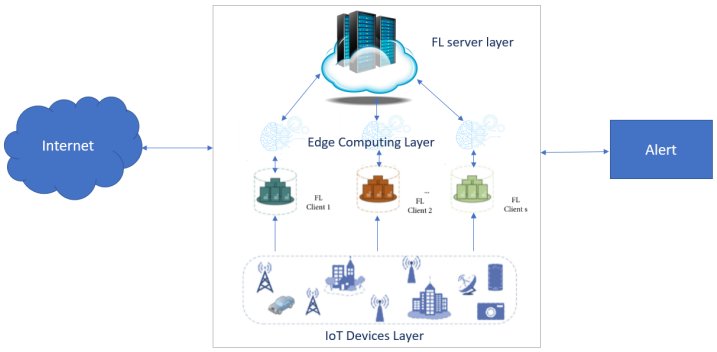

Fig. 1. Proposed Federated learning architecture

When compared to synchronous stochastic gradient descent, the FedAVG method requires 10-100 times the amount of communication cycles.

The authors of [15] utilized convolutional neural networks and LSTM to detect abnormalities in times series data collected by industrial IoT (IIoT) sensors. FL is implemented using the deep learning frameworks, and a gradient compression approach is suggested to increase communication efficiency. In this work, our goal is to develop more efficient and scalable federated learning algorithms tailored for the constraints of IoT environments.

3 Proposed Federated learning based IDS

In this section we describe our proposed Federated learning based intrusion detection model for Internet of Things. We discuss the system architecture and the design elements of our local and global model as well as each network element along with its interactions. Figure 1 depicts the architecture of our proposed framework with i number of clients, a shared model, and a central FL server entity. The subsequent sections contain design specifics for each component of the proposed architecture.

3.1 Server

The central entity, the FL server, is mainly responsible for the coordination among FL clients to train the global model. First, it initializes the pre-training weights and sets the global model hyper-parameters. These weights and parameters are then sent to a set of selected clients for training purposes. Once clients return the trained weights, the global aggregation strategy, FedAVG [14] which simply aggregates the weights from each client equally, is applied. This process is repeated until the desired accuracy is achieved. Finally, the server transfers optimum model weights to all other clients in the network.

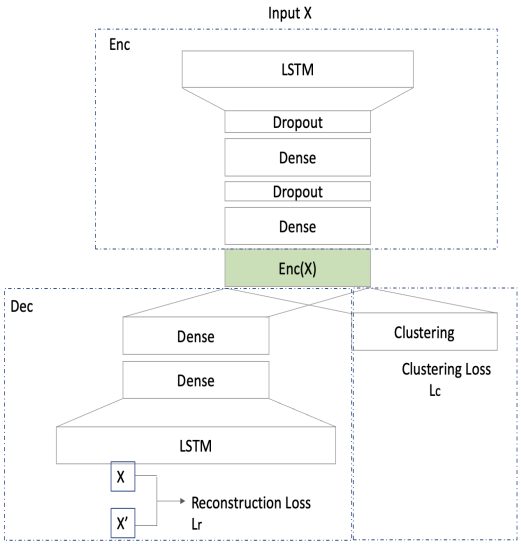

Fig. 2. Deep embedded clustering model

3.2 Client

IoT devices are generally low-resource devices with limited capability for computing machine learning models. On the other hand, due to the heterogeneous nature of IoT devices, we execute the training model on a device in the same network to avoid the complexity of implementing ML applications on them. These clients monitor the traffic of the IoT devices within the same network and apply pre-processing techniques, described in Section 4.2, to prepare the data for the local training on the client. After running a specific number of local rounds, only model weights are transferred to the server for aggregation process.

3.3 Federated learning Model

In this research, we combine the benefits of federated learning with deep embedded clustering. Deep embedded clustering (DEC) is a machine learning paradigm that combines deep learning with clustering [16, 17]. The main idea is to learn a deep representation of the input data such that it facilitates effective clustering in the embedded space. This is typically achieved by incorporating a clustering objective into the training of a deep neural network. Earlier research shows that deep embedded learning model performs well in federated learning settings specially where data is highly non-iid [18].

We design and develop our IDS for IoT using federated learning combined with deep embedded clustering model. IoT Devices in the federated learning setting

use deep embedded clustering to learn representations of their local data that are suitable for intrusion detection. The clustering layer in the deep embedded clustering model help devices identify normal and anomalous patterns in their local data. Aggregated model updates from all devices contribute to the improvement of the global intrusion detection model, making it robust and adaptive to diverse IoT environments. Our solution works by clustering the clients' data in an unsupervised manner in two phases. First, model parameters are pre-trained by a deep autoencoder model, and second, a clustering layer is applied to optimize the learned encoding into the cluster assignment. We implement the autoencoder part of the DEC model based on the LSTM autoencoder(AE) followed by the clustering layer. Figure 2 presents the overall model consisting of one LSTM layer, two Dense layers, and a Dropout layer.

LSTM autoencoder (AE) is a type of unsupervised neural network that uses back propagation to generate output vectors that are similar to the inputs. It compresses the input data into a lower-dimensional space before reconstructing the original data from that representation. To learn the nonlinear representation of the data, it employs a nonlinear activation function and multiple layers. The LSTM AE is made up of an encoder component $f_w(.)$ and a decoder component $g_w{}'(.)$. The objective of our LSTM AE is to minimize the mean squared error (MSE) between input (x) and output (x') where $x' = g_w(f_w(x))$. The reconstruction loss is measured by Equation 1.

$$L_r = \frac{1}{n}\sum_{i=1}^{n} \|G_{\omega'}(F_{\omega}(x_i)) - x_i\|^2 \tag{1}$$

where L_r represents the reconstruction loss, n is the number of records and X_i is the ith input.

The clustering layer is coupled to the embedding layer and converts the learned representation (z_i) of the input traffic data (x_i) into soft labels. This layer iteratively fine-tunes and minimizes the Kullback–Leibler (KL) divergence between the distribution of soft labels and a predetermined target distribution as shown in Equation 2.

$$L_c = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{2}$$

where q_{ij} denotes the similarity of the embedded points z_i, and cluster center μ_j formulated in Equation 3.

$$q_{ij} = \frac{\sum_j 1 + ||z_i - \mu_j||^2}{1 + ||z_i - \mu_j||^2} \tag{3}$$

The target distribution p_{ij} is calculated based on Equation 4.

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{4}$$

The goal of the entire model is to minimize a loss function (L) that is the weighted sum of the reconstruction and clustering losses (L_r, L_c) as presented in Equation 5.

$$L = \alpha L_r + \gamma L_c \tag{5}$$

So, if we consider $\alpha = 1$ and $\gamma = 0$, we will get the same result as the LSTM AE, and if we choose $\alpha = 0$ and $\gamma = 1$, we will get clusters corresponding to random weights. In this paper, we report the results of $\alpha = 1$ while varying γ.

4 Experimental Setup

In this section we describe our experimental environment, hyper-parameter settings, and dataset used to implement our proposed solution. we trained our model 2.40 GHz Quad Core Intel Core i7 CPU with 4 threads and 8 GB RAM and used Python 3.6 on Anaconda Jupyter for development. We implemented our federated learning framework using Flower architecture [19]. Table 1 shows the hyper-parameters used in our simulation setting.

Table 1. Simulation Setting

parameters	Values
optimizer	Adam
learning rate	0.0001
batch size	64
epochs	1000
activation function	Tanh, Relu
γ	1
timesteps	1
features	23
clustering loss	kld
Autoencoder loss	MSE
Number of clients	10

4.1 Dataset

In this study, we use Bot-IoT data set to evaluate the proposed intrusion detection system. Bot-IoT dataset includes both normal and attack traffic from IoT devices,

collected via Nod-Red with 72,000,000 records. Various attack classifications, such as DoS, DDoS, and Reconnaissance are included in the IoT-Bot. The attacks come from both internal and external networks to simulate real-world attack scenarios. It contains the most variance in features compared to the other IoT device data sets with 46 features such as Transaction state, Traffic category and duration. In all of our experiments, we split the data 80% for training and 20% for testing, yielding 1,048,576 training and 733,705 testing records.

4.2 Pre-processing

In this paper, we only focus on binary classification of the intrusion detection system other than classifying the attack types. Anomaly traffic data refers to observations that belong to any attack class. The initial step is to pre-process data for later training purpose. We follow below stages to prepare our data:

- The resulting dataset includes device information such as source and destination IP addresses, Source MAC, Destination MAC and etc. We delete these features as these data would be specific to each device and might result in over-fitting of our model. The final processed dataset contains 35 features.
- We apply normalization on data to make them lie between 0 and 1 range.
- To transform the labeled string to numerical values, we employ one-hot encoding. Binary classification of 1 and 0 is used to demonstrate malicious and normal traffic respectively.

5 Evaluation Metrics

In this section we describe the evaluation metrics used to assess the performance of our proposed model. The clustering accuracy, precision and recall used to compare our model effectiveness with previous work. In addition, we used unsupervised metrics to observe our model NMI (normalized mutual information) and ARI (Adjusted Randomized Index).

We use accuracy metric to classify the effectiveness of the proposed machine learning mode. Accuracy is defined as,

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

where TP refers to true positive, FP refers to false positive, TN is true negatives and FN is false negatives. We also used recall, precision and F1 score to asses our purposed model performance. Recall measuress how many correct positive predictions were produced out of all possible positive predictions. Recall is calculated as,

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

We also evaluate precision, which is defined as positive class predictions that actually belong to the positive class.

$$Precision = \frac{TP}{TP + FN} \tag{8}$$

F1-Measure generates a single score that accounts for both precision and recall mean in a single number and defined as,

$$F1 = 2. \frac{Precision \,.\, recall}{Precision + Recall} \tag{9}$$

6 Results

In this section we present the experimental results of our proposed framework. The model clustering quality assessed by calculating clustering accuracy (ACC), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). The precision, recall, NMI and ARI are all shown in Table 2 for different number of nodes. It is clear that NMI and ARI are both high, indicating that our model performs with high accuracy.

Table 2. The accuracy, precision, recall, NMI, and ARI of our approach on the Bot-IoT dataset with varying number of client are compared. This is the result for a global server round of 20 and client local epochs of 4 with $\gamma = 5$.

No. of Clients	Accuracy	Precision	Recall	NMI	ARI
10	99.99	99.99	1	84	88

In Bot-IoT dataset, the distribution of normal data versus attack data is less than 1%. As a result, we employ a random sampling technique for attack traffic. To balance the dataset, we first extract all 9,490 normal traffic records and then randomly select the attack traffics from each file. Figure 3 depicts the final distribution result for the training and testing datasets.

We evaluate our model performance by changing the γ variant. Figure 4 shows the interaction between γ and the two clustering metrics, NMI and ARI. We found that when $\gamma = 0$, the clustering is random with NMI and ARI less than 20%. By increasing $\gamma = 1$ both NMI and ARI improved up to 84% and 88% respectively. This shows that our system achieved the highest clustering performance at $\gamma = 3$ on the full-feature Bot-IoT dataset. We compare the training Accuracy and NMI when the clustering and reconstruction losses are both taken into account and observed the result by changing the γ value from 0 to 10. We observe that when $\gamma = 0$, the model works solely based on only LSTM AE without any optimization by KL

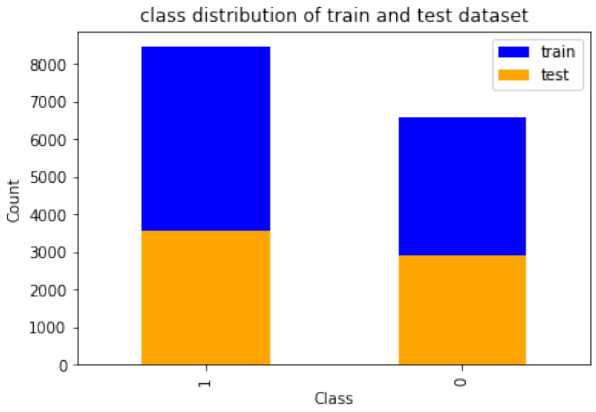

Fig. 3. Distribution of attack and normal traffic (attack=1 , Normal=0) for balanced dataset

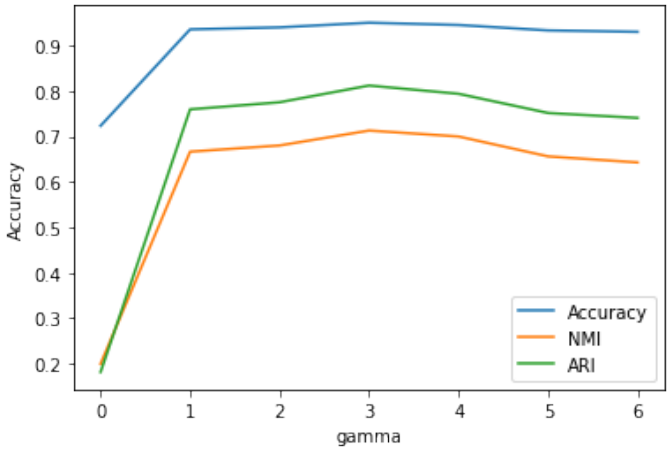

Fig. 4. Interactions between γ, NMI, ARI and Accuracy

divergence. At this stage the accuracy is 43%, However by increasing γ, training accuracy, test accuracy, NMI and ARI are improving and reaching to 91%.

Table 3 provides a performance comparison of our approach with other state-of-art studies. In particular, we compare our federated learning IDS using LSTM-Autoencoder with existing centralized IDS systems and federated learning based systems. We evaluate different machine learning techniques including SVM, LSTM, KNN, NB. Results show that in comparison to these current techniques, our proposed model using LSTM-AEC in federated setting achieved higher accuracy with limited number of clients, demonstrating the effectiveness of our solution.

Table 3. Performance Comparison of Proposed Model with Existing Research

Author	Type	Approach	Accuracy
koroniotis et al. 2018[4]	Centralized	SVM	99.98
		RNN	97.90
		LSTM	98.05
Al-Hawawreh et al. 2021[20]	Centralized	KNN	84.35
		NB	72.98
		LR	88.09
		DPE_TIDD	99.42
Weinger et al. 2020[21]	FL - 5 nodes	NONE Sampling	94.74
		RAND	94.65
		SMOTE	91.97
		ADASYN_TIDD	87.91
Our Approach	FL - 10 nodes	LSTM-AEC	99.998

7 Conclusion

In this study, we introduced a novel approach to intrusion detection in IoT environments by combining the power of federated learning and deep embedded clustering. The proposed model harnesses the decentralized nature of federated learning to train robust intrusion detection models on resource-constrained IoT devices while preserving user privacy. Specifically, we implemented our model using LSTM-AEC. Simulation results indicate that the proposed solution enhances the model's ability to discern complex patterns within local data, facilitating accurate detection of both known and novel intrusion scenarios.

Further research is warranted to explore scalability issues and optimize the model for even more resource-constrained IoT devices. Additionally, real-world deployment and validation of the proposed federated learning deep clustering model using LSTM-AEC will be crucial to assess its effectiveness in diverse IoT environments.

References

1. 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," vol. 23, no. 3, pp. 1622–1658. [Online]. Available: http://arxiv.org/abs/2104.07914
2. M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. ACM, pp. 37–45. [Online]. Available: https://dl.acm.org/doi/10.1145/3416013.3426457
3. B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL FOR UNSUPERVISED ANOMALY DETECTION," p. 19.
4. 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." [Online]. Available: http://arxiv.org/abs/1811.00701

5. A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," vol. 8, no. 11, p. 1210, number: 11 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/8/11/1210
6. S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions." [Online]. Available: http://arxiv.org/abs/2106.09527
7. T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "D\"IoT: A federated self-learning anomaly detection system for IoT." [Online]. Available: http://arxiv.org/abs/1804.07474
8. Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, "Communication-efficient federated learning for anomaly detection in industrial internet of things," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, ISSN: 2576-6813.
9. Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-task network anomaly detection using federated learning," in *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*. ACM Press, pp. 273–279. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3368926.3369705
10. Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion detection for wireless edge networks based on federated learning," vol. 8, pp. 217 463–217 472, conference Name: IEEE Access.
11. C. Wang, B. Wang, H. Liu, and H. Qu, "Anomaly detection for industrial control system based on autoencoder neural network," vol. 2020, p. 8897926, publisher: Hindawi. [Online]. Available: https://doi.org/10.1155/2020/8897926
12. T. T. Huong, T. P. Bac, D. M. Long, T. D. Luong, N. M. Dan, L. A. Quang, L. T. Cong, B. D. Thang, and K. P. Tran, "Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach," vol. 132, p. 103509. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0166361521001160
13. M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," vol. 7, no. 9, pp. 8852–8859, conference Name: IEEE Internet of Things Journal.
14. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data." [Online]. Available: http://arxiv.org/abs/1602.05629
15. "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," vol. 8, no. 8, pp. 6348–6358. [Online]. Available: http://arxiv.org/abs/2007.09712
16. J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.
17. X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." in *IJCAI*, 2017, pp. 1753–1759.
18. A. Mashhadi, J. Sterner, and J. Murray, "Deep embedded clustering of urban communities using federated learning," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
19. Flower documentation. [Online]. Available: https://flower.dev/docs/index.html
20. M. Al-Hawawreh, N. Moustafa, S. Garg, and M. S. Hossain, "Deep learning-enabled threat intelligence scheme in the internet of things networks," vol. 8, no. 4, pp. 2968–2981, conference Name: IEEE Transactions on Network Science and Engineering.
21. 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)*, pp. 206–213.

© 2024 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license