Cyber Security and Applications 3 (2025) 100068

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models

Babatunde Olanrewaju-George [a,*], Bernardi Pranggono [b,*]

[a] Department of Engineering and Mathematics, Sheffield Hallam University, UK
[b] School of Computing and Information Science, Anglia Ruskin University, UK

ARTICLE INFO

Keywords:
Deep learning
Federated learning
Intrusion detection system
Internet of things
Machine learning

ABSTRACT

The adoption of the Internet of Things (IoT) in our technology-driven society is hindered by security and data privacy challenges. To address these issues, Artificial Intelligence (AI) techniques such as Machine Learning (ML) and Deep Learning (DL) can be applied to build Intrusion Detection Systems (IDS) that help securing IoT networks. Federated Learning (FL) is a decentralized approach that can enhance performance and privacy of the data by training IDS on individual connected devices. This study proposes the use of unsupervised and supervised DL models trained via FL to develop IDS for IoT devices. The performance of FL-trained models is compared to models trained via non-FL using the N-BaIoT dataset of nine IoT devices. To improve the accuracy of DL models, a randomized search hyperparameter optimization is performed. Various performance metrics are used to evaluate the prediction results. The results indicate that the unsupervised AutoEncoder (AE) model trained via FL is the best overall in terms of all metrics, based on testing both FL and non-FL trained models on all nine IoT devices.

1. Introduction

The Internet of Things (IoT) networks face significant security and data privacy challenges due to their inherent characteristics such as heterogeneity, scalability, and resource constraints. These challenges are exacerbated by the fact that IoT devices often collect and transmit sensitive data, making them attractive targets for cyber-attacks. Despite the security and privacy challenges hindering the total adoption of the IoT in our technology-driven society, the benefits of interconnecting devices and sensors to exchange information over the Internet without human intervention outweigh these challenges. To better secure these systems and networks from potential security breaches, efforts are continually being made. Artificial Intelligence (AI) techniques, including Machine Learning (ML) and Deep Learning (DL), have gained widespread adoption and are being used to ensure better security for IoT networks. These AI techniques are particularly suitable for IoT applications because of the massive amount of data generated by IoT devices connected to the internet, and the ability of AI methods to analyze and process these big datasets.

In recent literature, using DL as an example has helped improve response latency in IoT device applications, enhanced energy consumption, and provided more sophisticated protection to these systems [1]. The Convolutional Neural Networks (CNN) and the Long Short-Term Memory (LSTM) algorithms are well-established DL models implemented in many applications. On the other hand, IoT networks themselves are not without challenges. A most frequent issue is the challenge of resource-constrained; in terms of computing capability and power capacity, which can hinder the full deployment of sophisticated intrusion detection system (IDS) on IoT devices. Furthermore, with the growing number of IoT devices connected to the internet, hackers are constantly devising new ways of attack. Therefore, there is a need to develop a more accurate IDS for the IoT using robust techniques like DL models. DL models, with their ability to learn complex patterns and make predictions, are well-suited for detecting sophisticated cyber-attacks that may not be easily identifiable using traditional rule-based systems. Researchers proposed a Federated Learning (FL) intrusion detection scheme that decentralizes the training of the IDS model implemented on the individual connected devices, allowing the possibility of the learning and inference to be done locally on the devices [2]. FL allows for decentralized learning across multiple devices while keeping the data on the original device. This enhances data privacy as sensitive data does not need to be transferred to a central server for model training. By combining FL and DL, we achieve the best of both worlds. FL ensures privacy-preserving model training, while DL provides accurate and robust intrusion detection capabilities. This synergy addresses the unique challenges posed by IoT networks. FL has been used in different fields to provide a more efficient system architecture, such as in cloud computing [3].

Peer review under responsibility of KeAi Communications Co., Ltd.
* Corresponding authors.
E-mail addresses: tjoj0107@gmail.com (B. Olanrewaju-George), bernardi.pranggono@aru.ac.uk (B. Pranggono).

https://doi.org/10.1016/j.csa.2024.100068
Received 9 October 2023; Received in revised form 31 March 2024; Accepted 2 August 2024
Available online 3 August 2024

Several studies have considered the deep autoencoder (AE) model as an unsupervised learning model in FL-based IDS for IoT devices [4]. Also, other researchers have considered a supervised Deep Neural Network (DNN) model for FL-based IDS for IoT devices [5]. However, to the best of our knowledge, no single study brings both the unsupervised and the supervised DL models together under one study over a recent IoT device dataset for a clearer understanding of their performance in IoT applications. The combination of unsupervised and supervised learning models is significant in the context of IDS for IoT due to the following reasons:

- Unsupervised learning models like AEs can learn the normal behavior of IoT devices by reconstructing the input data. They can detect anomalies by identifying data instances that deviate significantly from the normal behavior.
- Supervised learning models can then classify these anomalies into specific types of attacks based on labeled attack data. This combination allows for effective detection and classification of both known and unknown cyber-attacks.

In the context of IDS for IoT, combining unsupervised AEs (for feature learning) with supervised DL models (for classification) within the FL framework ensures robustness, privacy, and accurate intrusion detection.

In this study, we build and evaluate unsupervised and supervised DL models for detecting anomalous events in network traffic through the IoT devices, such as webcams, doorbells, baby monitors, thermostats, and security cameras. Using a publicly available and very detailed dataset that captures recent attack features for IoT intrusion detection studies, the DL models are trained to evaluate the results in the confusion matrix, accuracy, precision, F1-Score, True Positive Rate (TPR), and False Positive Rate (FPR) for detecting anomalous events. This detection will be achieved by combining the efficiency of DL models in handling complex tasks with a FL approach that decentralizes the training process of the intrusion detection models. We hope that this combination of DL methods with FL will improve both efficiency and robustness of IoT devices anomaly detection compared to the non-FL model. Hence, we propose a FL based IDS for IoT using unsupervised and supervised DL models. The unsupervised DL model proposed is the deep AE model. An autoencoder is a neural network that learns to represent its input data in a lower-dimensional space. This is done by first compressing the input data, and then reconstructing it from the compressed representation. The compression process ensures that the autoencoder learns the most important features of the input data, and the reconstruction process ensures that the autoencoder learns the relationships between these features. The supervised DL models proposed is the DNN model with three layers. For comparison, the FL model results will be compared to non-FL model results using several recent IoT devices.

The remainder of this paper is arranged as follows. Section 2 presents related work on IDS for IoT network. Section 3 describes the methods used in the study: the dataset, data pre-processing process, and DL and FL models are discussed. Section 4 explains our experiment setup. The experiment results are discussed and analyzed in Section 5. Finally, Section 6 draws the conclusion.

2. Related work

Meidan et al. proposed an anomaly detection model using deep AE. The study used N-BaIoT dataset. The study showed that the deep AE model performed better than other models (local outlier factor (LOF), one-class SVM, and Isolation Forest) [6].

Intelligent detection of IoT botnet using ML and DL is proposed by Kim et al. [7]. The study used N-BaIoT dataset to evaluate the performance of the proposed method. The study implemented several DNN models: CNN, RNN, and LSTM. The simulation results showed that CNN performed better than other models.

Zhang et al. built a platform based on FL for IoT that has a module for device anomaly data detection and another module for realistic evaluation of FL on IoT devices [8]. The overall design comprises a dataset, model, algorithm, and system design. The software architecture consists of the application layer, the algorithm layer, and the infrastructure layer supporting the implementation of FL on AI-enabled IoT edge devices such as Raspberry Pi. Two recent datasets (N-BaIoT and LANDER) are combined in this work with a Deep AE model for anomaly detection. Results obtained using accuracy, precision and false positive rate as metrics demonstrate the efficacy of FL in detecting a large range of attack types.

Rahman et al. proposed a FL-based scheme for IoT intrusion detection to decentralize the training of the IDS model to be done on the individual connected devices, allowing the possibility of the learning and inference to be done locally on the devices [2]. These help to maintain the privacy of the data exchanged across connected devices and can enhance accuracy by exchanging updates from neighboring devices in the network using a remote server. Results obtained suggest that in terms of model accuracy in detecting anomalies, the centralized system was best. On the other hand, the FL approach could reach similar accuracy with better data privacy compared to the centralized approach.

Khan et al. proposed an IDS based on DL methods to address the susceptibility of the MQTT protocol during communication within IoT devices in a network [5]. Two datasets are combined in this study to evaluate and compare the developed system's performance. Comparison is done with conventional ML models such as the DT, RF, NB, and KNN. Other DL models such as the LSTM, and GRU, are also compared. Results obtained suggests the proposed Deep Neural Network (DNN) model attains the highest accuracy of 97.13 % compared to the LSTM and GRU models in one of the compared datasets.

Attota et al. combined multiple IoT data with FL methods to train an IDS system that detects, classifies and defends against various attacks [9]. Results obtained suggest the approach has higher accuracy when compared to conventional non-FL methods. To test the IoT devices using FL methods, an experiment is performed using the PySyft DL framework with ten vital IoT devices. The dataset used in this study is the lightweight MQTT protocol dataset.

Shahid et al. [10] proposed a similar FL IDS for IoT devices similar to [9]. However, the ML models used are the Logistic Regression (LR) and the Multi-label classification (MLC) model. The NSL-KDD dataset is selected for the study. The FL experiment is performed using Python Libraries and PySyft to create virtual instances of FL clients.

Hezam et al. [11] studied the use of deep learning approaches to detect Botnet attacks in IoT environment. The study implemented three DL algorithms: recurrent neural network (RNN), convolutional neural network, and long short-term memory (LSTM)-RNN to counter distributed denial of service (DDoS) attacks targeting IoT networks. N-BaIoT dataset is used. Results obtained suggests the RNN achieved accuracy of 89.75 %.

Alkahtani and Aldhyani [12] used CNN-LSTM model to botnet attack in IoT applications. The N-BaIoTdataset was used and the experiment results shows that CNN-LSTM shows best performance with accuracies of 90.88 %.

Campos et al. [13] proposed an FL-enabled IDS approach based on a multi-class classifier considering different data distributions for the detection of different attacks in IoT scenarios. This study is implemented using the recent ToN-IoT dataset. An aggregation function known as the Fed+ is proposed against the conventional FedAvg algorithm. Its advantage is that it mitigates the limitation of the FedAvg algorithm caused by convergence issues in scenarios with non-iid and highly skewed data. To deploy the FL framework on real IoT devices, a simulated and distributed testbed called the IBMFL is adopted. Evaluation of results obtained suggests that the Fed+ aggregator performed better than the FedAvg algorithm based on the evaluation with the mixed scenario dataset.

Om Kumar et al. [14] studied the use of recurrent kernel convolutional neural network in IDS for IoT. The study is implemented using

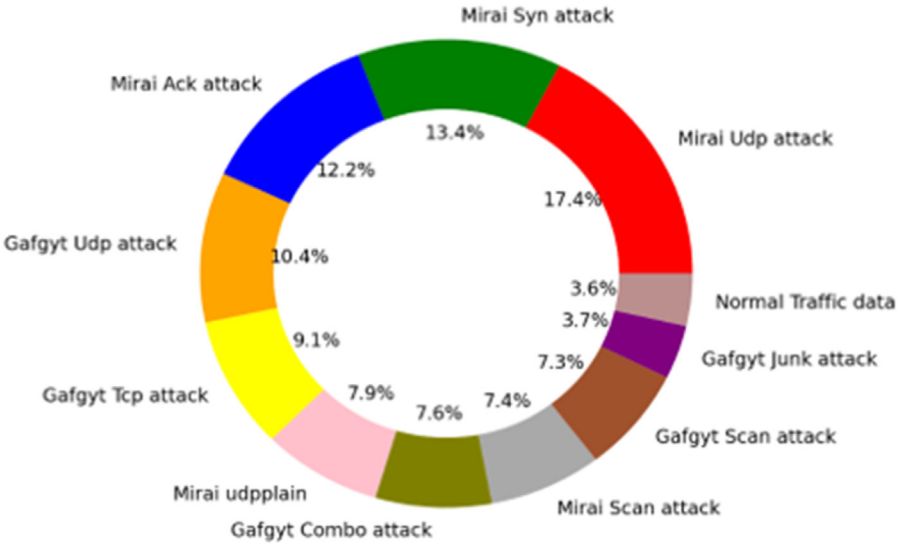

Fig. 1. All nine IoT devices data distribution in a pie chart.

N-BaIoT dataset. The study shows that the RCNN model obtains precision of 92.27 % and F1-score of 94.40 %.

3. Methodologies

This section introduces the dataset used and data pre-processing steps of all the devices data.

3.1. Dataset

The publicly available dataset that captures recent attack features for IoT intrusion detection studies called the N-BaIoT dataset [6] is used. The dataset was developed due to the lack of public botnet datasets for anomaly detection applications. To create this dataset, nine actual traffic data are obtained from commercial IoT devices infected with the two most common IoT-based botnets – Mirai and BASHLITE. The dataset characteristics are multivariate and sequential. It is most suited for classification and clustering tasks. There are 7,062,606 instances with 115 distinct features. The malicious or abnormal data is divided into 10 attacks, carried out by the two powerful botnets that have shown their harmful capabilities in several applications. Therefore, the data comprises ten classes of attacks and one normal or benign class. The N-BaIoT dataset was selected due to its relevance to real-world IoT environments and its comprehensive coverage of various IoT device types and network activities. This dataset contains network traffic data collected from nine different IoT devices, including smart home appliances and wearable devices, under both normal and attack conditions. Therefore, it provides a diverse and representative sample of the network traffic patterns typically encountered in IoT deployments, making it suitable for evaluating the effectiveness of IDS solutions in practical scenarios. Under the attack condition, the nine IoT devices has been infected by Mirai and BASHLITE malware.

3.2. Data analysis

Fig. 1 show the percentage distribution of all nine IoT devices data in terms of the malicious and normal data features.

3.3. Data pre-processing

The data is pre-processed by first separating normal data and abnormal data into their respective columns on the data table. Normal data is represented with 0, while abnormal (attacks) data is represented with 1. The training label is created by setting the normal training set to represent it. Then a random sample of the abnormal data is created and concatenated with the normal testing set created earlier. They are shuffled, and mixed data between the normal testing set and the abnormal data is created. Then a label is created for the mixed data. Subsequently, the normal training set, the normal threshold set, and the mixed data are scaled using a scaler function that returns the data between 0 and 1. The aim is to use the scaled normal threshold set for the computation of a threshold that determines the normal and abnormal observations. The scaled mixed data will be used for the evaluation of the model. Furthermore, the scaled normal threshold set, the mixed data, the mixed data label, and the training label are converted to tensors to fit the data modeling requirements in Python. The data is thereafter loaded onto the device as part of its data modeling process in Python's PyTorch library. A similar data pre-processing module is built to test IoT devices for Federated modeling.

The high-level methodologies diagram of the study is shown in Fig. 2.

3.4. Models development

In our FL framework, the unsupervised AE and other supervised models function as follows:

- Each IoT device trains an AE locally on its data to learn the normal behavior. The AE model parameters are then sent to the central server.
- The server aggregates these parameters to create a global AE model, which is sent back to the devices for further local training. This process is repeated until the global AE model converges.
- The anomalies detected by the AE are then classified into specific types of attacks using supervised models trained in a similar federated manner.

We used the Random Search hyperparameters optimization technique [15] to find the best parameters for compiling and building an efficient DL model. Following that, the best hyperparameters were obtained and utilized to train DL models on the training data. Randomized search for hyperparameter optimization is chosen for its ability to efficiently explore a wide range of hyperparameter configurations without exhaustive grid search. This approach involves randomly selecting hyperparameter values from predefined ranges and evaluating their performance, thus providing a more comprehensive exploration of the hyperparameter space. By leveraging randomness, randomized search avoids being trapped in local optima and can uncover hyperparameter settings that yield superior performance. This method improves the accuracy of

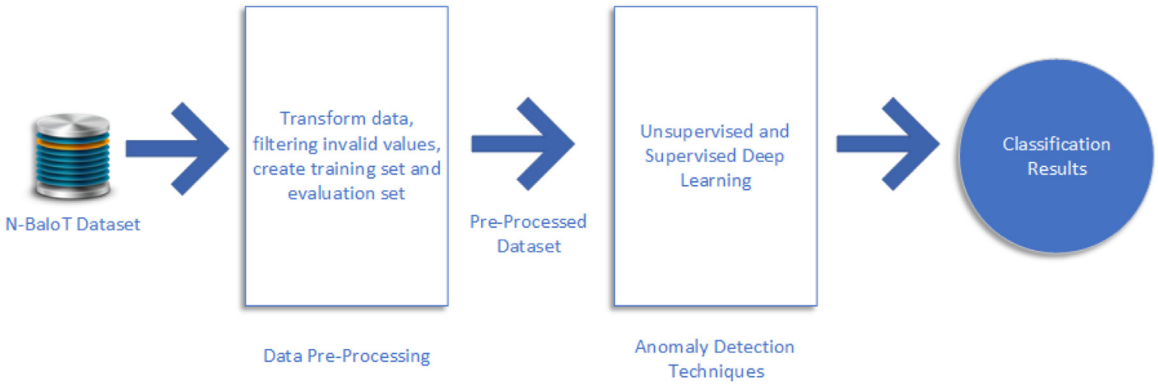

Fig. 2. High-level methodologies diagram of the study.

DL models by fine-tuning hyperparameters to better suit the characteristics of the dataset and the complexity of the model, ultimately leading to improved generalization and predictive performance.

4. Experimental setup

This section describes the methods used to build the unsupervised and supervised DL models. It details the steps used to build the FL algorithm, the hyperparameter optimization performed, training the FL and non-FL DL models, evaluating and testing the trained FL and non-FL models using the nine client IoT devices data, with the Accuracy, Recall, F1-score, True Positive Rate, and False Positive Rate metrics. The experimental setup involved training and evaluating DL models both within the Federated Learning (FL) framework and using traditional non-FL methods. For FL-trained models, each IoT device acted as a client, locally training its model on its data while periodically synchronizing updates with a centralized coordinator. Performance evaluation criteria included various metrics such as accuracy, precision, recall, and F1-score. These metrics were used to assess the models' abilities to accurately detect and classify intrusions while considering factors such as false positives and false negatives.

4.1. Deep learning model

One unsupervised and one supervised DL model are selected and built to evaluate the performance of anomaly detection and FL. The DL models built include the AE and the DNN. Several studies have applied the DNN, CNN, LSTM, and AE models in developing IDS for IoT anomaly detection applications [5,16–18].

However, very few studies have considered both unsupervised and supervised DL models for IDS in IoT applications using FL methods and recent IoT device datasets. AE models are designed and trained to recreate the input vector, contrary to other DL models that predict class labels. Its training is unsupervised, and the purpose of the network is to encode data in both low and high dimensionality spaces and achieve feature extraction. Deep autoencoders can identify and extract the important features from complex data, which can be used for a variety of tasks, such as classification, clustering, and dimensionality reduction. When data from IoT devices is collected and stored in a central server for model training, there is a risk that this data could be stolen or compromised in a data breach. This could expose sensitive information such as the IP addresses, MAC addresses, and open ports of individual IoT devices. This information could then be used by attackers to hack into these devices or to launch denial-of-service attacks. In this study, FL intrusion detection scheme that decentralizes the training of the IDS model implemented on the individual connected devices, allowing the possibility of the learning and inference to be done locally on the devices. When data is not collected and stored in one place, it is difficult to process unlabeled data and train a model on edge devices. To address this challenge, we have developed an unsupervised deep learning approach that uses autoencoders to learn from unlabeled data.

Several variants exist to handle different data patterns and perform specific functions. Its structure consists of the encoding and decoding parts. The former compresses the input data into an encoded representation, while the latter decompresses the knowledge representations and then reconstructs the data back to the original form [6]. Just like the CNN and the RNN algorithms, the DNN algorithm is a special type of NN algorithm that is very efficient in extracting non-linear features from data [19]. It has become very popular in several AI applications, which include image classification tasks, speech recognition, computer vision, etc. [20].

In operation, the DNN architecture receives its data via the input layer. The linear layer is then applied to perform a linear transformation operation on the input data received from the input layer. Other parameters in this layer are the size of the input sample and the size of each output sample. The linear layer learns using additive bias. After this operation, a non-linear activation function layer is applied to the weighted sum of the linear network output values from the previous step, to reduce the linearity and convert it to a non-linear function. Next, at this point to avoid overfitting the model, a batch normalization operation can be added. To further reduce overfitting, a dropout regularizer can also be added. The dropout layer works by discarding some of the function nodes and reducing dependencies between them [19]. Specifically, the dropout layer randomly zeros some of the input tensor elements during training with the probability it obtains from a Bernoulli distribution. Several linear layers with activation functions, dropout layers, and batch normalization layers can be stacked to learn more complex features.

In this study, the proposed network structure is a deep AE with an encoder layer, a hidden layer, and a decoder layer. Fig. 3 shows the deep AE architecture built in this study. Here, the encoder and the decoder parts are seen. In operation, the encoder begins by receiving the normal training set, which in our case is the IoT device's data to train the algorithm. Also, in this study, only the normal features were used to train the detection model. In this study, 115 features are the input to this encoder linear layer. Similar to Medan et al. [6], only 75 % of the input features are allowed at the first linear layer. This is followed by a hyperbolic tangent activation function to add non-linearity to the input data. A second linear layer is added with 50 % of the input features and a hyperbolic tangent function. In the third linear layer, only 33.3 % of the input features are allowed with the hyperbolic tangent function. In the fourth layer, a linear layer with only 25 % of the input features and a hyperbolic tangent function are added. The encoder is contained in a sequential layer and performs the function of compressing the input at each of the linear layers into an encoded representation. Once done, the output is transmitted to the decoder part.

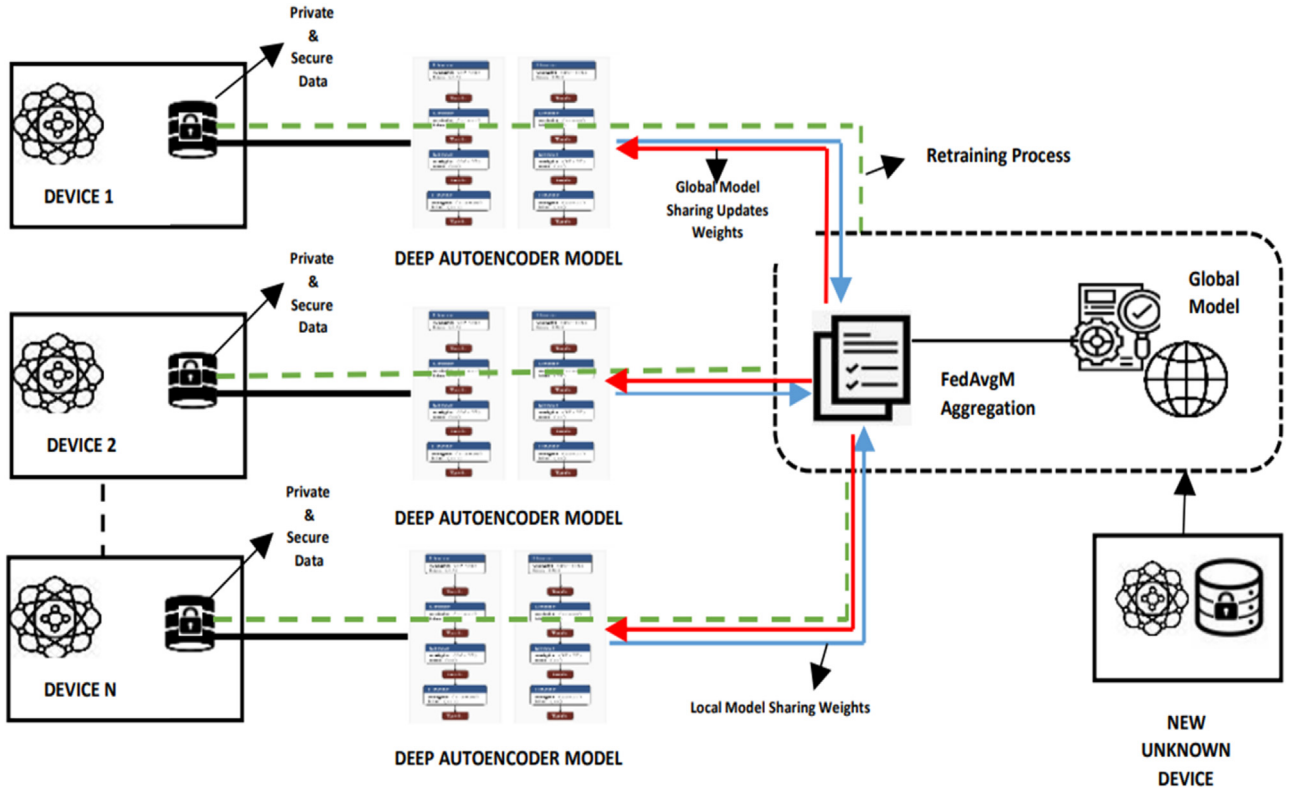

Fig. 3. Proposed Deep AE Model.

The decoder, like the encoder, is also contained in a sequential layer. It begins its job by receiving only 25 % of the input features in the first linear layer, together with the hyperbolic tangent function. The output of this layer is transmitted to the second linear layer, which receives only 33 % of the input features and a hyperbolic tangent function. The third linear layer receives only 50 % of its input features together with the hyperbolic tangent function. The fourth and last linear layer receives only 75 % of the input features together with the hyperbolic tangent function. The encoder thereafter decompresses what the model has learnt and reconstructs the data to its original form.

The DNN architecture proposed in this study uses a linear layer to perform a transformation on the received input data. Fig. 4 shows the DNN architecture built for this study. Here, the architecture consist of four linear layers, similar to that of the AE architecture above. The first linear layer receives the input features of 115 with 256 hidden units of the NN. A linear transformation is performed on the data using the Rectified Linear Unit (ReLU) activation function. The ReLU activation function is a non-linear transformation for negative inputs, but it is a linear transformation for positive inputs. Then a dropout layer of 0.2 is added to regularize the algorithm and prevent overfitting. The output of this layer is transmitted to another linear layer with 128 hidden units in the presence of another ReLU activation function. A linear transformation is again performed on the data and passed through a dropout layer of 0.2. The output is transmitted to the third linear layer with 64 hidden units and a ReLU activation function. Another linear transformation is performed on the data in the presence of the ReLU activation function and then passed through a dropout layer of 0.2. To obtain the output and make predictions of the normal and abnormal classes, it is passed through a linear layer.

Generally, the unsupervised AE model and the supervised DNN model have similar linear layers. The only difference is that the DNN model has only 4 linear layers while the AE model has 4 linear layers both in the encoder and the decoder parts. Another major difference is that the AE model learns unsupervised while the DNN model learns via

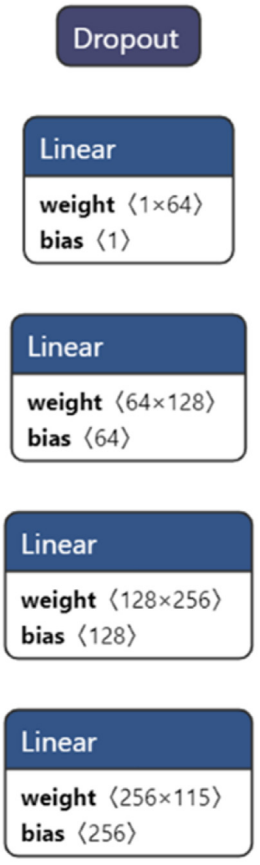

Fig. 4. The DNN architecture.

Table 1
Performance metrics.

Performance Metric	Definition
Accuracy	$\frac{TP+TN}{(TP+TN+FP+FN)}$
Precision	$\frac{TP}{(TP+FP)}$
Recall	$\frac{TP}{(TP+FN)}$
F1-Score	$\frac{2\ x\ Precision\ x\ Recall}{(Precision+Recall)}$

supervision. This is part of what the study investigates, to see how both models are well suited for FL-based IDS in IoT device applications.

4.2. Federated learning model

In this study, the Federated Averaging Algorithm with Momentum (FedAvgM) was selected as inspired by [21]. The researchers reported better performance of the algorithm compared to the Federated Averaging Algorithm (FedAvg). The latter's optimization process is similar to that of the Stochastic Gradient Descent (SGD) algorithm. The former leverages the SGD optimization process with momentum vector addition while updating its weights. This is expected to enhance its accuracy in performance. In operation, the FedAvgM aggregates the model's weights received from every client and update the global model with updated weights.

4.3. Train and test

Before training begins, the client models are synchronized with the global weights. The optimal parameters obtained from the hyperparameter optimization performed are used to initialize the model. The federated modelling begins with the training of the client model and the global model. The clients are updated, and the models are retrained on the global server. All client models are aggregated and saved. To test and evaluate the model, the Accuracy, precision Recall, F1-score, TPR, and FPR metrics are defined. The trained global model is loaded and used to test on the client devices. All nine client IoT devices are tested in this study. This is done by first computing the threshold and then testing the devices. The threshold computation is performed by adding the sum of the sample's Mean Squared Error (MSE) mean and the sample's MSE Standard Deviation (SD) over the normal training dataset.

4.4. Performance metrics

The machine learning models will be evaluated using standard performance metrics: accuracy, precision, recall, and F1-score (see Table 1) [22]. Where true positive (TP) means anomalous traffic correctly identified, true negative (TN) means normal traffic correctly identified, false positive (FP) means normal traffic incorrectly identified as anomalous, and false negative (FN) means anomalous traffic incorrectly identified as normal.

5. Results and discussion

In this section, we presents the results of the unsupervised and supervised DL models for both FL and non-FL predictions for all nine IoT devices. It starts by presenting the optimal parameters used to build each DL. These parameters are obtained from the random search hyperparameter optimization performed using the Neural Net classifier as the estimator. Four optimal parameters which include the initial learning rate, batch size, maximum training epoch, and the optimizer algorithm are tuned. Thereafter, the optimal parameters are used to build the DL supervised and unsupervised models for both FL and non-FL predictions on the nine IoT devices dataset. Results obtained are presented in the following sections and the findings are discussed.

Table 2
Optimal Parameters Used to Build the AE DL model.

IoT Device	Initial Learning rate	Batch size	Max. Training Epoch	Optimizer Algorithm
1	0.01	256	40	SGD
2	0.01	64	20	Adam
3	0.001	64	40	SGD
4	0.0001	256	40	Adam
5	0.0001	128	20	SGD
6	0.001	128	10	Adam
7	0.0001	256	40	SGD
8	0.001	64	40	SGD
9	0.001	64	10	Adam

Table 3
Optimal Parameters Used to Build the DNN DL model.

IoT Device	Initial Learning rate	Batch size	Max. Training Epoch	Optimizer Algorithm
1	0.001	128	40	Adam
2	0.01	128	40	SGD
3	0.01	128	10	SGD
4	0.01	64	20	SGD
5	0.001	128	40	SGD
6	0.001	128	20	SGD
7	0.0001	128	40	SGD
8	0.0001	128	20	SGD
9	0.0001	64	10	Adam

From Tables 2 and 3, the results of the random search optimization were performed to obtain both the unsupervised and supervised DL model optimal parameters. For the unsupervised DL AE model, it is observed that different parameter combinations are needed to build the most accurate AE model using each of the nine IoT device data. Generally, this is also true for the DNN model. However, from the DNN model optimal parameters in Table 3, it is inferred that this architecture has a preference for a batch size of 128 and an SGD optimizer algorithm for most of the nine IoT devices data used to train it. Specifically, taking the IoT device-1 as an example, an initial learning rate of 0.01, batch size of 256, training epochs of 40, and an SGD optimizer algorithm are needed to accurately build the unsupervised AE model for anomaly detection in this device. On the other hand, for the same device, an initial learning rate of 0.01, a batch size of 128, training epochs of 40, and an Adam optimizer algorithm are needed to accurately build a DNN model for anomaly detection in this device. However, these parameter combinations would not have been easily guessed for such a study. Hence, the importance of first performing hyperparameter optimization before building the models.Fig. 5 shows the Average performance results of the comparison between FL and non-FL AE model predictions across all metrics.

Fig. 6 shows the Average performance results of the comparison between FL and non-FL DNN model predictions across all metrics.

Fig. 7 shows the Average performance of both the unsupervised and supervised models in FL and non-FL predictions based on all metrics.

Results from these figures suggest a similar performance pattern for all metrics except for some insignificant differences. For example, the performance pattern in terms of accuracy, precision, F1-Score, TPR, and the FPR metrics are very similar to that of the FL training scheme except for very negligible differences. But for the Recall metrics, the performance pattern is the same as the performance pattern of all devices and device training sets in the FL scheme. For the unsupervised AE model, Fig. 5 shows the average performance comparison of the model trained via the FL training scheme and non-FL training scheme. The comparison between FL and non-FL training schemes indicates that both perform well in terms of all metrics, except for the FPR metric. The figure shows that the FL training scheme is the best option for training the unsupervised AE model for anomaly detection applications in IoT devices based on the FPR metric. However, Fig. 6 shows the DNN model's performance

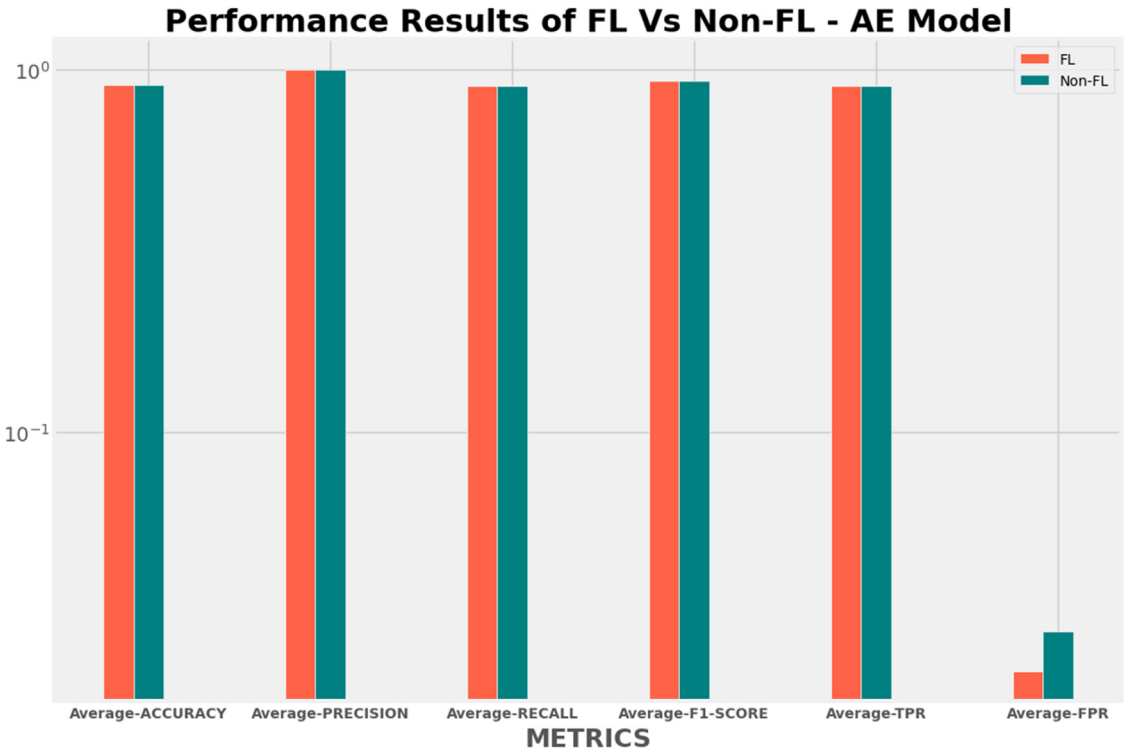

Fig. 5. Average performance results of FL vs. Non-FL by the AE model.

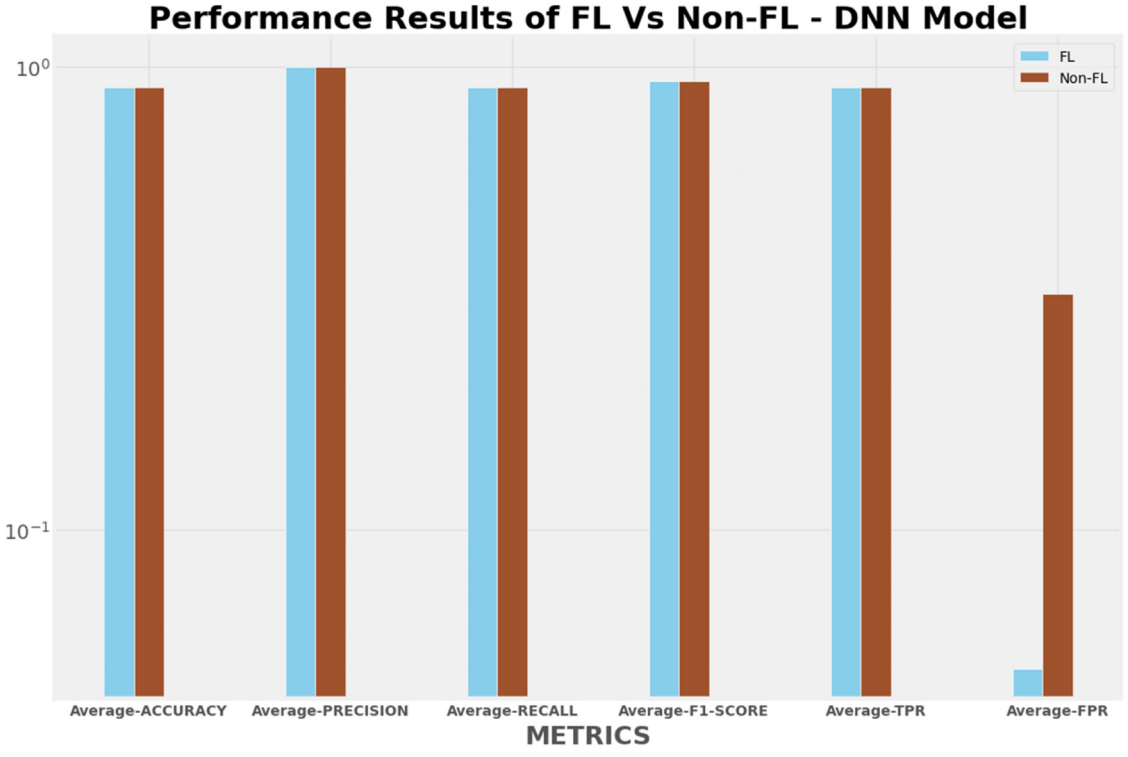

Fig. 6. Average performance results of FL vs. Non-FL by the AE model.

in terms of comparing the FL and the non-FL training scheme. Here, it is observed that both schemes compete at par based on all metrics except the FPR metric. However, it is showed, based on this metric, that training the DNN model via the FL training scheme is best.

In addition, to have a broader view of how all models performed under all training schemes, Fig. 7 shows some details. Here, we see that both the unsupervised and the supervised DL models compete at par in terms of all metrics, except in terms of the FPR. Hence, using this metric, it is evident that the unsupervised AE DL trained via the FL training scheme is best. The AE trained via the non-FL training scheme is second and even better than the DNN trained via the FL scheme. These show the capability of

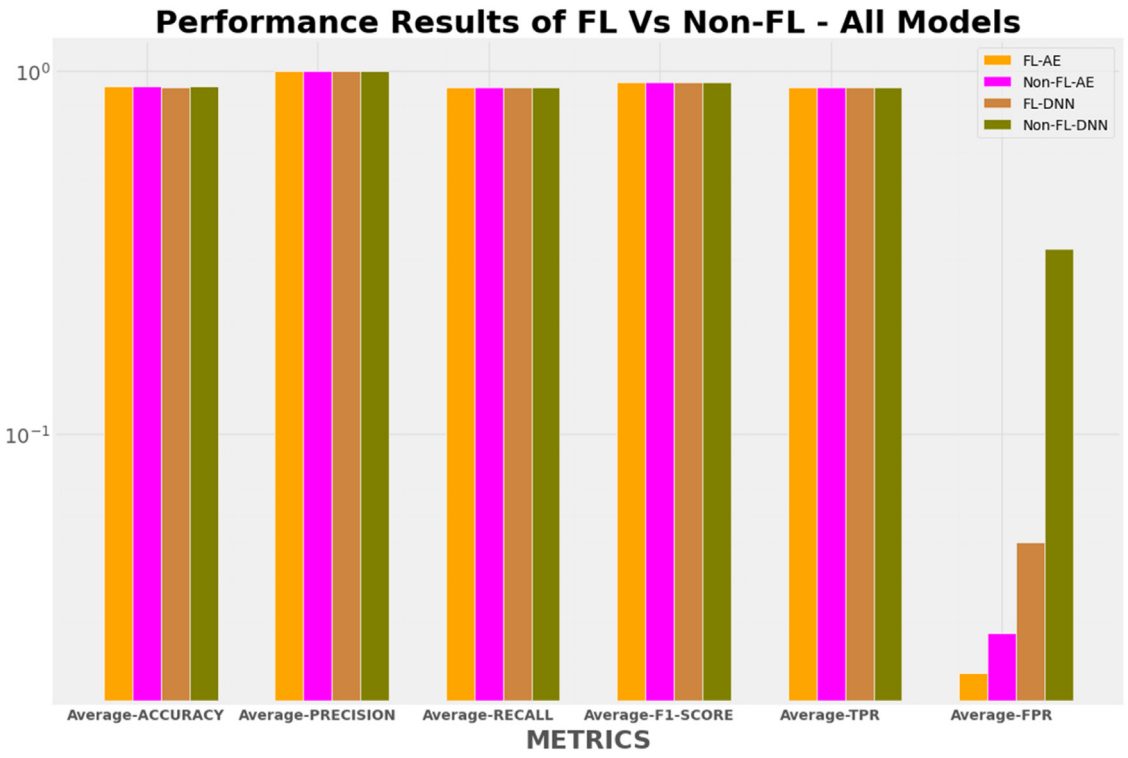

Fig. 7. Average performance results of FL vs. Non-FL by the all models.

the unsupervised AE model over the supervised DNN model for anomaly detection applications in IoT devices. One possible reason why FL can improve FPR metric in anomaly detection is that it can learn from diverse and heterogeneous data sources without compromising their quality or security [23]. This can help the model capture the variability and complexity of normal and abnormal data patterns more accurately and robustly. Anomaly patterns can vary across different devices or servers due to unique characteristics of each local environment. FL allows models to be trained on local data, enabling them to capture these localized anomaly patterns. By incorporating device-specific knowledge, the model can better distinguish between normal and anomalous instances, potentially leading to fewer false positives. In dynamic environments, new types of anomalies may emerge over time. FL's ability to continually learn from local data means that models can adapt to these changes more effectively. This adaptability can result in better anomaly detection and a lower FPR, as the model becomes more accurate over time.

6. Discussion

The comparison between FL-trained and non-FL models provides valuable insights into the efficacy of FL in the context of IDS for IoT. Specifically, the results indicate that FL-trained models not only achieve comparable or better performance than non-FL models but also offer the advantage of preserving data privacy by training models locally on individual IoT devices. This highlights the potential of FL as a decentralized approach to enhancing the security and privacy of IoT networks while maintaining high detection accuracy.

The findings of this study have broader implications for the field of IoT security, suggesting that the proposed approach of using FL with DL models can effectively address security and privacy challenges in IoT environments. By leveraging FL, IDS for IoT devices can achieve robust threat detection capabilities while mitigating concerns related to centralized data storage and processing. Furthermore, the proposed approach can be adapted to other network environments beyond IoT, such as industrial control systems or smart city infrastructures, where similar security and privacy concerns exist.

Practical considerations for implementing the proposed Intrusion Detection System (IDS) approach in real-world IoT environments involve several key aspects. Firstly, ensuring compatibility and interoperability with existing IoT devices and network infrastructure is essential for seamless integration. Secondly, addressing resource constraints on IoT devices, such as limited computational power and memory, requires optimization of DL models and FL algorithms for efficiency. Additionally, establishing robust communication protocols and security mechanisms for data transmission between devices and the centralized coordinator is crucial to safeguarding data privacy and integrity. Finally, ongoing monitoring and maintenance are necessary to adapt the IDS to evolving threats and changes in the IoT environment over time.

In addition, to have a broader view of how all models performed under all training schemes,

Fig. 7 shows some details. Here, we see that both the unsupervised and the supervised DL models compete at par in terms of all metrics, except in terms of the FPR. Hence, using this metric, it is evident that the unsupervised AE DL trained via the FL training scheme is best. The AE trained via the non-FL training scheme is second and even better than the DNN trained via the FL scheme. These show the capability of the unsupervised AE model over the supervised DNN model for anomaly detection applications in IoT devices. One possible reason why FL can improve FPR metric in anomaly detection is that it can learn from diverse and heterogeneous data sources without compromising their quality or security [23]. This can help the model capture the variability and complexity of normal and abnormal data patterns more accurately and robustly. Anomaly patterns can vary across different devices or servers due to unique characteristics of each local environment. FL allows models to be trained on local data, enabling them to capture these localized anomaly patterns. By incorporating device-specific knowledge, the model can better distinguish between normal and anomalous instances, potentially leading to fewer false positives. In dynamic environments, new types of anomalies may emerge over time. FL's ability to continually learn from local data means that models can adapt to these changes more effectively. This adaptability can result in better anomaly detection and a lower FPR, as the model becomes more accurate over time.

Table 4
Performance comparison of the proposed FL- DNN methods.

Performance Metrics	Auto-Encoder [6]	CNN [7]	RNN [7]	LSTM [7]	FL [8]	LSTM-RNN [11]	CNN-LSTM [12]	RCNN (Om [14])	Proposed FL-DNN
Accuracy	–	–	–	–	93.7 %	89.47 %	90.88 %	98.20 %	90.39 %
Precision	99.30 %	–	–	–	88.2 %	88.99 %	93.04 %	92.27 %	99.99 %
Recall	99.99 %	–	–	–	–	82.87 %	91.91 %	96.64 %	90.10 %
F1-Score	–	91 %	41 %	62 %	–	78.19 %	88 %	94.40 %	93.12 %

Table 5
Accuracy comparison of the proposed FL- DNN methods with other ML models.

Performance Metrics	Logistic Regression	Random Forest	SVM	Naïve Bayes	Proposed FL-DNN
Accuracy	82.56 %	99.05 %	82.45 %	60.48 %	90.39 %

The performance comparison of our proposed FL-DNN with other studies that used N-BaIoT dataset is shown in Table 4. The performance comparison with reference ML models is shown in Table 5.

The limitations of the study include scalability issues related to FL, particularly in large-scale IoT deployments with a large number of devices. Additionally, the adaptability of the proposed approach to different types of IoT devices and network configurations may pose challenges and require further investigation. Furthermore, the practical deployment of the FL framework in real-world IoT environments may encounter obstacles such as network connectivity constraints or device heterogeneity, which need to be addressed for successful implementation.

Future research directions could explore the integration of other DL models or advanced FL techniques to further enhance the effectiveness of IDS for IoT security. For example, incorporating recurrent neural networks (RNNs) or graph neural networks (GNNs) may improve the detection of temporal or structural patterns in IoT network traffic. The use of ensemble methods for improved performance. Additionally, enhancing the FL framework with techniques such as differential privacy or secure aggregation could strengthen the privacy guarantees of FL-trained models in distributed IoT environments.

7. Conclusion

In this study, the FL-based IDS for IoT using unsupervised and supervised DL models is evaluated. The proposed method compares the performance of unsupervised and supervised DL models in securing the IoT network. To enhance security and also handle the data privacy challenges of the previous methods of developing the anomaly detection system, the FL training scheme is proposed, which is further compared to the non-FL training scheme. To achieve this, one unsupervised AE model and one supervised DNN model are built and trained using the publicly available and very recent IoT dataset for nine devices. To further make the model robust, and contrary to previous studies, a hyperparameter tuning of the DL model parameters is performed before building the models. To further make the model more robust, the FL algorithm called the FedAvgM is adopted in this study. Both FL and non-FL training is performed using all nine IoT devices and their data. The results of the study highlight the performance of the unsupervised AutoEncoder (AE) model and other DL models trained both within the Federated Learning (FL) framework and using non-FL methods. Specifically, the unsupervised AE model demonstrated superior performance across various evaluation metrics when tested on all nine IoT devices. Furthermore, the comparison between FL-trained and non-FL models revealed the effectiveness of FL in enhancing the performance and privacy of IDS for IoT devices.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Babatunde Olanrewaju-George: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation. **Bernardi Pranggono:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Funding

The work was supported in part by Sheffield Hallam University.

Data Availability Statement

The N-BaIoT dataset is available at http://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT.

References

[1] N.N. Alajlan, D.M. Ibrahim, TinyML: enabling of inference deep learning models on ultra-low-power IoT edge devices for AI applications, Micromachines. (Basel)Micromachines. (Basel) 13 (2022) 851, doi:10.3390/mi13060851.
[2] S.A. Rahman, H. Tout, C. Talhi, A. Mourad, Internet of things intrusion detection: centralized, on-device, or federated learning? IEEE Netw. 34 (2020) 310–317, doi:10.1109/MNET.011.2000286.
[3] C.L. Stergiou, K.E. Psannis, B.B. Gupta, InFeMo: flexible big data management through a federated cloud system, ACM Trans. Internet Technol. 22 (46) (2021) 1–46 22, doi:10.1145/3426972.
[4] K. Yadav, B.B. Gupta, C.-H. Hsu, K.T. Chui, Unsupervised federated learning based IoT intrusion detection, in: 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE). Presented at the 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), 2021, pp. 298–301, doi:10.1109/GCCE53005.2021.9621784.
[5] Muhammad Almas Khan, Muazzam A. Khan, S.U. Jan, J. Ahmad, S.S. Jamal, A.A. Shah, N. Pitropakis, W.J. Buchanan, A deep learning-based intrusion detection system for MQTT enabled IoT, Sensors 21 (2021) 7016, doi:10.3390/s21217016.
[6] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-BaIoT—network-based detection of iot botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (2018) 12–22, doi:10.1109/MPRV.2018.03367731.
[7] J. Kim, M. Shim, S. Hong, Y. Shin, E. Choi, Intelligent detection of iot botnets using machine learning and deep learning, Appl. Sci. 10 (2020) 7009, doi:10.3390/app10197009.
[8] T. Zhang, C. He, T. Ma, L. Gao, M. Ma, S. Avestimehr, Federated learning for internet of things, in: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, SenSys '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 413–419, doi:10.1145/3485730.3493444.
[9] D.C. Attota, V. Mothukuri, R.M. Parizi, S. Pouriyeh, An ensemble multi-view federated learning intrusion detection for IoT, IEEE Access. 9 (2021) 117734–117745, doi:10.1109/ACCESS.2021.3107337.
[10] O. Shahid, V. Mothukuri, S. Pouriyeh, R.M. Parizi, H. Shahriar, Detecting network attacks using federated learning for IoT devices, in: 2021 IEEE 29th International Conference on Network Protocols (ICNP). Presented at the 2021 IEEE 29th International Conference on Network Protocols (ICNP), 2021, pp. 1–6, doi:10.1109/ICNP52444.2021.9651915.
[11] A.A. Hezam, S.A. Mostafa, A.A. Ramli, H. Mahdin, B.A. Khalaf, Deep learning approach for detecting botnet attacks in IoT environment of multiple and heterogeneous sensors, in: N. Abdullah, S. Manickam, M. Anbar (Eds.), Advances in Cyber Security, Springer, Singapore, 2021, pp. 317–328, doi:10.1007/978-981-16-8059-5_19.

[12] H. Alkahtani, T.H.H. Aldhyani, Botnet attack detection by using CNN-LSTM model for internet of things applications, Security Commun. Networks 2021 (2021) e3806459, doi:10.1155/2021/3806459.
[13] E.M. Campos, P.F. Saura, A. González-Vidal, J.L. Hernández-Ramos, J.B. Bernabé, G. Baldini, A. Skarmeta, Evaluating federated learning for intrusion detection in internet of things: review and challenges, Comput. Networks 203 (2022) 108661, doi:10.1016/j.comnet.2021.108661.
[14] C.U. Om Kumar, S. Marappan, B. Murugeshan, P.M.R. Beaulah, Intrusion detection model for iot using recurrent kernel convolutional neural network, Wireless Pers Commun. 129 (2023) 783–812, doi:10.1007/s11277-022-10155-9.
[15] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.
[16] Y. Fan, Y. Li, M. Zhan, H. Cui, Y. Zhang, IoT defender: a federated transfer learning intrusion detection framework for 5G IoT, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). Presented at the 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), 2020, pp. 88–95, doi:10.1109/BigDataSE50710.2020.00020.
[17] V. Mothukuri, P. Khare, R.M. Parizi, S. Pouriyeh, A. Dehghantanha, G. Srivastava, Federated-learning-based anomaly detection for iot security attacks, IEEE Internet Things J. 9 (2022) 2545–2554, doi:10.1109/JIOT.2021.3077803.
[18] V. Rey, P.M. Sánchez Sánchez, A. Huertas Celdrán, G. Bovet, Federated learning for malware detection in IoT devices, Comput. Netw. 204 (2022) 108693, doi:10.1016/j.comnet.2021.108693.
[19] P. Madan, V. Singh, D.P. Singh, M. Diwakar, B. Pant, A. Kishor, A hybrid deep learning approach for ECG-based arrhythmia classification, Bioengineering 9 (2022) 152, doi:10.3390/bioengineering9040152.
[20] M.M. Forootan, I. Larki, R. Zahedi, A. Ahmadi, Machine learning and deep learning in energy systems: a review, Sustainability. 14 (2022) 4832, doi:10.3390/su14084832.
[21] Hsu, T.-M.H., Qi, H., Brown, M., 2019. Measuring the effects of non-identical data distribution for federated visual classification. https://doi.org/10.48550/arXiv.1909.06335.
[22] P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutorials 21 (2019) 686–728, doi:10.1109/COMST.2018.2847722.
[23] F. Cavallin, R. Mayer, Anomaly detection from distributed data sources via federated learning, in: L. Barolli, F. Hussain, T. Enokido (Eds.), Advanced Information Networking and Applications, Lecture Notes in Networks and Systems, Springer International Publishing, Cham, 2022, pp. 317–328, doi:10.1007/978-3-030-99587-4_27.