Federated Learning-Based Intrusion Detection System for IoT Environments with Locally Adapted Model

Souradip Roy
Department of Computer Science
North Dakota State University
Fargo, USA
souradip.roy@ndsu.edu

Juan Li
Department of Computer Science
North Dakota State University
Fargo, USA
j.li@ndsu.edu

Yan Bai
School of Engineering and Technology
University of Washington Tacoma
Tacoma, USA
yanb@uw.edu

***Abstract*— As the Internet of Things (IoT) becomes more prevalent, the need for intrusion detection systems (IDS) to protect against cyberattacks increases. However, the limited computing capabilities of IoT devices often require sending data to a centralized cloud for analysis, which can cause energy consumption, privacy issues, and data leakage. To address these problems, we propose a Federated Learning-based IDS that distributes learning to local devices without sending data to a centralized cloud. We also create lightweight local learners to accommodate IoT device limitations and locally adapted models to handle non-independent intrusion data distribution. We evaluate our method using NBaIoT and CICIDS-2017 datasets, and our results demonstrate comparable performance to centralized learning on metrics including accuracy, precision, and recall, while addressing privacy and data leakage concerns.**

Keywords—Internet of Things (IoT), Intrusion Detection Systems (IDS), federated learning, deep learning, cybersecurity

I. INTRODUCTION

The pervasiveness of the Internet of Things (IoT) has witnessed a rapid rise in recent years, with its applications encompassing diverse domains ranging from smart homes, healthcare, and transportation systems. As the popularity of IoT increases, it becomes a breeding ground for intrusions that can compromise the security and privacy of IoT devices and networks. A notable example of intrusion is Distributed Denial of Service (DDoS) attacks, which involves flooding the IoT network with traffic from multiple sources to overwhelm and shut down the network. Malicious software programs are also prevalent in IoT systems, and they can be introduced through infected devices or software updates. In addition, unauthorized devices can connect to the IoT network and pose security threats, while hackers may intercept communication between two devices and manipulate the data being transmitted. The exponential growth of IoT has thus spurred an augmented demand for intrusion detection systems (IDS) that can protect against sophisticated cyberattacks, ensuring the security and privacy of IoT devices and networks.

Deep learning has shown great promise in IDS due to its ability to learn complex patterns and identify abnormal behavior. However, the limited computing capabilities of IoT devices have made it difficult to implement deep learning-based IDS on these devices. One common approach to addressing this challenge is to use centralized IDS methodologies that require sending live data from IoT devices to distant data centers for analysis. However, this approach poses several problems. First, sending data to a centralized cloud requires significant energy consumption, which is not ideal for IoT devices that often operate on battery power. Second, data sharing over the internet poses privacy issues, and data leakage can occur during transmission. Third, because IoT devices generate data with varying statistical distributions (non-IID), a single IDS may not be the best solution for this problem. Moreover, the limited processing capabilities of IoT devices must be taken into account while designing an IDS.

To overcome these challenges, we propose a Federated Learning-based IDS approach that enables the distribution of learning to local devices without sending data to a centralized cloud. In federated learning, multiple devices collaborate in training a model while preserving data privacy by keeping the data local. Each local device, or client, has an IDS model that can be trained using its own data. The local model is trained for a specified number of epochs and then communicates the model parameters to a master model for aggregation. The shared parameters are aggregated using techniques such as FedAvg [1], and the modified parameters are delivered back to the local clients.

Federated learning protects data privacy while also assisting clients with limited data production in achieving their desired goals. However, the statistical heterogeneity of the data produced by different devices makes it challenging to develop a generic IDS system that can be applied to all devices. To address this issue, we propose a Locally Adapted model of federated learning (LAFL) for IoT Intrusion Detection. This approach tailors detection models for specific IoT devices and IoT traffic, improving accuracy and efficiency while maintaining data privacy. LAFL captures the unique properties of IoT devices and traffic, resulting in enhanced accuracy and efficiency. The shared parameters are exchanged for aggregation, while the parameters of the locally adapted architecture are not shared. The locally adapted

This work was supported by the National Science Foundation (NSF) with award numbers: 1722913, 2218046 and 1921576.

models are beneficial for IoT devices with varying data distributions or security constraints. They can adapt to changes in data distribution or device activity, resulting in more accurate and up-to-date intrusion detection.

The rest of the paper is organized as follows: Section 2 provides an overview of related work in the field of IDS for IoT. Section 3 describes the proposed methodology in detail. Section 4 presents the experimental setup and results. Finally, we conclude the paper in Section 5 and provide directions for future work.

II. Related Work

IDS solutions for IoT networks can be classified into three approaches: signature-based, anomaly-based, and hybrid. Signature-based approach (e.g., [2]) is effective for known attacks, but it requires continuous human interventions and knowledge expertise to extract attack patterns and signatures to update the IDS model, which makes it inefficient and ineffective for IoT due to the heterogeneity, dynamicity, and complex nature of the network [3]. On the other hand, anomaly-based IDS detection (e.g., [4]) is effective for unknown attacks and requires fewer human interventions, making it advantageous for IoT [5]. The hybrid approach (e.g., [6]) combines both signature-based and anomaly-based approaches, but the reliance on pre-defined attack patterns makes it impractical for intrusion detection in IoT networks [7-9]. Therefore, anomaly-based intrusion detection systems are crucial for detecting intrusions in IoT environments.

Over the past few years, a number of anomaly-based IDSs using deep learning have been developed for IoT networks[10] [11] [12]. These systems aim to identify abnormal traffic patterns in the network. Some examples of these systems include the artificial neural network-based IDS proposed by Adam et al. [7], and the traffic classification system developed by Dias et al. [8]. Recent research has shown that deep learning-based models, such as Convolutional Neural Networks (CNN) [9], [13], variational autoencoders [14], and Recurrent Neural Networks (RNN) [11], and advanced Long Short-Term Memory (LSTM) [15] are more effective than traditional machine learning techniques, such as Decision Trees and Support Vector Machines.

For example, CNN system [16], [17] can automatically detect and learn key IoT intrusion features without requiring manual intervention. With their dense network architecture, CNNs are capable of both identification and prediction. However, the vast number of parameters in CNNs can make them computationally expensive and inefficient, particularly in the context of IoT devices. To address this, researchers have explored combining CNNs with techniques such as Autoencoders (AEs) to reduce the dimensionality of high-dimensional IoT data [18].

Compared to traditional machine learning architectures, federated learning employs a decentralized approach to train models and update them incrementally. Federated learning has been applied to cybersecurity and intrusion detection in IoT environments. Nguyen et al. [19] proposed a self-learning anomaly detection system based on LSTM and Gated Recurrent Units (GRU) to identify hacked IoT devices on the network. The system achieved 98.2% accuracy with a very low false alarm rate of 95.6%. In another study, Zhao et al. [20] utilized LSTM to develop an Intelligent Intrusion Detection system that achieved 99.21% accuracy and an F1 score of 99.21. Liu et al. [21] proposed using federated learning with CNN-LSTM models to analyze time series data collected from distributed edge devices in Industrial IoT settings. The proposed model aims to extract relevant information from the time series data. These systems that utilized LSTM techniques excel in processing and predicting time series data over extended periods. LSTM models possess a unique memory capability that can retain information from previous time steps, facilitating the learning process. Additionally, they are capable of handling noisy and continuous data representations. However, LSTMs are prone to overfitting, making it challenging to apply the dropout algorithm to mitigate this issue.

Two main types of personalized federated learning are model regularization and model interpolation. Mansour et al. [22] proposed a clustering approach to group clients with similar characteristics and train a distinct model for each cluster. In other studies, such as Bui et al. [23], federated learning is employed to train a set of local parameters based on each client and a set of global parameters, with user-representation suggested to achieve this objective. In contrast, Arivazhagan et al. [24] describe FedPer as a personalization approach consisting of base and personalization layers. The basic layers are trained using a standard federated learning technique and the FedAvg algorithm, while the personalized layers gather data from each IoT device. Thus, the FedPer algorithm accounts for the fact that the federated learning method produces statistically different results. Similarly, Liang et al. [25] suggest storing some neural network layers in the client and training the remaining model using Federated Learning. Chan et al. [26] proposed a different approach where they initially train a global model using traditional federated learning and then customize it with data from each device.

III. Methodology

The proposed LAFL model consists of a global model and multiple local models. The global model is responsible for training and updating the model parameters based on the information received from the local models. Each local model is created by the IoT devices that are responsible for collecting and processing data in their respective regions. The local models perform training on the data collected from IoT devices and send the updated model parameters to the global model for further optimization.

A. Global Model

The global model is a hybrid combination of a stacked autoencoder and a feedforward neural network (FNN). The bottleneck of the autoencoder is connected to the feed forward neural network for classification purpose. The stacked autoencoder is used for feature learning technique. A feature learning technique helps to understand the underlying features or patterns from the detection dataset. Feature learning reduces the number of features (dimensions) needed for learning, removes noisy and irrelevant features, and improves the detection rate of the system. There are two types of feature

learning techniques: supervised and unsupervised feature learning. The former learns on a labeled dataset and can be evaluated using the training and testing data, while the latter tries to make sense of unlabeled data by extracting features and patterns on its own. Therefore, we employ a stacked autoencoder, an unsupervised feature learning, as it does not need to provide labels for sample data. Using label data can introduce false positives as labelling of data requires huge expertise in the domain of cyber-security.

Secondly, the stacked autoencoder's feature learning technique helps to compress and reduce the feature space of the network traffic by removing multicollinearity that is observed in IoT traffic data. Multicollinearity is a state where multiple features in a dataset are highly correlated and contain similar information. Multicollinearity undermines the statistical significance of an independent variable and introduces noise in the data. To identify multicollinearity, we calculate the Variance Inflation Factor (VIF) corresponding to every independent variable in the dataset.

The stacked autoencoder is constructed using multiple sparse autoencoders consisting of one input layer, one hidden layer, and one output layer. The number of neurons in the output layer should be the same as that in the input layer. The hidden layers are connected in succession. A stacked autoencoder has an encoder and a decoder. The encoder is responsible for reducing the number of features by compressing them in each layer. Each hidden layer has an activation function that helps the neural network adjust its weights so that the desired output can be obtained. Our stacked autoencoder uses the tangent hyperbola function and the rectified linear unit respectively.

B. Local Model

The local models are created at the fog layer of the IoT network by first partitioning the data into multiple subsets based on the physical location of the IoT devices. Each subset is then used to train a separate neural network model. The neural network model for each subset is created with a base layers and local layer. The local models use the last two layers of the global model, as their base layers followed by a personalized layer and an output layer. Each local model has the same number of base layers and the same number of neurons in each of the base layers.

On the other hand, the local layers are unique to each model and are trained using only the data available in that particular model. These local layers are responsible for capturing the unique features and patterns present in the local data that may not be present in the global data. The use of local layers allows each model to adapt to the specific characteristics of the local data while still benefiting from the shared knowledge of the base layers. This approach helps in achieving better performance in IoT intrusion detection by combining the strengths of global and local models. Overall, the locally adapted layers in the proposed model allow for a fine-grained approach to feature extraction and pattern recognition that can improve the accuracy and robustness of the intrusion detection system in the IoT environment.

To ensure that the model stays up-to-date with the latest data, periodic model updates and re-federated learning are performed. In this process, local models send the parameters of their base layers to the global model. The global model aggregates these parameters to update itself and sends the updated parameters back to the local models, allowing them to update their own base layers accordingly. However, since different local models may have varying amounts of network traffic, their impact on the global model may differ. To address this, we assign weights to each local model based on its data volume. This ensures that local models with larger volumes of data have a greater influence on the global model. This weight is incorporated into the aggregation function, allowing for personalized updates to the global model based on the characteristics of each local model.

In the volume-driven aggregation function we used Volume Factor α_i to determine the impact of respective neural networks which is calculated based on the number of samples the neural network uses to train itself and the total number of samples present for all the models. We utilized the below equations to obtain the Volume Factor ai for each model during the aggregation process.

$$\alpha_i = \frac{d_i}{\sum_{i=1}^{K} d_i} \tag{1}$$

Eq. 1 is used to determine the Volume Factor denoted by α_i which is required during the aggregation method in the federated learning process. The i in α_i denotes the cluster number, d_i represents the number of data points in the i^{th} cluster. After determining the volume factor for each neural network, we multiply this value to the parameters of the respective neural network during the aggregation step. The volume-based aggregation function is expressed in Eq 2.

$$FedAvg_{Vol} = \frac{\sum_{i=1}^{K} \alpha_i P_i}{K}, P_i = (W_{lj}, b_j) \tag{2}$$

Eq. 2 calculates the updated parameters for the base-layers in each neural network when parameters are exchanged after certain number of training epochs. The W_{lj} is considered as the weight of the i^{th} neural network for l^{th} neuron in the j^{th} layer among all the base layers. Similarly, b_j denotes the bias for the same layer.

Furthermore, each model in the cluster is equipped with 3 hidden layers and 1 output layer, with 2 of the hidden layers being base layers and the remaining one being a personalized layer. The number of neurons in each layer is kept small to ensure that the models can be deployed on resource-constrained IoT devices without requiring significant computational resources.

The procedure of our proposed technique is described in Algorithm 1.

Algorithm 1: Locally Adapted Federated Learning (LAFL)

/* Assume that the global model has already been trained and now LAFL is being used for federated learning */

// Step 1: Initialize base-layers and local layers
Initialize the base-layers of n clients using Federated Transfer Learning (FTL) from the global model.
Initialize the parameters of local layers from the distribution of target domain

// Step 2: Exchange parameters for a fixed number of times
For exchange (e) = 1 to M
    // Step 3: Train local models using data points from each cluster
    For epoch (i) = 1 to N
        Train n_k model using data points from k^{th} cluster
    // Step 4: Calculate updated parameter for base-layers
    Calculate the updated parameters for base-layers using Eq 2
/*Step 5: Stop training if no improvement in efficiency for a fixed number of exchanges */
Stop training if there is no improvement in efficiency for q number of exchanges
End For
/* In Step 4, the weights for each local model are determined by the volume-driven aggregation mechanism, which takes into account the data volume of each cluster. */

IV. Evaluation

We conducted an evaluation of our technique by measuring various metrics, including Accuracy, F1-Score, Precision, and Recall. We used two publicly available datasets, namely CICIDS2017 [27] and NBaIoT [28].

A. Evaluations of CICIDS2017 Dataset

ICIDS2017 is a publicly available IDS dataset that comprises benign network flows as well as various types of attack flows that mimic real-world intrusion scenarios. This dataset is frequently used in recent cybersecurity studies for practical intrusion detection. CICIDS2017 [27] consists of network traffic analysis statistics obtained from CICFlowMeter, where each record contains 79 attributes, with 78 of them being related to network traffic and the last one indicating whether it is normal or a specific type of intrusion. The dataset includes samples from 14 distinct intrusion classes, which are classified under a single anomaly class. We preprocessed the data by removing records with missing values and applying Z-score normalization, as shown in Eq. 3.

$$Z = \frac{X_i - \overline{x}}{S_x} \qquad (3)$$

Here, X_i represents the value of an individual feature in a sample, $\overline{x}$ represents the mean value of the feature, and S_x represents the standard deviation. This preprocessing method is beneficial in handling extreme outliers while preserving the informational content of the samples.

Our experiments were conducted on a dataset of 36,000 samples. To train the initial global model, we randomly selected 500 samples. For the experimental models, we employed grid search techniques to determine the optimal number of hidden layers, neurons in each hidden layer, and activation functions. The test accuracy of the initial model is presented in Table I.

Table I: Performance of Initial Global Model for ICIDS2017 Data

Model Name	Precision	Recall	Accuracy
Initial Model	0.83	0.82	0.84

After training the initial model, we proceeded with Agglomerative Clustering on the sampled data to group similar network traffic patterns. The resulting dendrogram in Figure 1 helped us determine the optimal number of clusters. We chose 4 clusters for our experiments. Each local model is trained using data points from a specific cluster.

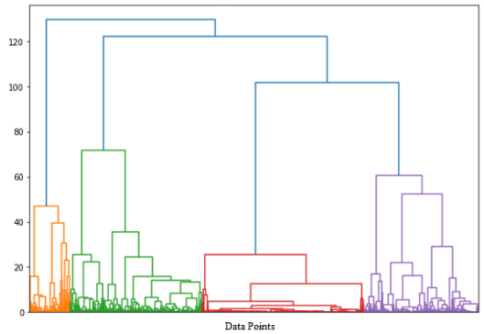

Fig. 1: Agglomerative Clustering of CICIDS2017 Dataset

We conducted a comparative analysis between our locally adapted model and a unified model without a personalized layer to evaluate the effectiveness of our approach. The unified model used the same base layers for all local models consisting of three identical layers. The results are presented in Fig. 2, where the x-axis represents the clusters used for training the local models and the y-axis represents the evaluation metrics, including accuracy, precision, and recall. The notation Ui represents the unified model trained using cluster i's data, while Pi denotes the personalized model using cluster i's data. As evident from Fig. 2, the locally adapted model outperformed the unified model, highlighting the significance of incorporating personalized layers to improve model performance. The personalized layers enable the local models to adapt to different traffic patterns, thereby enhancing their ability to accurately classify network traffic and detect anomalies. This finding underscores the importance of personalization in federated learning and underscores the potential advantages of incorporating locally adapted layers.

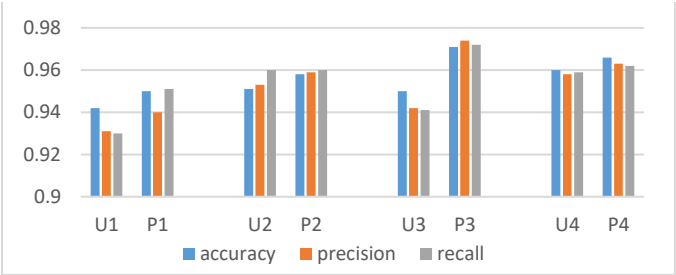

Fig. 2: Comparison of Performance between Personalized (P) and Non-Personalized (U) Models in Federated Learning.

We investigated the performance of our novel data-volume driven aggregation function, $FedAvg_{Vol}$ and compared it with the state-of-the-art *FedAvg* function. Again, the experiments were performed on four different clients. The results are presented in Fig. 3, As can be observed from Fig. 3, our proposed $FedAvg_{Vol}$ outperformed *FedAvg* function in most cases. This demonstrates the effectiveness of our data-volume

driven aggregation function in detecting anomalies in IoT networks in a Federated Learning setting. Our algorithm takes into consideration the data volume of each client during the aggregation process, which helps to mitigate the effect of clients with varying data volumes and improve the overall performance of the model. The results of this experiment provide evidence that our proposed approach can enhance the performance of Federated Learning in IoT networks and potentially other domains where clients have varying data volumes. By considering the data volume during the aggregation process, we can effectively leverage the data from all clients and improve the accuracy of the model without compromising privacy.

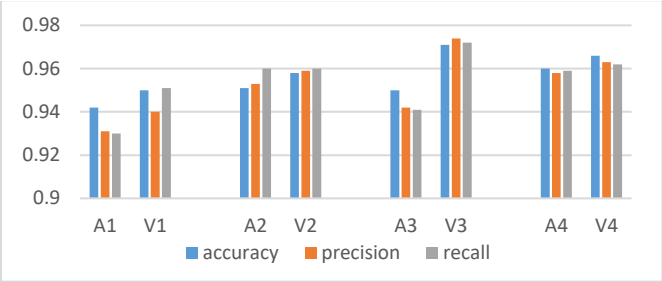

Fig. 3: Comparison of Performance between Average-based (A) and Volume-based (V) aggregation function in Federated Learning.

B. Evaluations of NBaIoT Dataset

We conducted an additional evaluation of our proposed method using the N-BaIoT dataset [28], a well-known dataset for network-based detection of IoT botnet attacks. The feature set of this dataset includes only the packet size of outbound packets, which is an aggregation of source IP, source MAC-IP, channel, and socket. The statistics provided for the packet size feature are mean and variance. In addition, the dataset includes other features such as packet count, packet jitter, and packet size of both inbound and outbound traffic. All of the statistics for these features were collected over five different time windows, including 100ms and 500ms. This evaluation allows us to assess the generalizability of our proposed method to different datasets and use cases beyond our initial experiments.

To develop an initial global model for the NBaIoT dataset, we conducted several experiments, similar to those performed on the CICIDS2017 dataset. The dataset contains a total of 115 features, but to reduce the complexity of the neural network models and make them suitable for IoT devices, we only used the Packet Size feature for both inbound and outbound traffic. We trained and tested the model to identify the best initial model. Initially, we used a time window of 100ms, and then we aggregated the packet size feature of 100ms and 500ms time windows, and so on, to determine the optimal time window for the model.

Table II shows that aggregating the 100ms and 500ms time windows significantly improved the efficiency of the initial model. However, further increasing the time windows did not result in a significant increase in efficiency. Thus, we chose to use only the inbound and outbound packet size features for 100ms and 500ms time windows in all of our experiments to strike a balance between accuracy and computational complexity. Additionally, we used a small number of samples (500) to train the initial model. Unlike the CICIDS2017 dataset, we did not use clustering techniques for the N-BaIoT dataset, as all data in the dataset belonged to the same IoT cameras category. Table III presents the performance of the initial model on three evaluation metrics: Precision, Recall, and Accuracy.

Table II. Efficiency Comparison of Different Time Window Aggregations for NBaIoT Dataset.

Time Window	**Accuracy (%)**
100ms	69
100ms + 500ms	82
100ms + 500ms + 1.5sec	84
100ms + 500ms + 1.5sec + 10sec	84.6
100ms + 500ms + 1.5sec + 10sec + 1min	85

Table III. Performance of Initial Global Model for NBaIoT Data

Model Name	**Precision**	**Recall**	**Accuracy**
Initial Model	0.83	0.82	0.84

After training the initial model, we utilized the LAFL Federated Learning approach to train personalized models for each device in the NBaIoT dataset. As we did with the CICIDS2017 dataset, we compared the performance of the personalized models with that of the unified models. The evaluation metrics, including precision, recall, and accuracy, are presented in Figures 4, 5, and 6, respectively. The results indicate that the personalized models consistently outperform the unified models in all of the aforementioned metrics. This finding emphasizes the importance of personalized models in enhancing the performance of Federated Learning models. It also highlights the potential benefits of tailoring the models to the specific characteristics of each device's data to accurately identify anomalies in IoT networks.

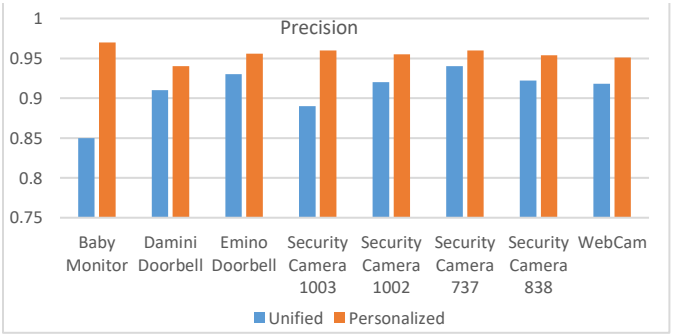

Fig. 4: Comparison of Precision between Personalized and Unified (Non-Personalized) Models in Federated Learning.

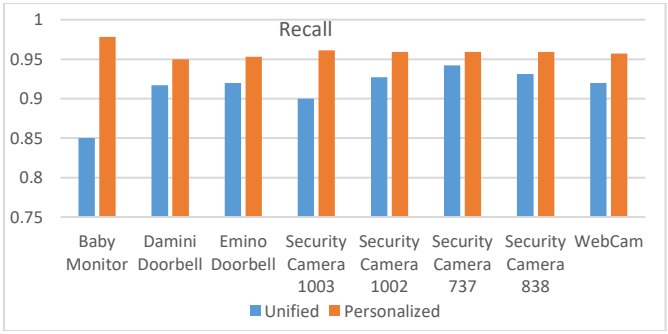

Fig. 5: Comparison of Recall between Personalized and Unified (Non-Personalized) Models in Federated Learning.

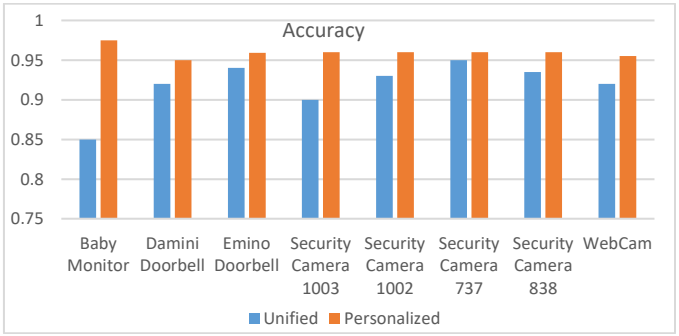

Fig. 6: Comparison of Accuracy between Personalized and Unified (Non-Personalized) Models in Federated Learning.

We conducted experiments on the NBaIoT dataset to evaluate the performance of $FedAvg_{Vol}$ and compared it with the *FedAvg* function. The results of these experiments, in terms of accuracy, precision, and recall, are presented in Fig. 7-9. Our proposed function outperformed the *FedAvg* function in most cases, demonstrating its effectiveness in detecting anomalies in the NBaIoT dataset.

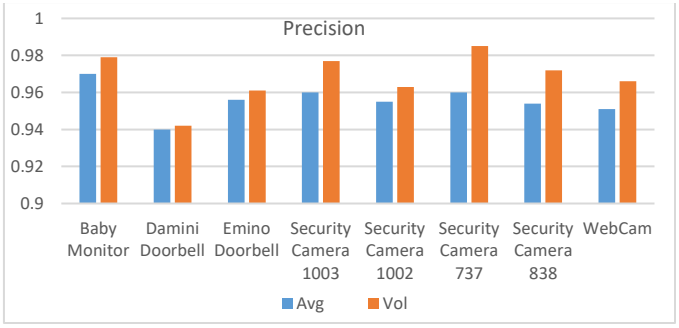

Fig. 7: Comparison of Precision between Average-based (A) and Volume-based (V) aggregation function in Federated Learning.

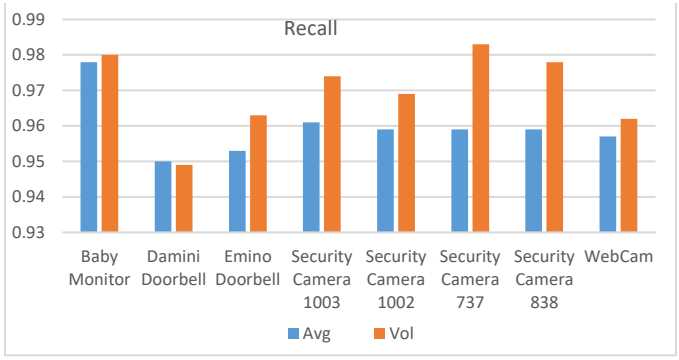

Fig. 8: Comparison of Recall between Average-based (A) and Volume-based (V) aggregation function in Federated Learning.

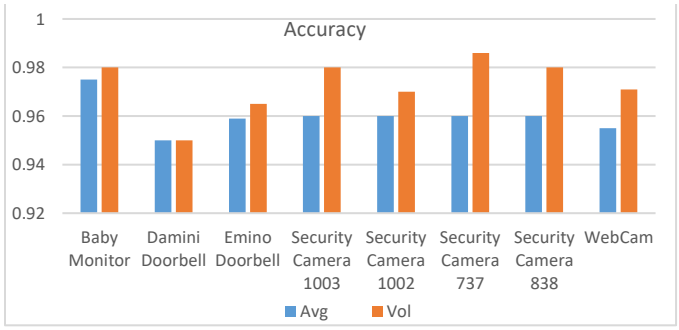

Fig. 9: Comparison of Accuracy between Average-based (A) and Volume-based (V) aggregation function in Federated Learning.

In conclusion, the experiments conducted on the CICIDS2017 and NBaIoT datasets demonstrate the effectiveness of personalized models and data-volume driven aggregation functions in improving the performance of federated learning models for anomaly detection in IoT networks. The personalized models were able to outperform the unified models, highlighting the importance of personalization in federated learning. Furthermore, the data-volume driven aggregation function, was shown to outperform the state-of-the-art FedAvg function in most cases, indicating its potential for improving the efficiency and accuracy of federated learning models. These findings provide valuable insights for developing more robust and efficient federated learning models for anomaly detection in IoT networks.

V. Conclusions

In conclusion, the proposed a Locally Adapted model of federated learning (LAFL) -based IDS method provides a promising solution to the challenges of intrusion detection in IoT networks. The experiments conducted on the NBaIoT and CICIDS-2017 datasets demonstrate the effectiveness of our approach in achieving comparable performance to centralized learning while addressing privacy and data leakage concerns. The personalized layer and data-volume driven aggregation function further enhance the performance of the method. Future work could involve exploring the use of more complex datasets and optimizing the proposed approach to further improve its performance. Additionally, investigation into the

impact of different hyperparameters on the method's performance could also be an area of future research.

REFERENCES

[1] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, "On the convergence of fedavg on non-iid data", 2019, arXiv preprint arXiv:1907.02189.

[2] N.U. Sheikh, H. Rahman, S. Vikram, H. AlQahtani, "A lightweight signature-based IDS for IoT environment", 2018, arXiv Preprint arXiv:1811.04582.

[3] W. Li , S. Tug , W. Meng and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments", Future Generation Computer Systems, vol 96, pp 481-489, July 2019.

[4] S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model", Journal of Computational Science, vol 25, pp 152-160, March 2018.

[5] P. K. Keserwani, M. C. Govil, E. S. Pilli & P. Govil, "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model", Journal of Reliable Intelligent Environments, vol 7, 2021, pp 3-21.

[6] C. A. de Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral and G. dos S. Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments", Computer Networks, vol 180, October 2020.

[7] N. Ádám, B. Madoš, A. Baláž, and T. Pavlik, "Artificial neural network based IDS," 2017, doi: 10.1109/SAMI.2017.7880294.

[8] L. P. Dias, J. J. F. Cerqueira, K. D. R. Assis, and R. C. Almeida, "Using artificial neural network in intrusion detection systems to computer networks," 2017, doi: 10.1109/CEEC.2017.8101615.

[9] Z. Chiba, N. Abghour, K. Moussaid, A. El omri, and M. Rida, "Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms," Comput. Secur., 2019, doi: 10.1016/j.cose.2019.06.013.

[10] A.H. Qureshi, H. Larijani, J. Ahmad and N. Mtetwa, "A novel random neural network based approach for intrusion detection systems". In 2018 10th Computer Science and Electronic Engineering (CEEC), 2018. pp. 50–55.

[11] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks", IEEE Access, 2017, http://dx.doi.org/10.1109/ACCESS.2017.2762418.

[12] A.A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things", Futur. Gener. Comput. Syst,2018, http://dx.doi.org/10.1016/j.future.2017.08.043.

[13] M. J. Kang and J. W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," PLoS One, 2016, doi: 10.1371/journal.pone.0155781.

[14] J. Kim, A. Sim, J. Kim, K. Wu, and J. Hahm, "Transfer Learning Approach for Botnet Detection Based on Recurrent Variational Autoencoder," 2020, doi: 10.1145/3391812.3396273.

[15] X. Wang and X. Lu, "A Host-Based Anomaly Detection Framework Using XGBoost and LSTM for IoT Devices", 2020, https://doi.org/10.1155/2020/8838571

[16] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao and L. Cui, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion", Measurement, vol 154, March 2020.

[17] A. El-Ghamry, A. Darwish, A. E. Hassanien, "An optimized CNN-based intrusion detection system for reducing risks in smart farming", Internet of Things, Vol 22, July 2023.

[18] A. Andalib and V. T. Vakili, "A Novel Dimension Reduction Scheme for Intrusion Detection Systems in IoT Environments", 2020, https://doi.org/10.48550/arXiv.2007.05922

[19] 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 Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jul. 2019, pp. 756–767.

[20] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, ''Intelligent intrusion detection based on federated learning aided long short-term memory,'' Phys. Commun., vol. 42, Oct. 2020, Art. no. 101157.

[21] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, 'Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach,'' IEEE Internet Things J., vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[22] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh. "Three approaches for personalization with applications to federated learning", arXiv preprint arXiv:2002.10619, 2020, https://doi.org/10.48550/arXiv.2002.10619

[23] D. Bui, K. Malik, J. Goetz, H. Liu, S. Moon, A. Kumar, and K. G. Shin, "Federated user representation learning", arXiv preprint arXiv:1909.12535, 2019, https://doi.org/10.48550/arXiv.1909.12535

[24] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. "Federated learning with personalization layers", arXiv preprint arXiv:1912.00818, 2019, https://doi.org/10.48550/arXiv.1912.00818

[25] P. P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, and L. P. Morency. "Think locally, act globally: Federated learning with local and global representations". arXiv preprint arXiv:2001.01523, 2020.

[26] Y. Chen, J. Wang, C. Yu, W. Gao, and X. Qin. "Fedhealth: A federated transfer learning framework for wearable healthcare", arXiv preprint arXiv:1907.09173, 2019.

[27] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.

[28] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici 'N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders', IEEE Pervasive Computing, Special Issue - Securing the IoT (July/Sep 2018).