Intrusion Detection Using Federated Learning in Non-IID Data Environments

DOI: 10.54654/isj.v3i23.1061

Ha Xuan Tung*, Tran Linh Trang, Vo Khuong Linh

***Abstract*— Intrusion Detection Systems (IDS) are essential for protecting networks by detecting threats. Machine learning (ML) based approaches have enhanced the effectiveness for IDS. However, it raises privacy concerns for IDS based on ML approaches due to the need for centralized data. To address this issue, Federated Learning (FL) has been applied to IDS. It allows ML models to be trained on decentralized devices/clients without sharing raw data. However, FL faces challenges with non-independent and identically distributed (non-IID) data, which reduces the performance of intrusion detection models. This paper introduces a loss function that jointly learns compact local representations on each client and a global model across all clients, enhancing the robustness of FL in non-IID data environments. Experimental results demonstrate that our method significantly improves the accuracy and robustness of FL systems for IDS in non-IID environments.**

***Tóm tắt*— Hệ thống phát hiện xâm nhập (IDS) đóng vai trò quan trọng trong việc bảo vệ hạ tầng mạng bằng cách phát hiện các mối đe dọa. Các phương pháp dựa trên học máy (ML) đã nâng cao hiệu quả của IDS. Tuy nhiên, điều này đã gây ra lo ngại về quyền riêng tư do các phương pháp IDS dựa trên ML yêu cầu dữ liệu tập trung. Để giải quyết vấn đề này, Học liên kết (FL) đã được áp dụng trong IDS, cho phép các mô hình ML được đào tạo trên các thiết bị/khách hàng phân tán mà không cần chia sẻ dữ liệu thô. Tuy nhiên, FL gặp khó khăn với dữ liệu không độc lập và đồng nhất (non-IID), dẫn đến hiệu suất của các mô hình phát hiện xâm nhập bị giảm. Bài báo này giới thiệu việc sử dụng hàm mất mát giúp học các biểu diễn cục bộ gọn nhẹ trên mỗi khách hàng và một mô hình.**

***Keywords*— Anomaly detection, deep learning, federated learning, intrusion detection systems.**

***Từ khóa*— Phát hiện bất thường, học sâu, học liên kết, học phân tán, hệ thống phát hiện xâm nhập.**

I. INTRODUCTION

IDS are essential for safeguarding network infrastructures and sensitive information from cyber threats. As cyberattacks grow increasingly sophisticated, the demand for advanced IDS solutions is more pressing than ever. Machine learning and deep learning methods have significantly bolstered IDS capabilities by identifying intricate intrusion patterns [1]. However, the deployment of ML and DL within IDS has sparked privacy concerns, primarily due to the substantial volumes of sensitive data required for training [2]. FL offers a promising solution by enabling model training on decentralized devices without needing to share raw data. Despite its advantages, FL encounters difficulties with the non-independently and identically distributed (non-IID) nature of IDS data, which can lead to performance inefficiencies [3, 4]. Personalized FL (pFL) methods have been proposed to mitigate these issues, but many fail to account for crucial aspects, such as personalization in the loss function [3].

In this work, we propose an innovative approach that incorporates a loss function designed to simultaneously learn compact local representations for each client while developing a cohesive global model [12], aimed at improving the resilience of FL for IDS in the context of non-IID data. The core objective of this loss function is to align the local representations across all clients, preventing the local models from becoming overly tailored

This manuscript was received on October 17, 2024. It was reviewed on November 27, 2024, revised on December 16, 2024 and accepted on December 17, 2024.

* Corresponding author

to their limited datasets. As a result, it enhances the global model's ability to generalize by balancing local adaptability with broader generalization. The proposed FL framework for IDS, dubbed Fed-IDS, delivers improvements in both the accuracy and robustness of the IDS system.

The main contributions of this paper are as follows:

- We propose using the loss function that jointly learns compact local representations on each client and a global model across all clients for local training.
- We introduce the FL based system for IDS named as Fed-IDS that effectively learns from both local model and global model by using above loss function.
- We perform extensive experiments on two IDS datasets to demonstrate the superiority of the proposed solution. The experimental results show that our proposed system significantly improves the accuracy and robustness of federated learning models with the non-IID data of the IDS problem.
- The rest of the paper is organized as follows. Section II presents the background of the FL system. Section III summarizes previous research on federated learning and intrusion detection systems. The proposed system is described in Section IV. The experimental settings are provided in Section V. After that, Section VI presents experimental results together with analysis. Conclusions and future works are discussed in Section VII.

II. BACKGROUND

FL is a decentralized machine learning paradigm that enables multiple clients to collaboratively train a shared model while keeping their data localized. Introduced by Google in 2016 [18], FL addresses critical challenges in modern machine learning, such as data privacy, communication efficiency, and the feasibility of training models in environments where data centralization is impractical or prohibited. Unlike traditional centralized machine learning approaches, where data is collected and processed on a central server, FL ensures that raw data remains on the clients' devices, significantly reducing the risk of data breaches and enhancing compliance with privacy regulations like GDPR and HIPAA.

A typical FL setup is illustrated in Figure 1. In this setup, each client trains a local model using its private data and sends only model updates - such as gradients or weights - to a central server. The server aggregates these updates using algorithms like Federated Averaging (FedAvg) [6] to produce a global model, which is then redistributed to the clients. This iterative process continues until the global model converges. The decentralized nature of FL makes it particularly suitable for scenarios involving sensitive or distributed data, such as IDS [7], mobile devices [8], and healthcare applications [9].

Compared to traditional centralized learning, FL offers several advantages. First, it minimizes the need for data transfer, reducing communication overhead and latency, which is especially beneficial in resource-constrained environments like edge devices or IoT networks. Second, FL inherently supports data heterogeneity, as clients may have non-IID (independent and identically distributed) data, which is common in real-world applications. Third, FL aligns with the growing demand for privacy-preserving AI, as it ensures that sensitive data, such as medical records or user activity logs, remains on local devices.

However, FL also introduces unique challenges compared to centralized learning. These include handling statistical heterogeneity across clients, ensuring secure and efficient communication, and addressing issues like stragglers (clients with slower computation or communication). Additionally, FL systems must be robust against adversarial attacks, such as model poisoning or data inference attacks, which exploit the decentralized nature of the system.

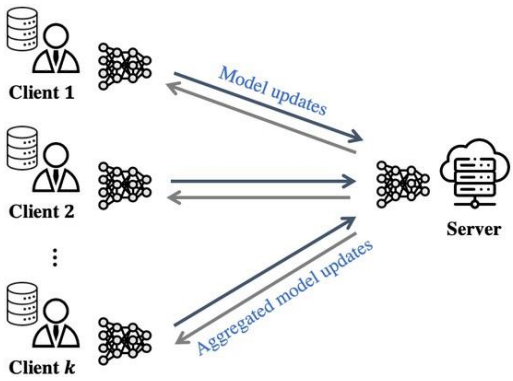

Figure 1. Description of a client model

III. Related Work

FL has gained widespread adoption in IDS as a means of enhancing security without sacrificing data privacy. FL enables collaborative training of machine learning and deep learning models across multiple devices or servers while ensuring that raw data remains localized and is never directly shared. This approach effectively mitigates privacy concerns by allowing sensitive data to remain at its origin while still contributing to a global model capable of detecting advanced cyber threats [1]. However, a key challenge in FL is managing non-IID data, where data from different clients may have diverse distributions due to factors like user behavior, environmental differences, or data collection methods [2]. This heterogeneity can impair model performance as traditional FL techniques struggle to effectively aggregate updates from clients with non-IID data.

To address the issue of non-IID data in FL, several techniques have emerged. FedRep [10] divides the model into a shared representation layer and a task-specific layer, allowing the global model to learn robust representations that can generalize across varied data distributions. FedGH [11] improves global model performance by aggregating only the predictive components of local models, thereby reducing communication costs and enhancing results in non-IID environments. Additionally, approaches such as personalized loss functions enable each client to optimize a loss function that is better suited to its local data distribution. This makes the global model more adaptable to individual client needs. For example, MOON [12] introduces a contrastive loss mechanism to align model updates from different clients, thereby improving generalization. Another method, LG-FedAvg [5], utilizes a personalized loss function that balances both global and local loss components for each client.

In our work, we propose using a local loss function, based on the one outlined in [12], to incorporate both local and global representations during local training. This loss function strike a balance between local adaptability, which captures the unique characteristics of each client's data, and global representations. By doing so, our proposed FL system, called Fed-IDS, becomes more efficient in detecting intrusions across a variety of network environments.

IV. Methodology

In this section, we present the proposed FL system for the IDS problem. First, we present the local model trained in each client. Next, we will present the training process of the FL system for IDS (Fed-IDS).

A. Local Model

In the FL framework, i.e., Fed-IDS, each client i in the set of participated clients of the FL system uses the local model trained on its data. This local training is executed after the clients receive the global model from the server. In our proposed FL system, the local model is a Multiple Layer Perceptron (MLP) [19] with three hidden layers. This model is trained to learn a representation H_i from its local data (X_i, Y_i) where X_i, Y_i are the data samples and the corresponding labels of the local data. The local training process is optimizing the parameters θ_i^{ℓ} of the local model $\theta_i^{\ell}: X_i \rightarrow H_i$.

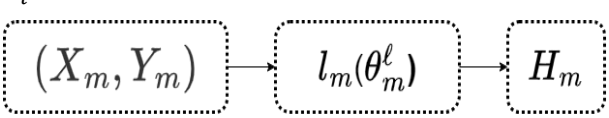

Figure 2. Description of a local model

The training process of the local model is minimizing the loss function with two terms. The first term is the Cross Entropy function [13] used for the classification purpose. The second loss term adapted from [12] is The contrastive loss

$L_{i_{CO}}$ helps a local model align with the global model and distinguish from it of the previous local model described in Equation 1.

$$L_{i_{CO}} = -\log\left(\frac{e^{\left(\frac{\cos(\mathbf{r}_i^t, \mathbf{r}_g^t)}{\tau}\right)}}{e^{\left(\frac{\cos(\mathbf{r}_i^t, \mathbf{r}_g^t)}{\tau}\right)} + 1}\right) \quad (1)$$

where $\mathbf{r}_i^t$ present the representations of the local model θ_i^ℓ at the communication round t, respectivel. $\mathbf{r}_g^t$ presents the representation of the global model θ_i^g at the communication round t, τ is is the temperature that adjusts the sensitivity of the softmax function, and cos is the cosine similarity. The cosine similiarity between two vector a and b is calculated by Eq. 1 bellows.

$$cos(a,\ b)\ =\ \frac{(a \cdot b)}{(||\ a\ ||||\ b\ ||)} \quad (2)$$

Specifically, the local training process of the client i is described as in Algorithm 1 The inputs of this algorithm are the local data (X_i, Y_i), the global model θ_i^g sent from the server to the client i and some hyper-parameters such as the learning rate, the number of local epochs. For each local epoch, the local data is fit to the local model θ_i^ℓ and the global model θ_i^g to calculate the gradients $\nabla_{\theta_i^\ell}\mathcal{L}_i^\ell(\theta_i^\ell, \theta_i^g)$ (as in the line 3) and $\nabla_g\mathcal{L}_i^{\theta_i^g}(\theta_i^\ell, \theta_i^g)$ (as in the line 4), respectively where $\mathcal{L}$ is calculated as Equation 1 After complete P epochs, the client obtains the local model θ_i^ℓ (as in the line 7) that will be sent to the server for aggregation.

The Cross Entropy loss element is the supervised loss which is utilized to classify the local data into specific classes. The supervised loss function used in the proposed system is the CE loss. The CE loss function for a client i is defined as follows:

$$\mathcal{L}_{i_{CE}} = -\sum_{c=1}^{C}(y_c \times \log\phi_i(x_c), \quad (3)$$

where C is the number of data classes. y_c is the class of the data x_c, $\phi_i(x_c)$ is the output of x_c with the model ϕ_i.

Drawing from two loss elements above, the loss function for each client i of the proposed system is defined as follows:

$$\mathcal{L}_i^g(\theta_i^\ell, \theta_i^g) = \mu * \mathcal{L}_{i_{CO}} + \mathcal{L}_{i_{CE}}, \quad (4)$$

where $\mathcal{L}_{i_{CO}}$ and $\mathcal{L}_{i_{CE}}$ are the loss elements calculated in Eq. 1 and Eq. 3, respectively. μ is a hyperparameter used to adjust the weight of the contrastive loss element to the loss function $\mathcal{L}_i$[1].

Algorithm 1 Local training for a client.

1: **Input:** Local data X_i and label Y_i, Global model θ_i^g, P is the number of epoch for local training, the learning rate η_ℓ and η_g for training θ_i^ℓ and θ_i^g, respectively.

2: **for** each local epoch p = 1,2,... P **do**

3: Feed the data (X_i,Y_i) into the local model θ_i^ℓ and the global model θ_i^g to calculate the loss function as in Equation 1;

4: **Update** $\theta_i^\ell \leftarrow \theta_i^\ell - \eta_l\nabla_{\theta_i^\ell}\mathcal{L}_i^\ell(\theta_i^\ell, \theta_i^g)$.

5: **Update** $\theta_i^g \leftarrow \theta_i^g - \eta_g\nabla_{\theta_i^\ell}\mathcal{L}_i^\ell(\theta_i^\ell, \theta_i^g)$

6: **end for**

7**: Output:** θ_i^ℓ, θ_i^g

B. Fed-IDS

In the Fed-IDS framework, each client trains a model using its local data, as detailed in Section I and Algorithm 2. The system is based on the Federated Averaging (FedAvg) approach [6]. Initially, the server initializes the global model θ^g and transmits it to a randomly selected subset of clients, denoted as S_t.

During each training round t, selected clients initialize their local models, θ_i^l, and train on their local datasets, updating their models based on unique data patterns. After local training, clients send their updated models, θ_i^g, back to the server. The server then aggregates these updates to create a unified global model, which is broadcast to clients for the next round.

At the end of all training rounds, each client has a final local model θ_i^{ℓ}, fine-tuned with both. Local data and insights from the global model. This process enables Fed-IDS to facilitate collaborative learning while ensuring data privacy and robustness against variations in clients' data distributions.

Algorithm 2 Training procedure of Fed-IDS.

1: **Input:** The training round *T*, the number of clients *M*, the global model θ^g, the local models θ_i^{ℓ} where $i = 1,2,...M$ and local training data of *m* clients, the participant rate *r* in the range (0,1].
2: Server initializes the global model θ^g with weights $\theta^{g\,(0)}$;
3: All clients initialize the local models with weights $\theta_i^{\ell(0)}$ where $i = 1,2,...M$;
4: **for** each round $t = 1,2,...$ T **do**
5: $S_t \leftarrow$ (random set of $r \times M$ clients),
6: **for** each client $i \in S_t$ **in parallel do**
7: Set the global model $\theta_i^g = \theta_i^{g\,(t-1)}$,
8: Set the local model $\theta_i^l = \theta_i^{l\,(t-1)}$,
9: **Update**,
10: $\theta_i^{\ell}, \theta_i^g \leftarrow$ **Local Training**$(\theta_i^{\ell}, \theta_i^g)$
11: Send θ_i^g to the serve for aggregation,
12: **end for**
13: Server aggregation: $\theta^g(t) \leftarrow \frac{1}{M}\sum_{i=1}^{M} \theta_i^{g(t)}$
14: **end for**
15: **Output** The local model θ_i^{ℓ} for each client

For the IDS problem, after training, each client has the local model θ_i^{ℓ} used to detect the network attacks for IDS. Specifically, each client collects real-time network traffic data. The data is then preprocessed and labeled according to known attack patterns and normal traffic. Next, the collected network traffic is transformed into feature vector that can be input into the local model as the training data [14, 16]. The preprocessed data is inputted the trained local model θ_i^{ℓ} of the Fed-IDS. This model predicts the output H_i to identify whether the traffic is normal or a type of attacks. Based on that, the client determines whether an alert is raised or not, and appropriate actions are taken (e.g., logging the event, notifying administrators, or block).

V. Experimental setting

A. Datasets

NSLKDD Dataset

The NSLKDD dataset [14] is particularly useful for bench-marking IDS because it contains a wide variety of normal and malicious network traffic. This dataset consists of five class labels, where label 0 represents normal traffic, and the remaining labels represent different groups of abnormal traffic. The number of data samples for the training and testing datasets are 126000 and 22545, respectively.

CICIDS Dataset

The CICIDS dataset [16] focuses on attacks commonly observed on Internet of Things (IoT) devices. The CICIDS dataset includes several types of attacks tailored to these devices, making it an essential dataset for assessing IDS performance in IoT contexts. We randomly divide this dataset into the training and the testing set with the number of data samples as 333190 and 80640, respectively.

For each dataset, we divided the data among the clients in two different scenarios. First, the Independent and Identically Distributed (IID) setting has evenly distributed across clients where each client has a balanced mix of all classes. In the non-IID setting, we distributes the training data to the clients in a skewed manner where each client has a different distribution of classes.

B. System Configuration

All the experiments in this work are performed on the computing cluster with the hardware as Intel Core i7-9700K CPU @ 3.60GHz, 32GB RAM, NVIDIA GTX 1080 Ti and the software as Python 3.8, TensorFlow 2.4, and PyTorch 1.7. The simulation environment is managed using Docker containers to ensure the consistent performance across multiple runs.

A local model of a FL setting in this experiment is a Deep Neural Network (DNN) with the following architecture. The number of

neurons in the input layer is equal to the number of features of the experimental dataset. It is followed by three hidden layers with 64, 32 and 16 neurons, respectively. The output layer has the number of neurons being equal to the number of classes. The number of epoch in local training is 10 and the learning rate is 0.005. The number of training rounds is set as 2000.

We evaluate the performance of several FL systems, including FedAvg [6], FedGH [11], FedPer [17], FedRep [10], and Fed-IDS, under both the IID and non-IID data distribution settings. The same local model architecture is used in all these FL systems. The hyper-parameters are set as following: the learning rate 0.01, the batch size as 64, the number of communication round T as 100, and the number of local epoch training P as 5.

C. Evaluation Metrics

The performance of the FL systems is evaluated using following metrics on the testing set:

- **Accuracy**: The overall correctness of the model's predictions, indicating how well it distinguishes between benign and malicious traffic.
- **Precision and Recall**: These metrics are particularly important in IDS, as they reflect the model's ability to correctly identify malicious traffic (precision) and its effectiveness in detecting all instances of attacks (recall).
- **F1-Score**: A harmonic mean of precision and recall, providing a balanced measure of the model's performance, especially in the presence of class imbalance.

VI. Result and discussion

The experiments are conducted to evaluate the performance of various FL systems on network intrusion detection using two benchmark IDS datasets, i.e., CICIDS and NSLKDD. In this section, we present the performance of the FL systems in both the IID and non-IID setting.

A. Performance on IID Setting

Table 1 and 2 show the results of different FL systems on the NSLKDD dataset and the CICIDS dataset, respectively, with the IID setting.

The results presented in Table I shows that almost the experimental FL systems achieve the high accuracy on the NSLKDD dataset. However, the Precision, Recall, and F1 scores are relatively low for FedAvg, FedPer, FedRep, and FedGH techniques.

TABLE 1. EXPERIMENTAL RESULTS OF THE FL METHODS ON THE NSLKDD DATASET WITH IID SCENARIO

Algorithm	Accuracy	Precision	Recall	F1
FedAvg	0.98	0.76	0.75	0.75
FedPer	0.98	0.83	0.82	0.82
FedRep	0.98	0.79	0.79	0.79
FedGH	0.92	0.61	0.61	0.60
Fed-IDS	0.98	**0.85**	**0.86**	**0.84**

Enhances the Precision, Recall, and F1 scores. This indicates that the Fed-IDS system is more effective to detect both normal traffic and attack traffic of the IDS problem.

TABLE 2. EXPERIMENTAL RESULTS OF THE FL METHODS ON THE CICIDS DATASET WITH IID SCENARIO

Algorithm	Accuracy	Precision	Recall	F1
FedAvg	0.92	0.32	0.32	0.31
FedPer	0.96	0.54	0.53	0.51
FedRep	**0.99**	0.84	0.84	0.83
FedGH	0.83	0.27	0.29	0.25
Fed-IDS	**0.99**	**0.90**	**0.87**	**0.87**

The similar observation is shown in Table 2 that highlights the performance of the FL methods on the CICIDS dataset with the IID setting. Among the FL systems, the Fed-IDS system exhibits strong performance, specifically the Accuracy 0.99, the Precision 0.90, the Recall as 0.87 and the F1-score as 0.87. These results indicate that Fed-IDS particularly effective at maintaining high performance for the IDS problems. The reason is that the loss function used in the local training of Fed-IDS can learn from both local and global model effectively.

However, we can observe a performance imbalance between the NSLKDD and CICIDS datasets, which can be attributed to several key factors. The CICIDS dataset contains more complex and diverse attack patterns from IoT devices, including modern network traffic and attack scenarios, making it more challenging for

model classification. Additionally, while NSLKDD has a well-balanced distribution of classes with 126,000 training samples and 22,545 testing samples, CICIDS has a larger and potentially more imbalanced distribution with 413,830 records that were split into training and testing sets. The difference in data characteristics, attack types, and class distributions between these datasets significantly impacts the model's performance.

B. Performance on non-IID Setting

Table 3 and Table 4 show the results of different FL systems on the NSLKDD dataset and the CICIDS dataset, respectively, with the non-IID setting. We can observe that these FL systems generally perform better with the IID setting as presented in the previous section due to the uniform distribution of data across clients. However, in the non-IID setting, the experimental FL systems show varying degrees of performance degradation. The FedAvg is not designed for the non-IID setting, and thus, it performs very ineffective on the both NSLKDD and CICIDS datasets. The F1 scores of FedAvg are smaller than 0.5 showing the negative impact of the non-IID setting for a standard FL system.

Moreover, these tables also show that the personalized FL systems, i.e., FedPer, FedRep, FedGH, and Fed-IDS, can enhance the performance of the IDS problem with the non-IID setting. The reason is that these personalized FL techniques can assist the global model to fit with the local training, and thus, improving the accuracy of the local model with its local data. Moreover, the strong performance of Fed-IDS across both datasets underscores their suitability for FL tasks, offering a good balance between accuracy, precision, and recall. Notably, Fed-IDS demonstrates superior performance in non-IID settings compared to IID scenarios, as evidenced in III. The reason is that the loss function of Fed-IDS prevents the local model in the FL setting from overfitting to its own local data. Thus, in the local training, the local adaptation and global generalization are both considered to improve the generalization ability of the global model. As a result, it enhances the performance of the IDS problem.

TABLE 3. EXPERIMENTAL RESULTS OF THE FL METHODS ON THE NSLKDD DATASET WITH THE NON-IID SCENARIO

Algorithm	Accuracy	Precision	Recall	F1
FedAvg	0.88	0.43	0.40	0.41
FedPer	0.99	0.97	0.99	0.98
FedRep	0.99	0.99	0.99	0.99
FedGH	**1.0**	**0.99**	**0.99**	**0.99**
Fed-IDS	**1.0**	**0.99**	**0.99**	**0.99**

TABLE 4. EXPERIMENTAL RESULTS OF THE FL METHODS ON THE CICIDS DATASET WITH THE NON-IID SCENARIO

Algorithm	Accuracy	Precision	Recall	F1
FedAvg	0.97	0.30	0.34	0.30
FedPer	0.98	0.66	0.77	0.63
FedRep	0.99	0.85	0.80	0.82
FedGH	0.94	0.50	0.51	0.50
Fed-IDS	**1.0**	**0.88**	**0.83**	**0.85**

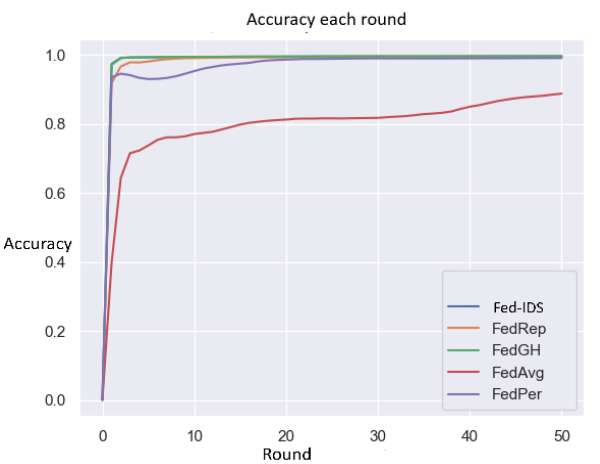

Figure 3. Accuracy of FL systems on NSL-KDD (non-IID)

The superior performance of non-IID over IID settings in Fed-IDS can be attributed to several key factors. In the non-IID environment, personalized FL techniques enable local models to better adapt to their specific data distributions while maintaining alignment with the global model. This improvement occurs because the loss function in non-IID settings prevents local models from overfitting to their own data while promoting both local adaptation and global generalization. The combination of local specialization and global knowledge sharing allows models to learn more robust and diverse features, leading to enhanced generalization ability. Additionally, the non-IID setting better reflects real-world scenarios where different

clients have varying data distributions, making the trained models more practical and effective for actual deployment in intrusion detection systems.

VII. Conclusion

In this paper, we have conducted the number of researches on the (FL for the intrusion detection system IDS. Based on that, we propose using the loss function that jointly learns compact local representations on each client and a global model across all clients for local training. This helps the proposed FL system for IDS (Fed-IDS) to learn the IDS data effectively. The experimental results show that using Fed-IDS enhances the accuracy of the IDS problem in the case of non-IID data distribution. The loss term of Fed-IDS acts as a regularizer to synchronize the local representations learned from the participated clients. This prevents the local model from overfitting to its own local data. As a result, the accuracy of the IDS problem is increased significantly in the two benchmark IDS datasets.

Our future work will focus on three main directions to enhance the pFLOCL system. First, we will explore various model architectures (CNN, LSTM, transformers) beyond the current DNN structure. Second, we will investigate system scalability by testing with different numbers of clients and analyzing performance-communication trade-offs. Third, we will experiment with diverse non-IID data distribution scenarios. Additionally, we plan to conduct comparative studies with other state-of-the-art federated learning approaches in IDS to better understand our system's strengths and limitations.

References

[1] D. Musleh, M. Alotaibi, F. Al-Haidari, A. Rahman, and R. Mohammad, "Intrusion detection system using feature extraction with machine learning algorithms in IoT," in *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, pp. 1–19, 2023. doi: https://doi.org/10.3390/jsan12020001.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., "Advances and open problems in federated learning," arXiv preprint arXiv:1912.04977, 2019. doi: https://doi.org/10.48550/arXiv.1912.04977.

[3] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds*, vol. 33, Curran Associates, Inc., 2020, pp. 3557–3568.

[4] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," arXiv preprint arXiv:1703.03400, 2017. doi: https://doi.org/10.48550/arXiv.1703.03400.

[5] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," arXiv preprint arXiv:2001.01523, 2020. doi: https://doi.org/10.48550/arXiv.2001.01523.

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

[7] M. Bhavsar, Y. Bekele, K. Roy, J. Kelly, and D. Limbrick, "FL-IDS: Federated learning-based intrusion detection system using edge devices for transportation IoT," in *IEEE Access*, vol. PP, pp. 1–1, Jan. 2024. doi: https://doi.org/10.1109/ACCESS.2024.1234567.

[8] C. You, K. Guo, G. Feng, P. Yang, and T. Q. S. Quek, "Automated federated learning in mobile-edge networks—fast adaptation and convergence," in *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13571–13586, Aug. 2023. doi: https://doi.org/10.1109/JIOT.2023.3262664.

[9] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," in *IEEE Intelligent Systems*, vol. 35, pp. 83–93, 2020. doi: https://doi.org/10.1109/MIS.2020.2994885.

[10] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning*, 2021, pp. 2089–2098.

[11] L. Yi, G. Wang, X. Liu, Z. Shi, and H. Yu,

"FedGH: Heterogeneous federated learning with generalized global header," arXiv preprint arXiv:2303.13137, 2023. doi: doi.org/10.48550/arXiv.2303.13137.

[12] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10713–10722. doi: https://doi.org/10.1109/CVPR.2021.01072.

[13] G. Legate, L. Caccia, and E. Belilovsky, "Re-weighted softmax cross-entropy to control forgetting in federated learning," arXiv preprint arXiv:2301.12345, 2023. doi: https://doi.org/10.48550/arXiv.2301.12345.

[14] J. Kaliappan, T. Revathi, and K. Sundararajan, "Intrusion detection using artificial neural networks with best set of features", *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp.728-734, Jan. 2015.

[15] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 7611–7623, 2020.

[16] B. Reis, E. Maia, and I. Praça, "Selection and performance analysis of CICIDS2017 features importance," in *Selection and Performance Analysis of CICIDS2017 Features Importance*, pp. 56–71, Apr. 2020.

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

[18] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. doi: https://doi.org/10.48550/arXiv.1610.05492.

[19] T. N. Quy, N. T. Tung, D. Q. Trung, and D. H. Viet, "Convolutional neural network based sidechannel attacks", *Journal of Science and Technology on Information Security*, vol. 1, no. 15, pp. 26-37, Jun. 2022. doi: https://doi.org/10.54654/isj.v1i15.834.

ABOUT THE AUTHOR

Ha Xuan Tung
Workplace: Le Quy Don Technical University.
Email: xuantungmta@gmail.com
Education: 2013-2018: Engineer of Information Technology.
Recent research direction: Information security.

Tên tác giả: **Hà Xuân Tùng**
Cơ quan công tác: Đại học Kỹ thuật Lê Quý Đôn.
Email: xuantungmta@gmail.com
Quá trình đào tạo: 2013-2018: Kỹ sư Công nghệ thông tin.
Hướng nghiên cứu hiện nay: An toàn thông tin

Tran Linh Trang
Workplace: Political University.
Email: Trangtrang5225@gmail.com
Education: She graduated from University of Languages and earned a master's degree from University of Languages and International Studies in in 2023.
Recent research direction: Teaching methods, Technology-Enhanced Language Learning.

Tên tác giả: **Trần Linh Trang**
Cơ quan công tác: Trường Đại học Chính trị.
Email: Trangtrang5225@gmail.com
Quá trình đào tạo: Tốt nghiệp đại học năm 2018, tốt nghiệp thạc sĩ năm 2023
Hướng nghiên cứu hiện nay: Phương pháp giảng dạy, Ứng dụng Công nghệ trong học ngoại ngữ.

Vo Khuong Linh
Workplace: Nguyen Hue University.
Email: vokhuonglinh@gmail.com
Education: He received his Engineering Degree in Information Technology and Master Degree in Computer Science from Le Quy Don Technical University.
Recent research direction: Information security.

Tên tác giả: **Võ Khương Lĩnh**
Cơ quan công tác: Đại học Nguyễn Huệ
Email: vokhuonglinh@gmail.com
Quá trình đào tạo: Tốt nghiệp Đại học năm 2017 chuyên ngành CNTT và tốt nghiệp Thạc sĩ năm 2024, chuyên ngành Khoa học máy tính tại trường Đại học kỹ thuật Lê Quý Đôn.
Hướng nghiên cứu hiện nay: An toàn thông tin