Personalized federated learning-based intrusion detection system: Poisoning attack and defense

Thein, Thin Tharaphe
Shiraishi, Yoshiaki
Morii, Masakatu

(Citation)
Future Generation Computer Systems,153:182-192

(Issue Date)
2024-04

(Resource Type)
journal article

(Version)
Version of Record

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Personalized federated learning-based intrusion detection system: Poisoning attack and defense

Thin Tharaphe Thein [*], Yoshiaki Shiraishi, Masakatu Morii

Department of Electrical and Electronic Engineering, Kobe University, Kobe, Hyogo 657-8501, Japan

ARTICLE INFO

Keywords:
Personalized federated learning
Intrusion detection system
Poisoning attacks
Poisoning defense

ABSTRACT

To deal with the increasing number of cyber-attacks, intrusion detection system (IDS) plays an important role in monitoring and ensuring the security of the computer network. With the power of machine learning and deep learning, intelligent IDS systems have gained increasing attention due to their efficiency and high classification accuracy. However, the premise of machine learning/deep learning is that the data must be in one central entity (e.g., server) to train the model. This causes additional concerns, such as data transmission costs and privacy leakage. Federated learning complements this shortcoming with a privacy-preserving decentralized learning technique. In federated learning, the data are not shared with the server, local model training is performed where the data reside and only the model parameters are exchanged with the server. This work investigates the federated learning-based IDS approach in the context of IoT data to study the main challenges imposed by federated learning. Two main issues, such as data heterogeneity and poisoning attacks launched by malicious clients, are the main focus of this study. As real-world IoT datasets are heterogeneous, we propose a personalized federated learning-based IDS approach to handle imbalanced data distributions. Moreover, a curious yet malicious client can poison the local data or model to corrupt the global intrusion detection model due to the distributed nature of federated learning, where the central server has no control over the client's local training process. This study demonstrates that the existence of a malicious client can degrade the performance of the federated learning-based IDS model. Accordingly, we propose a robust approach called pFL-IDS to combat poisoning attacks against the federated learning-enabled IDS on heterogeneous IoT data. Our approach introduces mini-batch logit adjustment loss to local model training to obtain a personalized model tailored to each local data distribution. Moreover, we design a detection mechanism at the server to identify malicious agents by considering the cosine similarity of local models from the non-poisoned client's centroid. The non-poisoned centroid is determined from the similarity between the pre-computed global model and the local models. If the poisoning attack is successful, poisoned clients will be closer to the pre-computed global model; any models further from the pre-computed model are taken as the non-poisoned clients. With this two-phase client similarity alignment, we identify poisoned clients and restrict their aggregation on the global intrusion detection model. In comparison with the baseline methods, we demonstrate that our pFL-IDS can detect poisoning attacks without compromising performance.

1. Introduction

With the rapid development of the Internet, the world has become more connected, and the deployment of the Internet of Things (IoT) devices has increased. It is reported that IoT devices will reach 55.7 billion by 2025 [1]. The massive amount of data generated by these devices, combined with their vulnerable nature, open more attack interfaces and opportunities for malicious parties to conduct cyber-attacks [2]. In 2016, the infamous Mirai botnet attack triggered Internet security breaches by compromising massive IoT devices [3]. This further signifies the need to strengthen the security of the IoT network ecosystem to protect the network from cyber-attacks. Therefore, an in-depth network traffic analysis is necessary; the intrusion detection system plays a vital role in detecting and mitigating unwanted attacks.

To date, different techniques have been utilized in network intrusion detection systems to detect network anomalies, which are either signature-based or behavior-based approaches or a combination of both [4]. Combined with its efficiency and effectiveness, the machine learning/deep learning-based IDS was proven as a perfect candidate with high detection accuracy. However, the traditional machine

[*] Corresponding author.
E-mail address: thein.thin@gsuite.kobe-u.ac.jp (T.T. Thein).

https://doi.org/10.1016/j.future.2023.10.005
Received 1 March 2023; Received in revised form 9 July 2023; Accepted 6 October 2023
Available online 24 October 2023

learning-based method requires the data to be in one central place to analyze the data gathered from all user devices. This increases the data transmission costs and the risk of privacy leakage as the data from user devices contain sensitive or confidential information. To resolve these issues, McMahan et al. proposed a privacy-preserving and distributed federated learning paradigm for on-device learning [5]. Federated learning is a client–server architecture that comprises multiple clients tied to a central server. It enables clients/devices to train a shared model collaboratively without the actual data exchange to guarantee user privacy, reduce data transmission costs, and improve the overall model accuracy. In the context of IoT networks, research on federated learning-based IDS has been emerging recently [4]. In this system, the clients train the global intrusion detection model on their local dataset to compute local model updates and upload them to the server. The server combines local models to generate the global intrusion detection model, which is sent back to the clients for further training. This process is repeated until the global model converges.

Despite its benefit, federated learning still has some limitations, such as difficulty in handling the heterogeneity in the data distribution of the client and the poisoning attacks executed by malicious clients [4,6,7]. In realistic scenarios, the client data are confirmed non-independent and identically distributed (non-IID) in contrast to the ideally assumed IID scenarios in many research methodologies. In federated learning-based IDS, some clients (devices) may only have attack traffic while others may have normal traffic or consist of different kinds of attack traffic. Moreover, as demonstrated in recent studies [8,9], federated learning-based IDS is prone to data poisoning and model poisoning attacks as the server has no control over the behavior of the client. Poisoning on data can be done either by modifying the ground truth labels of the client dataset (label-flipping attack) or injecting false training data (clean label attack). In model poisoning, malicious clients can manipulate local model parameters to align the global model's objective closer to the attacker's objective. Both poisonings can cause misjudgment in the IDS, for instance, recognizing benign traffic as attack traffic or vice versa. There has been extensive research on minimizing the influence of the malicious client using robust aggregation techniques in the server [10,11] or detecting and removing poisoned clients [12–14] before the global model aggregation.

In this study, we propose the federated learning-based IDS that can effectively detect network anomalies of IoT data. Our proposal explores two topics: the effect of data heterogeneity and the behavior of the federated learning-based IDS when poisoning attacks occur. We model two poisoning attacks, such as label-flipping [15] and model update poisoning [16], to understand the behavior of the poisoned clients against the federated learning-based IDS. First, we analyze the consequence of these poisoning attacks and evaluate how well the existing robust server aggregators [10,11] can defend against these poisoning attacks. We find that the existing robust aggregators cannot effectively mitigate the influence of the poisoned clients as the heterogeneity of the client's data distribution increases, thus degrading the performance of federated learning-based IDS. The imbalanced data distribution can negatively impact the federated learning model even if poisoned clients do not exist. As the main objective of federated learning is to improve a single global model from all clients, client data distribution drift (i.e., heterogeneous data) can result in poor convergence on the global model. Fig. 1 compares the FedAvg aggregation technique applied to IID and non-IID data scenarios. It has two clients, each performing two local updates. Client 1 updates the initial global model, denoted as w, to its optimal solution, w_1^*. Similarly, client 2 updates w to its optimal solution, w_2^*. In Fig. 1(b), it can be observed that the two client models move in different directions during the update process, primarily due to the non-identical distribution of data. Consequently, when averaging the two client models to obtain the final global model, the global model does not converge to its optimal position w^*, making it difficult to obtain the optimal global model.

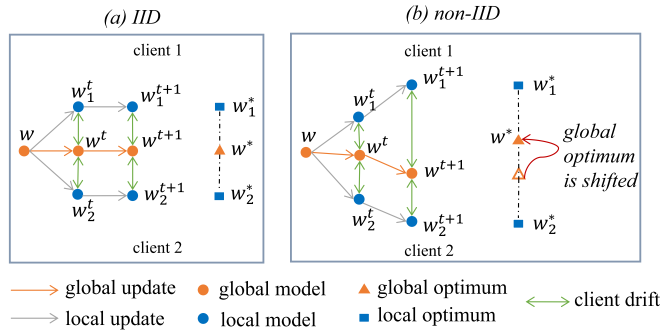


Fig. 1. Illustration of client drift in FedAvg algorithm for two clients with 2 local update steps. (a) IID data. The averaged global optimum w^* is located equally from both clients' local optima w_1^* and w_2^*. (b) non-IID data. The averaged global optimum w^* is not equidistant from both local optima due to client drift caused by client imbalanced data distribution. As a result, the global optimum w^* is deviated from the true global optimum, and the global model cannot converge.

Various techniques have been introduced to compensate for the problem of the non-IID data, such as multi-task learning, model clustering, model parameter decoupling, and knowledge distillation [17]. Our approach deals with the non-IID data through model decoupling strategy. Decoupling means that the neural network model is decoupled into two parts: feature extractor (body) and classifier (head). Then, we introduce the mini-batch logit adjustment loss to the head classifier in addition to the cross-entropy loss, which is explained in Section 3.2, to make up for the non-IID data. With these two losses, each client optimizes its local model based on the importance of the underlying class distribution, making the global model more resilient to the non-IID distribution.

We design the poisoned client detector on the server that can distinguish the poisoned clients from the non-poisoned clients and restrict their participation from the global model aggregation to achieve our objective of reducing the impact of poisoned clients on the global intrusion detection model. The concept of detecting poisoned clients has been explored in various studies [12–14]. However, existing approaches assume that the ratio of poisoned clients is small or need a clean dataset at the server to compare the client model with the server model or did not tailor for the non-IID data. Our detection approach can identify poisoned clients without drastically degrading the performance, and it does not require an additional clean dataset at the server. In this paper, we analyze the behavior of label-flipping attacks and model update poisoning attacks against the federated learning-based IDS for IoT data under non-IID scenarios. The main contributions of our study are summarized as follows.

1. We propose personalized federated learning for intrusion detection system (pFL-IDS), a robust federated learning-based IDS approach with a poisoned client detector, to handle IoT data heterogeneity and poisoning attacks.
2. With the logit adjustment loss, we model the personalized federated learning that is customized to the data distribution of the client. We demonstrate that personalized logit adjustment is suitable if the data distribution between clients varies considerably, which is common in IoT data.
3. To mitigate the influence of the poisoned clients from the global intrusion detection model, we design a poisoned client detector at the server. Our detector can identify poisoned clients and limit their participation in the global intrusion detection model aggregation. The detector first analyzes the last layer of all client models to determine the model updates that are poisoned. Our approach is based on two-step client similarity alignment, followed by re-weighting. In the first step, we take the pre-computed global model by averaging all local models.

Table 1
A summary of existing works on FL-based IDS.

Reference	Personalized FL	Defense methods for poisoning attacks	Neural network	Main contributions
Val et al. [8]	–	Coordinate-wise median and trimmed mean	MLP, Autoencoder	Evaluate the resilience of FL models against malicious clients
Popoola et al. [19]	–	–	DNN	Detection of zero-day IoT botnet attack
Fan et al. [20]	✓	–	CNN	Learn customized IDS model using federated transfer learning
Mothukuri et al. [21]	–	–	GRUs with a Random Forest ensembler	Combine the output of different GRUs layers with random forest ensembler
Attota et al. [22]	–	–	ANN	Utilize various data views of IoT traffic to maximize the detection accuracy
Ferrang et al. [6]	–	–	DNN, CNN, and RNN	Comprehensive survey and experimental analysis of FL for cyber security
Ours (pFL-IDS)	✓	Poisoned client detector	CNN	Personalized FL-based IDS for non-IID data with a poisoned client detector

As all poisoned clients have the same objective, the poisoned model should be closer to the pre-computed global model if the poisoning attempt is successful. Considering this, we can identify a potential non-poisoned client. In the second step, we compute the angular deviation between local model updates from the dimensionality-reduced centroid of the non-poisoned client. The non-poisoned client will have a smaller angular deviation from the centroid compared to the poisoned client. With that angular similarity, we re-weight each model and take the average of the potentially clean models as a global model for the next round.

4. pFL-IDS is evaluated using the N-BaIoT dataset to study the effectiveness of our approach [18]. We sample different data distribution scenarios from the dataset to simulate the imbalanced data and experiment with different poisoning attacks. We also compare our poison detector with other state-of-the-art methods [10,11,13]. In addition, we demonstrate that our pFL-IDS achieves better performance in the detection of IoT network traffic anomaly. Even if some clients are poisoned in the federated training process, pFL-IDS can reduce the attack success rate, effectively mitigating the negative impact of poisoning attacks from the global intrusion detection model.

The rest of the paper is organized as follows. The background of federated learning, intrusion detection system, and poisoning attacks against the federated learning-based IDS are provided in Section 2. Section 3 presents our proposed method, consisting of the personalized local model and server-side poisoned client detector. The detailed experiment settings and results and the impact of the poison attacks are provided in Section 4. Finally, Section 5 presents the conclusions and future works.

2. Preliminaries

This section provides the background related to federated learning, intrusion detection, existing studies on the federated learning-based IDS, and poisoning attacks and defenses against the federated learning paradigm.

2.1. Federated learning

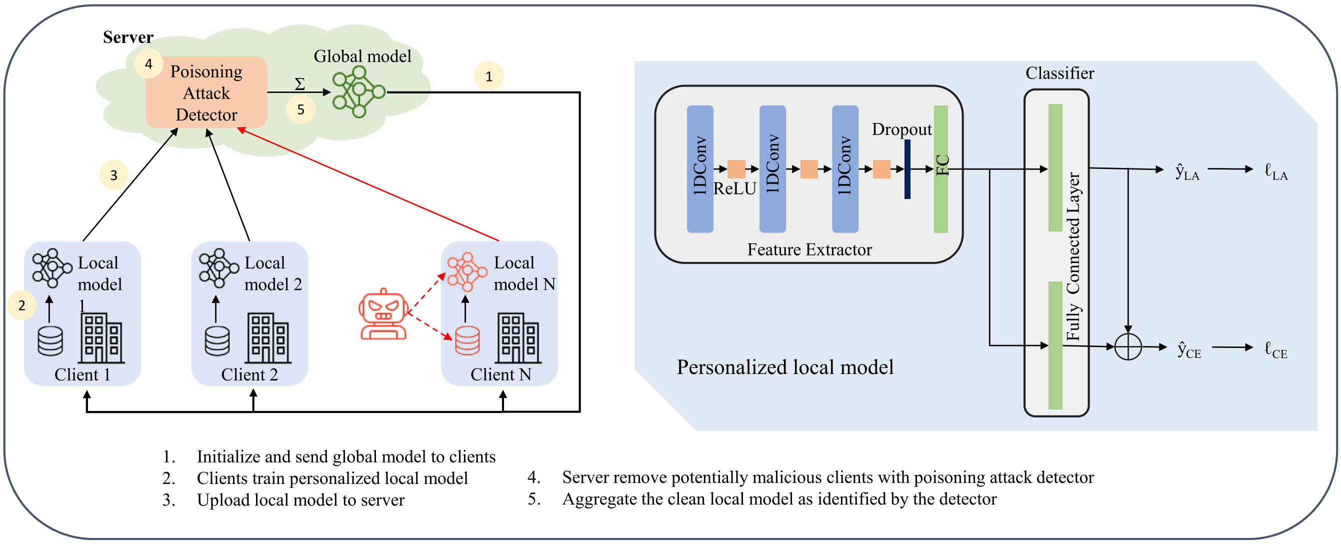
In federated learning, each client trains the local model using local data and shares the trained model parameters (not data) with the rest of the clients through a central server in a peer-to-peer manner. The central server combines all local models to create a unique global server model that is shared back with the client for further training. After several communication rounds, the global model that contains the knowledge of multiple clients is obtained. As federated learning is iteratively trained, the learning model can be improved in each round, improving the accuracy. In summary, the federated learning process includes local training, model parameters exchange between clients and the server, and global model aggregation. FedAvg [5] is the mainstream server aggregation algorithm to update the global model by taking a weighted average of the model parameters from clients.

Depending on the type of data available and the model exchange method, federated learning can be categorized into horizontal federated learning, vertical federated learning, and federated transfer learning. If each client has a different dataset with the same feature spaces, horizontal federated learning is applicable. On the other hand, vertical federated learning is suitable if each client has common entities with different feature spaces. Federated transfer learning is useful if each client has different domains or related tasks (e.g., the task of transferring a pre-trained image classification model to video classification). The horizontal federated learning scenario is compatible with the intrusion detection system since each client has different network traffic samples with the same feature spaces.

2.2. Intrusion detection system (IDS)

IDS plays an important role in protecting the network by identifying potential malicious attacks through network monitoring and analysis. IDS can be generally divided into signature-based and anomaly-based approaches [4]. The former compares the attacks with the predefined signature database to identify known attacks with a high recognition rate, but it cannot recognize new type of attacks consequently. The anomaly-based approach overcomes the previous shortcoming using artificial intelligence that learns specific feature patterns of the traffic such that the deviation from the observed behavior is regarded as an anomaly. However, the traditional machine learning-based method requires the data to be in one place to analyze the data gathered from all user devices. This increases data transmission costs and the risk of privacy leakage since the data from user devices may contain sensitive or confidential information.

Privacy-preserving federated learning has been extensively applied in the context of network intrusion detection systems due to its reputation of no need to exchange private data. Val et al. [8] proposed federated learning-based IoT network anomaly detection based on both supervised deep learning and unsupervised autoencoder. Moreover, they consider the presence of adversarial poisoning attacks and evaluate how well the existing robust aggregators [10,11] can mitigate the impact of the poisoning attacks. Popoola et al. [19] proposed zero-day botnet attack detection for IoT edge devices and demonstrated that federated learning-based DNN outperformed the conventional DNN model in terms of attack detection accuracy, low communication overhead, and data privacy. Fan et al. [20] proposed IoTDefender, a transfer



1. Initialize and send global model to clients
2. Clients train personalized local model
3. Upload local model to server
4. Server remove potentially malicious clients with poisoning attack detector
5. Aggregate the clean local model as identified by the detector

Fig. 2. The architecture of our pFL-IDS system.

learning-based IDS for 5G IoT. IoTDefender combined data using federated learning and learned personalized attack detection models by transfer learning while ensuring privacy. Monthukuri et al. [21] proposed federated learning-based anomaly detection for IoT networks based on GRUs. They improved the classification accuracy by combining the predictions from different layers of GRUs models with an ensemble of the random forest of models, demonstrating the improvement of the attack detection compared to centralized machine learning. Attota et al. [22] proposed MV-FLID, an ensemble multi-view federated learning for IoT intrusion detection. MV-FLID learns three separate ANN models for three views of network traffic (i.e., uniflow, bi-flow, and packet). The predictions of these models are combined through random forest to improve the attack detection accuracy. Ferrag et al. [6] provided a comprehensive survey on federated learning for IoT intrusion detection systems. They demonstrated that federated learning outperformed centralized learning by evaluating three IoT traffic datasets with different deep learning models. The aforementioned studies, as summarized in Table 1, demonstrated the effectiveness of federated learning for the intrusion detection; however, most of them did not consider data heterogeneity and how to combat the possible poisoning attacks launched by malicious clients. Therefore, we consider both issues to improve federated learning-based IDS.

2.3. Poisoning attacks against federated learning-based IDS

In this section, we discuss the behavior of the poisoning attacks against the federated learning-based IDS. Considering this, we assume that the federated server is not compromised and only consider the poisoning attacks that can occur at the client. The number of attackers is assumed to be less than 50% of the total client. We define the attacker's goals and capabilities as follows.

Attacker's goals: The attacker aims to corrupt the global intrusion detection model either by label-flipping or model poisoning such that the global model outputs wrong predictions on the IoT network traffic. For instance, the global intrusion detection model will misjudge malicious IoT traffic as benign or vice versa and reduce the performance of the model.

Attackers' capability: The attacker can modify the labels of the local dataset or the trained model parameters to achieve the desired goals, but it cannot modify the data.

This study investigates two kinds of poisoning attacks, such as label-flipping and model update poisoning attacks, to understand the behavior of poisoning attacks on federated learning-based IDS.

2.3.1. Label-flipping attack

In the label-flipping attack, the attackers manipulate the label of the local dataset to inject falsified local model updates into the server aggregator. The attacker flips the selected source class of the training data to the target class without modifying the features. In the federated learning-based IDS system for anomaly detection, depending on the attacker's goal, the label-flipping can be done in three ways as follows.

1. *Flip benign as the attack label*: The goal is to always predict the traffic as an attack such that the model will have a high false positive rate (FPR) (i.e., false alarms)
2. *Flip attack as the benign label*: As the goal is to always predict the traffic as benign, the model will not correctly detect the attack traffic making the true negative rate (TNR) close to zero.
3. *Flip both labels to each other*: The goal is to make the model's accuracy close to zero.

2.3.2. Model poisoning attack

The model poisoning attacks studied in this paper modify the parameters of the local model to corrupt the global model. Two types of poisoning are observed, the model update scaling and same global model attacks.

1. *Model update scaling attack*: Malicious clients poison the model by multiplying the local model parameters with the negative scaling factor to corrupt the local model such that the gradient of the poisoned model will be in the opposite direction as that of the benign model. This attack is easy to perform and does not require prior knowledge of client data.
2. *Same global model attack*: Similar to the previous attack, this attack does not train the local model at all but replaces the global model parameters with the same number for all poisoned clients.

2.4. Defense

A variety of robust server aggregation algorithms have been proposed, including coordinate-wise median [10], coordinate-wise trimmed mean [10], and multi-Krum [11], to defend against the poisoning attacks in federated learning. These are deployed on the server in the place of FedAvg and are meant to reduce the negative impact of the poisoning attacks. In addition to these robust aggregators, there is also a method of detecting the poisoned clients before starting the global model aggregation process. The intuition behind this is that as the objective of the poisoned clients is different from the non-poisoned client, their gradients should be close to each other, but far away from the rest of non-poisoned clients. Both techniques provide protection against poisoning attacks to some extent. In this study, we compare the following defense methods against our pFL-IDS.

Coordinate-wise median: It sorts the i^{th} parameters of n local models and takes the median of the i^{th} parameters, which is $w^i = median\left\{w_n^i : n\epsilon N\right\}$. When n is an even number, the mean of the two middle values is the median value. When n is odd, the median is the middle parameter.

Coordinate-wise trimmed mean: Similar to the coordinate-wise median, it sorts the i^{th} parameters of n local models and removes the smallest and largest parameters β before the computation of the mean of the rest of parameters $n - 2\beta$.

Multi-Krum: It is a variant of Krum [11], which computes the Euclidean distance between model parameters and averages the top $k = n(totalclients) - f(poisonedclient) - 2$ nearest neighbors to obtain the global model. If $k = 1$, multi-Krum becomes Krum. If $k = n$, it becomes the same as FedAvg. The multi-Krum poorly performs on non-IID data since it is designed to work with IID data.

Poisoned Client Detector: The goal is to detect the poisoned clients and limit their participation in the global model aggregation such that the attacker's goals cannot be achieved. The previous work by Jebreel et al. [13] detects the poisoned client by calculating the cosine similarity of the dimensional-reduced last-layer gradients of local models from the non-poisoned centroid. They assumed that the number of poisoned clients is not larger than 20% so that the normal centroid is always located in the range of the non-poisoned clients, and hence, the poisoned clients can be efficiently identified and removed. As a result, the model performance may be degraded if the poisoned client is larger than the assumed 20%. Moreover, even though the authors have experimented with non-IID data distribution, their proposal did not investigate how to deal with the non-IID data without compromising performance. Compared to [13], our pFL-IDS can defend against poisoned clients up to 30% with a low attack success rate. Moreover, our model is designed to handle performance degradation caused by non-IID clients. The detail of our server-side poisoned client detector is presented in Section 3.3.

3. Proposal

In this section, we provide a detailed explanation of the proposed pFL-IDS, which is a novel personalized federated learning approach for the intrusion detection system that is designed to work with non-IID IoT data. Considering the vulnerabilities of federated learning against poisoning attacks, we propose a poisoned client detector on the server before proceeding wtih the usual global model aggregation step of federated learning. Fig. 2 shows the architecture of our pFL-IDS. The federated learning process is as follows.

1. *Global model initialization*: At the start of the communication round, a global intrusion model is initialized at the server and is sent to the clients.
2. *Local training*: Each client trains the global model with private data to produce the local model.
3. *Server Aggregation*: The clients upload their local models to the server, which updates the global model parameters with the aggregation of the client model parameters. The aggregator FedAvg [5] is a simple weighted average of the client's model parameters as defined by $w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^t$. Our pFL-IDS has an additional step before the global model aggregation to mitigate the negative impact of the poisoning attacks on federated learning. We identify the possible poisoned clients and prohibit them from joining the global aggregation process. We proceed with the global model aggregation only after the detection and removal of the poisoned client.
4. *Model transmission*: The updated global model is sent back to the clients for further training.

Steps 2 to 4 is repeated for multiple communication rounds until a superior global intrusion detection model is achieved.

3.1. Convolutional neural network

The remarkable performance and strong feature extraction ability of the two-dimensional CNN have been demonstrated in image classification and computer vision areas. Meanwhile, the one-dimensional neural network (1D-CNN) is designed to work with sequential data, such as audio signals, text, or time-series data. We use 1D-CNN for the client's intrusion detection model training due to its good performance on time series data. Fig. 2 shows our 1D-CNN architecture. It consists of three convolution layers with kernel size 1 and stride 1, followed by the ReLU activation layer. The input of the first convolution layer is 115, which is the total number of features of the N-BaIoT dataset. The convolution layers have 256, 128, and 64 kernel filters respectively. A dropout layer with $p = 0.2$ is added to prevent overfitting. After the convolution layers map the input to the feature embedding, two fully connected layers are stacked on top to output the correct predictions for the client intrusion detection model.

3.2. Local training: A personalized local model

As the federated learning process involves aggregating multiple local models to produce a single global model, if the imbalanced data distribution is observed among clients, performance degradation happens in FedAvg aggregation. Since the global model is the weighted average of the client's local models, the aggregated global model moves toward the average of the clients' optima. When the clients have IID data, the global model w^{t+1} is closer to the true global optimum w^* as it is the average of the client optimum w_1^* and w_2^*. Therefore, the global optimum w^* is located equally from the local optima, and the client drift did not happen. However, if the clients have non-IID data, the global model w^{t+1} cannot be equally located from both local models since imbalanced data causes client drift when local models are updated (the green line in Fig. 1 indicates the degree of the client drift). Therefore, the resulting global optimum w^* is shifted from the true global optimum, which causes global model divergence and performance degradation. For example, in Fig. 1(b), w^* is closer to client1 optimum w_1^* instead of locating at the true global optimum. Fig. 1 illustrates the global optimum drifting phenomenon in FedAvg for IID and non-IID data.

The common approaches for learning class-imbalanced data in centralized machine learning include re-weighting [23–25], which re-weights each class by applying a factor calculated from the number of samples belonging to each class to make a customized loss function, and re-sampling [26,27], which resamples the samples in a mini-batch stochastic gradient descent. In this study, according to the idea proposed by [24], we apply the logit adjustment loss function in a mini-batch to balance the client data distribution during local model training. Logit adjustment loss is a modified version of cross-entropy loss that manipulates the predicted logits to calibrate the unbalanced data quantity of different classes in the logit space. In federated learning, non-IID clients can degrade the performance of the global model and logit adjustment loss applied in this work is meant to alleviate those adverse effects by making all clients to learn every class accurately. This results in more consistent local models among clients, and the generated global model is more robust to the model divergence and performance degradation. We regard this process as a local model personalization in federated learning.

Personalized federated learning has been introduced recently to learn a personalized model tailored to each client's data. Extensive efforts have been made to realize a personalize approach for non-IID data in federated learning, including techniques such as multi-task learning, model clustering, model parameter decoupling, and knowledge distillation [17,28]. [29–32] used the model decoupling approach to learn the personalized model. In addition, our approach deploys a similar local model decoupling strategy for personalized federated learning. Decoupling means the neural network model is decoupled into two modules: feature extractor (body) and classifier (head). Our decoupling model includes one feature extractor and two classifiers, which are optimized with two different loss functions. This type of model decoupling [32] is proven to be effective for both IID and non-IID data since it can optimize both objectives with two different loss functions.

In FedAvg [5], the entire network is optimized by the cross-entropy loss function given by

$$\ell_{CE}(y, f(x)) = -log\frac{exp(f_y(x))}{\sum_{y' \epsilon \mathbb{C}} exp(f_{y'}(x))}, \tag{1}$$

where $\mathbb{C}$ is the label space and $f_{y'}(x)$ is the output logits for class y'. However, we introduce the additional loss function called mini-batch logit adjustment loss besides the cross-entropy loss to optimize each client's local objective and learn the personalized model for each client, which is defined by

$$\ell_{LA}(y, f(x)) = -log\frac{exp(f_y(x) + \tau.log\alpha_y)}{\sum_{y' \epsilon \mathbb{C}} exp(f_{y'}(x) + \tau.log\alpha_{y'})}, \tag{2}$$

where, τ is the temperature scaling parameter to re-weight the logit adjustment loss. For simplicity, we set τ as 1. The model learns the feature embedding, followed by the first classifier (head), which is optimized with the logit adjustment loss. We calculate the logit-adjusted loss in a mini-batch since it is proven to be effective in handling imbalanced data [33]. After optimizing the first classifier, the second and first classifiers are optimized again with the cross-entropy loss. With these two loss functions, each client learns the local model based on the importance of the underlying class distribution, making the global model more resilient to the non-IID distribution. Our pFL-IDS optimizes the local model with the loss function $\ell_{LA} + \ell_{CE}$.

3.3. Server-side poisoned client detector

The poisoning attacks aim to reduce the performance of the intrusion detection model such that the model cannot protect the ICT system. It is important to mitigate the influence of poisoned clients from the global intrusion detection model. We proposed a poisoned client detector at the server to identify potential poisoned clients and limit their participation in the global intrusion detection model aggregation. The detector first analyzes the last layer of all client's models to determine the model updates that are poisoned. According to [12,13], the last-layer parameters of the poisoned model behave differently from the non-poisoned model. Therefore, comparing the model's last layer provides sufficient information to distinguish the poisoned models.

Algorithm 1: Personalized federated learning-based IDS with poisoned client detection

```
**Input**: Local dataset $\{D_1, D_2, \ldots, D_k\}$,
        Clients $k\epsilon K_t = \{1, 2, \ldots, k\}$,
        Initial global model $w^{t-1}$ $(w^0)$
**Output**: Global model

**Server executes**:
  for each round $t = 1, 2, \ldots, T$ do
    for every client $k\epsilon K_t$ do
      $w_k^t$ ←Client Update $(k, w^{t-1})$    #local model
  #compute the weighted similarity value for clients
  $\{\delta_k | k\epsilon K_t\}$ ← PoisonDetector $(w_k^t | k\epsilon K_t, w^{t-1})$
  #reweight and combine the client models
  $w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^t * \delta_k$    #global model

**Client Update** $(k, w)$:
  $B \leftarrow$ split local dataset into batches of size $| B |$
  for local epoch $i$ from 1 to $E$ do
    for batch $b\epsilon B$ do
      $w \leftarrow w - \eta\Delta\ell(w; b)$
  return $w$ to server

**PoisonDetector** $(w_k^t | k\epsilon K_t, w^{t-1})$:
  $w_{pre}^t \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^t$    #pre-computed global model
  Let $\Delta w_k^t, \Delta w^{t-1}$, and $\Delta w_{pre}^t$ be the parameters of
  the last layer of $\{w_k^t | k\epsilon K_t\}, w^{t-1}$, and $w_{pre}^t$, respectively
   #cosine similarity of pre-computed global and local model
  $\{\bar{\delta_1}, \bar{\delta_2}, \ldots, \bar{\delta_k}\} \epsilon\delta_k^1 \leftarrow \cos\left(\Delta w_{pre}^t, \Delta w^{t-1} - \Delta w_k^t\right)$
   # identify the normal updates
  $\{\Delta w_{nc}^h\}_{h=1}^{H} \leftarrow \delta_k^1 < 0$
   #calculate cosine similarity
  $\{\delta_1^*, \delta_2^*, \ldots, \delta_k^*\} \epsilon\delta_k^2 \leftarrow cos\left(median\left(\{\Delta\hat{w}_{nc}^h\}_{h=1}^{H}\right), \Delta\hat{w}^{t-1} - \Delta\hat{w}_k^t\right)$
   #combine two similarity vectors
  $\delta_k = -\delta_k^2 + \delta_k^1$
  $\delta_k = 0$ if $\delta_k < 0$    # normalize the similarity score to [0,1]
  $\delta_k = 1$ , otherwise
   return $\{\delta_k | k\epsilon K_t\}$
```

Since we aim to restrict the poisoned clients from the global model aggregation, we need to identify them and limit their influence on the global model. Therefore, we design the two-step client similarity alignment, followed by reweighting, to detect the poisoned client. The first similarity computation predicts the potentially normal clients. Then, the normal centroid of the client model is computed from that prediction, and the cosine similarity between the client models from the forecasted normal centroid is calculated again. Next, these two similarity scores are aligned and normalized into the range of [0,1]. Finally, the client models are reweighted by the normalized scores during the global model aggregation. First of all, as soon as clients upload the local models, the poison detector at server calculates the pre-computed global model w_{pre}^t, which is the weighted average of all local models as

$$w_{pre}^t = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t, k\epsilon K_t, \tag{3}$$

where, w_k^t is the local model of client k and K_t is the total number of clients. After that, we calculate the cosine similarity between the pre-global model and local models as

$$\{\bar{\delta_1}, \bar{\delta_2}, \ldots, \bar{\delta_k}\} = cos\left(\Delta w_{pre}^t, \Delta w^{t-1} - \Delta w_k^t\right), k\epsilon K_t, \tag{4}$$

where Δ indicates the last layer parameters of the neural networks model and w^{t-1} is the previous round global model. The intuition is that the poisoned model should be similar to the pre-global model if the poisoning attempt is successful, as all poisoned clients have the same objective throughout the learning process. Since the cosine similarity value is in the range of [−1,1], according to the above assumption, the similarity values of poisoned clients will be closer to 1 than normal clients. After putting a threshold on the similarity values, we separate the clients into poisoned and normal groups. If the client similarity value is less than 0, we define that client as normal, given by

$$\{w_{nc}^h\}_{h=1}^{H} = \delta_k^1 < 0, \tag{5}$$

where $\delta_k^1 \epsilon \{\bar{\delta_1}, \bar{\delta_2}, \ldots, \bar{\delta_k}\}$ and H is the number of normal clients. $median\left(\{\Delta\hat{w}_{nc}^h\}_{h=1}^{H}\right)$ is the dimension-reduced centroid of normal clients. After that, the angular deviations of local models from the normal centroid are computed again by

$$\{\delta_1^*, \delta_2^*, \ldots, \delta_k^*\} = cos\left(median\left(\{\Delta\hat{w}_{nc}^h\}_{h=1}^{H}\right), \Delta\hat{w}^{t-1} - \Delta\hat{w}_k^t\right), \tag{6}$$

where $\delta_k^2 \epsilon \{\delta_1^*, \delta_2^*, \ldots, \delta_k^*\}$ and $\hat{H}at$ sign refers to the dimension-reduced parameters. The normal client will have a smaller angular deviation from the centroid than the poisoned client.

Table 2
Statistics of the mini-N-BaIoT dataset.

Type		Number of samples
Mirai	Scan	7,000
	UDP	7,000
	UDPplain	7,000
	Syn	7,000
	Ack	7,000
BASHLITE	Scan	9,000
	Junk	9,000
	UDP	9,000
	TCP	9,000
	Combo	9,000
Benign		90,000

Table 3
Hyperparameters.

Neural network parameters	
Local model	1DCNN
Number of conv layer	3
Kernel filters in each conv layer	256, 128, and 64
Number of fully connected layer	2
Number of units in each FC layer	32, 2
Activation function	ReLU
Dropout	0.2
Optimizer	SGD
Learning rate	0.001
Momentum	0.9
Batch size	64
Federated learning parameters	
Number of clients	20
Local training epoch	4
Communication round	50

We aligned two similarities as $\delta_k = -\delta_k^1 + \delta_k^2$ and normalized it into the range of [0,1] as follows:

$$\begin{aligned} \delta_k &= 0, \quad \text{if } \delta_k < 0, \\ \delta_k &= 1, \quad \text{otherwise.} \end{aligned} \tag{7}$$

Finally, the local models are reweighted by the normalized similarity scores, and the global model for the next round is computed by

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t * \delta_k, k\epsilon K_t. \tag{8}$$

The detail procedure of our proposed model is shown in Algorithm 1.

4. Evaluation

We evaluated the performance of our pFL-IDS with the N-BaIoT dataset and compared it with several state-of-the-art approaches to investigate the effectiveness of the proposed method. The detail of the dataset, experiment setting, and behavior of poisoning attacks are provided in this section.

Experimental environment: We evaluated the proposed model from the client side. All the experiments are conducted with the PyTorch framework and executed on the PC with Intel Core i7-10750H CPU @ 2.60 GHz, 64 GB RAM.

Dataset: The experiment was performed on the publicly available N-BaIoT dataset [18], which is a collection of network traffic from nine commercial IoT devices infected with Mirai and BASHLITE. The dataset has more than 70 million traffic samples, each with 115 features. The dataset has two traffic categories for binary classification (i.e., benign and attack) and 10 sub-categories of attack carried by Mirai and BASHLITE. Since the original dataset has millions of data records, training such a large dataset requires expensive computational resources and higher time complexity. Therefore, we construct a small dataset from the 11 classes of the 9 IoT devices from the original dataset. We take 1000 traffic records per attack class from each device to build the mini-size N-BaIoT dataset. For example, for the benign class, we select 10,000 traffic data in each device. After combining all the selected traffic records, a mini dataset of 170,000 traffic samples is obtained, as shown in Table 2. We divided the mini-N-BaIoT in the ratio of 70 to 30 for the train and test datasets. The training dataset was distributed among the number of participating clients, and the test dataset was used to measure the performance of the federated intrusion detection model. In the practical federated learning scheme, the number of clients is large, and to simulate this situation, we further partition the training dataset according to the total number of clients. We set the total number of clients used in the experiment as 20. This way of dividing a dataset to mimic the multiple clients has been a well-known approach in the federated learning literature. The hyperparameters used in the experiment are shown in Table 3.

Data distribution: The training dataset is further partitioned for N clients to mimic the setting of federated learning. Depending on the data distribution mode (i.e., IID or non-IID), we sampled the client's local dataset from the training dataset. In the experiment, the total number of clients is 20, and it is assumed that all clients train locally at every round. The following data distribution scenarios are experimented as follows:

1. Non-IID: Dirichlet distribution with $\eta = 0.1$
2. Non-IID: Half of the clients have only benign samples and the rest have only attack samples
3. IID: Benign 50%, Attack 50%

In the case of IID, the number of benign and attack samples in the dataset is the same. Non-IID scenario 1 is simulated by the Dirichlet distribution with $\eta = 0.1$. In non-IID scenario 2, we omitted the attack samples for half of the clients, since the ratio of benign traffic is much larger in the real world and not all IoT devices are compromised. For scenarios 1 and 3, the total number of samples per client is fixed at 1000. For scenario 2, 500 samples are apportioned to the client that has only benign samples, and 1000 samples are employed for the rest of the clients.

Poisoning Attack Setting: The behaviors of three label flipping attacks and two model update poisoning attacks are examined. We assumed that the malicious client is at 30% of the total clients to compare pFL-IDS with other baseline methods. Therefore, in the experiment settings of the trimmed mean and multi-Krum, we set the trim ratio β as 0.3 and the poisoned client f as 5. For model update scaling attacks, the gradients are multiplied by a factor of −1. Similarly, all gradients of the poisoned models are replaced by a factor of −3 to simulate the same global model attack.

Evaluation metrics: We computed the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). FP indicates the number of normal data incorrectly classified as an attack, while FN represents the number of attack samples incorrectly classified as normal. TN and TP denote the number of normal or attack data that were accurately classified. Based on these values, the evaluation metrics, such as precision, recall, and F1 are computed. We also calculated the attack success rate (ASR) to measure the accomplishment of the poisoned clients based on their attack objective. If the attack is successful, ASR is 1 (i.e., the global intrusion detection model outputs the target class as desired by the poisoned client instead of the source class); otherwise, it is 0. ASR for label-flipping attacks and model update poisoning attacks are computed as

$$\mathrm{ASR}_{\mathrm{lf}} = \frac{\mathrm{Test}_{\text{target class}}}{\mathrm{Test}_{\text{source class}}}, \quad \mathrm{ASR}_{\mathrm{mp}} = \frac{\mathrm{FP}+\mathrm{FN}}{\text{Total test samples}}. \tag{9}$$

Keeping ASR as low as possible while maintaining good performance for the intrusion detection system is necessary for it to become an effective defense mechanism against poisoning attacks.

Table 4
Evaluation of Non-IID data (scenario 1). The attack ratio is 30%. The best results are bolded.

Poisoning attack		Metrics	FedAvg	Median	Trimmed mean	Multi-Krum	FL-Defender	Ours
None		Accuracy	**0.9909**	0.9794	0.9859	0.9885	0.9823	0.9822
		F1	**0.9904**	0.9778	0.9849	0.9879	0.9810	0.9809
		ASR	**–**	–	–	–	–	–
Label-flipping	Benign	Accuracy	**0.9857**	0.4706	0.4706	0.4706	0.4706	0.9511
		F1	**0.9850**	0.6400	0.6400	0.6400	0.6400	0.9457
		ASR	0.0267	1.0000	1.0000	1.0000	1.0000	**0.0066**
	Attack	Accuracy	0.9281	0.5294	0.9443	0.7479	0.5294	**0.9613**
		F1	0.9179	0.0000	0.9375	0.6359	0.0000	**0.9575**
		ASR	0.1460	1.0000	0.1123	0.5321	1.0000	**0.0741**
	Both	Accuracy	0.8378	0.4706	**0.9723**	0.4706	0.4706	0.9598
		F1	0.8316	0.6400	**0.9700**	0.6400	0.6400	0.9558
		ASR	0.1622	0.5294	**0.0277**	0.5294	0.5294	0.0402
Model Scaling Attack		Accuracy	0.4706	0.4706	0.4706	0.4706	**0.9776**	0.9642
		F1	0.6400	0.6400	0.6400	0.6400	**0.9759**	0.9607
		ASR	0.5294	0.5294	0.5294	0.5294	**0.0224**	0.0358
Same Model attack		Accuracy	0.4706	0.7367	0.9537	**0.9902**	0.5294	0.9651
		F1	0.6400	0.6117	0.9531	**0.9896**	0.0000	0.9618
		ASR	0.5294	0.2633	0.0463	**0.0098**	0.4706	0.0349

Table 5
Evaluation of Non-IID data (scenario 2). The attack ratio is 30%. The best results are bolded.

Poisoning attack		Metrics	FedAvg	Median	Trimmed mean	Multi-Krum	FL-Defender	Ours
None		Accuracy	0.9786	0.7445	0.7807	0.7782	0.5294	**0.9871**
		F1	0.9771	0.6275	0.6979	0.6934	0.0000	**0.9864**
		ASR	–	–	–	–	–	**–**
Label-flipping	Benign	Accuracy	0.9567	0.8544	0.9658	0.8646	0.8190	**0.9891**
		F1	0.9560	0.8184	0.9628	0.8352	0.7627	**0.9885**
		ASR	0.0811	0.0060	0.0110	0.0146	**0.0024**	0.0131
	Attack	Accuracy	0.7610	0.5294	0.5294	0.6369	0.5294	**0.9885**
		F1	0.6609	0.0000	0.0000	0.3749	0.0000	**0.9878**
		ASR	0.5051	1.0000	1.0000	0.7686	1.0000	**0.0098**
	Both	Accuracy	0.7743	0.6405	0.7075	0.6810	0.9721	**0.9893**
		F1	0.6883	0.3850	0.5510	0.4905	0.9697	**0.9886**
		ASR	0.2257	0.3595	0.2925	0.3190	0.0279	**0.0107**
Model Scaling Attack		Accuracy	0.6715	0.5294	0.5297	0.7046	0.9549	**0.9915**
		F1	0.4667	0.0000	0.0011	0.5454	0.9501	**0.9910**
		ASR	0.3285	0.4706	0.4703	0.2954	0.0451	**0.0085**
Same Model attack		Accuracy	0.5294	0.6639	0.8249	0.8605	0.5294	**0.9842**
		F1	0.0000	0.4465	0.7724	0.8284	0.0000	**0.9833**
		ASR	0.4706	0.3361	0.1751	0.1395	0.4706	**0.0158**

4.1. Results

To demonstrate the effectiveness of our proposal against the poisoning attacks on imbalanced data, the experiment was performed on three data distribution scenarios, and the results are presented in Tables 4, 5, and 6. We compared our pFL-IDS with FedAvg, coordinate-wise median, trimmed mean, multi-Krum, and FL-Detector. Moreover, the behavior of different kinds of poisoning attacks on non-IID data was also investigated. When the data are IID, most models performed well except the FedAvg for same model poisoning attack. Multi-Krum, FL-Defender, and our pFL-IDS can defend against all poisoning attacks and achieve comparable performance in all evaluation metrics. The experimental results for IID data are presented in Table 6.

The behavior of poisoning attacks on non-IID data (scenario 1)

We sample the non-IID dataset using Dirichlet distribution with $\eta = 0.1$. We choose $\eta = 0.1$ to simulate extremely imbalanced data. The Dirichlet distribution is commonly used in previous studies [12,13] to sample the non-IID distribution for federated learning. When there is no attack, all models achieved a remarkable accuracy and F1 score. However, in comparison with the IID data, the accuracy dropped about 1% in pFL-IDS . As shown in Table 4, the proposed model has an excellent performance for all poisoning attacks and has a low ASR value.The trimmed mean and FedAvg is the second best models, which can defend 3 out of 5 poisoning attacks. The FL-Defender, multi-Krum, trimmed mean and coordinate-wise median cannot absolutely defend against the benign label-flipping, with an ASR of 1 indicating that the attacker's goal is 100% achieved. Similarly, 100% attack success rate is also observed in the attack label-flipping against the FL-Defender and coordinate-wise median. For model scaling attack, FedAvg, median, trimmed mean and multi-Krum have an F1 score of 0.64 with a 0.4706 accuracy. This situation means that the intrusion detection model cannot classify the benign label at all (i.e., false alarms will continuously raise). The similar false alarm situation is also discovered in the same model poisoning attack against FedAvg. Meanwhile, our pFL-IDS can defend against both model poisoning attacks with 2-3% accuracy difference from the situation where no poisoning attacks are occurred.

The behavior of poisoning attacks on non-IID data (scenario 2)

In this scenario, the labeled attack samples were absent in half of the clients. Considering not all IoT devices are compromised and the labeled attack samples are scarce, the assumption of not having the attack samples in all client's devices is aligned with reality. The empirical results in Table 5 indicated that our model has superior results compared to all baseline methods. Even when there are no poisoning attacks, only FedAvg and our model perform well, while the accuracy of the three robust aggregators drops substantially. FL-Defender has the worst accuracy and F1, with the F1 score being close to zero (i.e., the model cannot classify almost all of the attack

Table 6
Evaluation of IID data. The attack ratio is 30%. The best results are bolded.

Poisoning attack		Metrics	FedAvg	Median	Trimmed Mean	Multi-krum	FL-Defender	Ours
None		Accuracy	**0.9904**	0.9815	0.9870	0.9896	0.9847	0.9894
		F1	**0.9898**	0.9802	0.9861	0.9890	0.9838	0.9888
		ASR	**–**	–	–	–	–	–
Label-flipping	Benign	Accuracy	0.9892	0.9895	0.9852	0.9886	0.9881	**0.9904**
		F1	0.9886	0.9888	0.9844	0.9879	0.9874	**0.9899**
		ASR	0.0203	0.0105	0.0201	**0.0094**	0.0097	0.0111
	Attack	Accuracy	0.9329	0.8515	0.9685	0.9876	**0.9893**	0.9890
		F1	0.9235	0.8135	0.9657	0.9869	**0.9886**	0.9883
		ASR	0.1385	0.3120	0.0600	0.0113	0.0145	**0.0099**
	Both	Accuracy	0.9271	0.9802	0.9786	**0.9887**	0.9886	0.9877
		F1	0.9164	0.9787	0.9769	**0.9880**	0.9878	0.9871
		ASR	0.0729	0.0198	0.0214	**0.0113**	0.0114	0.0123
Model Scaling Attack		Accuracy	0.9453	0.9628	0.9662	0.9877	0.9876	**0.9885**
		F1	0.9386	0.9592	0.9630	0.9870	0.9868	**0.9879**
		ASR	0.0547	0.0372	0.0338	0.0123	0.0124	**0.0115**
Same Model attack		Accuracy	0.4706	0.9831	0.9879	0.9894	0.9894	**0.9897**
		F1	0.6400	0.9819	0.9871	0.9887	0.9887	**0.9891**
		ASR	0.5294	0.0169	0.0121	0.0106	0.0106	**0.0103**

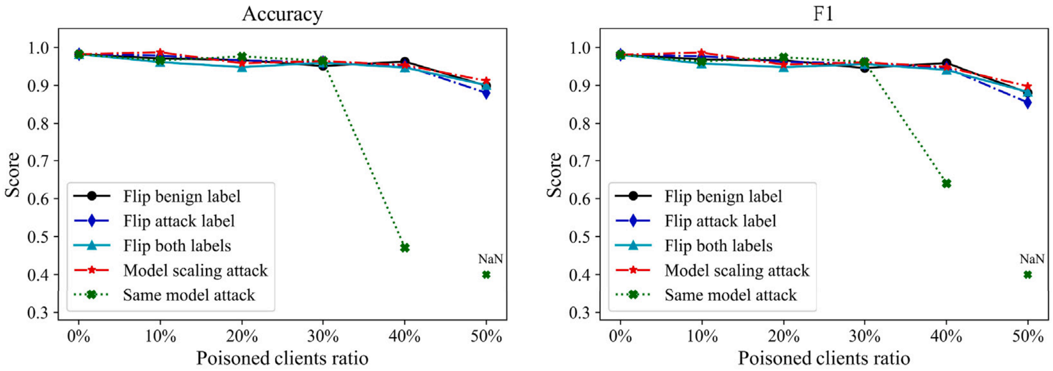


Fig. 3. The accuracy and F1 of pFL-IDS with different ratios of malicious client.

samples). This phenomenon is due to having only benign samples in some clients, making the FL-Defender wrongly identify the clients with both samples as anomalous, consequently reducing their influence on the global model. Therefore, the resulting global model has little or no knowledge of the attack samples and cannot classify the attack samples. When flipping the label of benign samples as an attack, all baseline models can defend the attack to a certain extent. When attack labels are flipped as benign, besides FedAvg, multi-krum and our model, the rest of the models have an ASR of 1, implying the goal of the poisoned clients is successfully achieved. In model update scaling attacks, the attack success rate of all baseline models except FL-Defender is pretty high, especially 47% ASR and an F1 score close to zero are observed in median and trimmed mean models. This means that the model classified all testing samples as benign, and thus, it cannot detect the intrusion attack samples at all. This situation is the worst as the objective of the intrusion detection system is to classify the attack samples as much as possible while keeping a low false positive rate. For the same model poisoning attack, the aforementioned behavior is observed in FedAvg and FL-Defender, and the rest of the baseline models can protect against poisoning to some extent. However, only our methods can effectively protect against all poisoning attacks with excellent performance while preserving a good accuracy and keeping the ASR as low as possible.

4.2. Discussion

As the robust aggregators (i.e., median, trimmed mean, multi-Krum) are designed to work with the IID data, they cannot defend against poisoning attacks when the data are non-IID. As the trimmed mean and multi-Krum must predefine the ratio of the malicious attackers, performance degradation could happen if the predefined ratio is less than that of the attackers. FL-Defender can correctly detect and migrate the influence of the poisoned clients in the IID mode; however, it cannot protect against all poisoning attacks in non-IID mode, particularly if one class is missing in the client local dataset, as shown in our experimental setting. Due to the client global optima drift phenomenon in non-IID data, the poisoned client detector of FL-Defender may wrongly recognize the benign model as the poisoned model (i.e., caused by client drift). As our pFL-IDS is customized to handle imbalanced data, it works well with non-IID data, and the empirical results are presented in Tables 4, 5, and 6.

Impact of different percentages of poisoned clients: The previous studies for poisoning attacks assumed that the portion of the compromised clients is less than 50%, with the poisoned client range of 10%–25% being the most studied one. Accordingly, our study adopted that assumption (<50%). To examine to which extent our pFL-IDS is resilient to poisoning attacks, we first stress-test with a higher ratio of the poisoned clients, and we found that pFL-IDS can withstand up to 30% of poisoned clients for all poisoning attacks with a low attack success rate as shown in Tables 4, 5, and 6. The other defense methods collapse in some poisoning attacks compared to ours, especially apparent in non-IID cases. Even though it is desirable to implement a robust model which can defend as many poisoned clients as possible, having more than 20%–30% of malicious clients in actual practice is rare. Since our model experiments with a few clients (just 20), 30% malicious clients means compromising only six clients. However, in large-scale IoT scenarios with millions of clients, even if the attacker poisons just a tiny fraction of clients, the attacker needs to compromise many clients. This situation is somewhat impractical and unlikely to happen in the real world. The impact of the different ratios of the poisoned clients for label-flipping and model poisoning attacks is illustrated in Fig. 3. When the poisoning ratio reaches 40%, our model cannot defend against the same model poisoning attack despite being resilient to other poisoning attacks. In principle, our pFL-IDS can protect up to 30% of the poisoning attacks while achieving excellent results for all evaluation metrics.

Support on large-scale IoT devices: Our proposed model is designed for cross-silo federated learning where the client number is usually small, and each client has sufficient computing power. Thus, in the experiment, we set the total number of clients as 20 and allowed every client to participate in one communication round. However, for large-scale IoT devices, this full participation assumption is infeasible as the server needs to communicate with a large number of IoT devices in each round, which can cause significant communication overhead and computation at the server. Especially since the server needs to wait for all clients to finish uploading the local model to generate a global model, if some clients are struggling to complete the model training, the server needs to wait indefinitely. To support large-scale IoT devices with our method, we can apply the client selection procedure to choose a fraction of the total clients to train in one round. This kind of client selection is commonly utilized in the existing literature. Moreover, to deal with the clients that cannot upload the models in the specified period, we can consider dropping those clients to avoid further delay. Currently, both issues (client selection and handling of the straggling clients) are being vigorously studied in the research community, and we also hope to contribute improvement in that direction in our future work.

Further discussion on the recent advancement in the Neural Networks: A suitable neural network architecture for a specific learning task is vital in building a good model. Our proposed method uses 1DCNN as a local model due to its superior performance in the analysis of the time series domain, especially on one-dimensional data. Although our current model can detect the IDS tasks with reasonable accuracy, we can further explore more neural networks to maximize the prediction performance. For example, a recent study [34] proposed a deep spatiotemporal model by combining two-dimensional CNN and LSTM to associate data's spatial and temporal dependencies. In this model, CNN extracts the spatial features, and LSTM finds the temporal relationships to improve the model performance. Likewise, [35] proposed a differentially private tensor-based recurrent neural network to protect against privacy leakage in IoT systems and demonstrated that the proposed approach could preserve the leakage of confidential information in IoT data. Both [34,35] applied their proposed methods in the decentralized learning scenarios, and perhaps, extending those models to the context of federated learning-based IDS can be exciting and meaningful research directions.

5. Conclusion

In this study, we designed a robust federated learning-based pFL-IDS to detect poisoning attacks for non-IID data. The empirical results revealed that the malicious client can degrade the IDS model performance, which is severe if the clients have non-IID data. As real world IoT IDS datasets are undoubtedly imbalanced, we proposed a personalized IDS approach for local model training in addition to the malicious client detector at the server to combat the poisoning attacks and the non-IID data. With the extensive experiment conducted, we demonstrated that pFL-IDS is effective in defending against both poisoning attacks regardless of the data distribution of the client. Moreover, the empirical results demonstrated that pFL-IDS outperforms all baseline methods on non-IID data and prove that it can work well with both IID and non-IID data.

CRediT authorship contribution statement

Thin Tharaphe Thein: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Yoshiaki Shiraishi:** Conceptualization, Methodology, Validation, Writing – review & editing, Supervision. **Masakatu Morii:** Project administration.

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.

Data availability

Data will be made available on request.

Acknowledgments

This research includes results obtained from the "Research and development on new generation cryptography for secure wireless communication services" commissioned by the Ministry of Internal Affairs and Communications of Japan under the "Research and Development for Expansion of Radio Resources (JPJ000254)" project. This work was also supported by JSPS KAKENHI, Japan Grant Number JP21H03444.

References

[1] Future of industry ecosystems: shared data and insights, 2021, https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/. (Last Accessed 18 December 2022).

[2] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2702–2733.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, et al., Understanding the mirai botnet, in: 26th {USENIX} Security Symposium ({USENIX} Security 17), 2017, pp. 1093–1110.

[4] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P.K.R. Maddikunta, T.R. Gadekallu, Federated learning for intrusion detection system: Concepts, challenges and future directions, Comput. Commun. (2022).

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

[6] M.A. Ferrag, O. Friha, L. Maglaras, H. Janicke, L. Shu, Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis, IEEE Access 9 (2021) 138509–138542.

[7] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.

[8] V. Rey, P.M.S. Sánchez, A.H. Celdrán, G. Bovet, Federated learning for malware detection in iot devices, Comput. Netw. 204 (2022) 108693.

[9] Z. Zhang, Y. Zhang, D. Guo, L. Yao, Z. Li, SecFedNIDS: Robust defense for poisoning attack against federated learning-based network intrusion detection system, Future Gener. Comput. Syst. 134 (2022) 154–169.

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

[11] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, Adv. Neural Inf. Process. Syst. 30 (2017).

[12] S. Awan, B. Luo, F. Li, Contra: Defending against poisoning attacks in federated learning, in: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26, Springer, 2021, pp. 455–475.

[13] N.M. Jebreel, J. Domingo-Ferrer, FL-Defender: Combating targeted attacks in federated learning, Knowl.-Based Syst. 260 (2023) 110178.

[14] X. Cao, M. Fang, J. Liu, N.Z. Gong, Fltrust: Byzantine-robust federated learning via trust bootstrapping, 2020, arXiv preprint arXiv:2012.13995.

[15] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, 2012, arXiv preprint arXiv:1206.6389.

[16] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.

[17] A.Z. Tan, H. Yu, L. Cui, Q. Yang, Towards personalized federated learning, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[18] 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 (3) (2018) 12–22.

[19] S.I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, O. Jogunola, Federated deep learning for zero-day botnet attack detection in IoT-edge devices, IEEE Internet Things J. 9 (5) (2021) 3930–3944.

[20] Y. Fan, Y. Li, M. Zhan, H. Cui, Y. Zhang, Iotdefender: A federated transfer learning intrusion detection framework for 5g iot, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), IEEE, 2020, pp. 88–95.

[21] 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 (4) (2021) 2545–2554.

[22] 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.

[23] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

[24] A.K. Menon, S. Jayasumana, A.S. Rawat, H. Jain, A. Veit, S. Kumar, Long-tail learning via logit adjustment, 2020, arXiv preprint arXiv:2007.07314.

[25] J. Ren, C. Yu, X. Ma, H. Zhao, S. Yi, et al., Balanced meta-softmax for long-tailed visual recognition, Adv. Neural Inf. Process. Syst. 33 (2020) 4175–4186.

[26] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[27] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[28] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, Z. Zhang, Fedproc: Prototypical contrastive federated learning on non-iid data, Future Gener. Comput. Syst. (2023).

[29] M.G. Arivazhagan, V. Aggarwal, A.K. Singh, S. Choudhary, Federated learning with personalization layers, 2019, arXiv preprint arXiv:1912.00818.

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

[31] L. Collins, H. Hassani, A. Mokhtari, S. Shakkottai, Exploiting shared representations for personalized federated learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 2089–2099.

[32] H.-Y. Chen, W.-L. Chao, On bridging generic and personalized federated learning for image classification, 2021, arXiv preprint arXiv:2107.00778.

[33] H. Lee, S. Shin, H. Kim, Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning, Adv. Neural Inf. Process. Syst. 34 (2021) 7082–7094.

[34] S. Tsokov, M. Lazarova, A. Aleksieva-Petrova, A hybrid spatiotemporal deep model based on CNN and LSTM for air pollution prediction, Sustainability 14 (9) (2022) 5104.

[35] J. Feng, L.T. Yang, B. Ren, D. Zou, M. Dong, S. Zhang, Tensor recurrent neural network with differential privacy, IEEE Trans. Comput. (2023).

Thin Tharaphe Thein received the B.E. degree from Yangon Technological University, Myanmar in 2019 and M.E. degree from Kobe University, Japan in 2021. She is currently pursuing doctoral degree at Kobe University. Her research focuses on data-driven cybersecurity, machine learning and threat intelligence.

Yoshiaki Shiraishi received his B.E. and M.E. degrees from Ehime University, Japan, and his Ph.D. degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a Lecturer at the Department of Informatics, Kindai University, Japan. From 2006 to 2013 he was an Associate Professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an Associate Professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and machine learning based cyberattack analysis. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IEICE, IPSJ.

Masakatu Morii received his B.E. degree in electrical engineering and his M.E. degree in electronics engineering from Saga University, Saga, Japan, and his D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering, Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronic Engineering, Faculty of Engineering, Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics, computer networks and information security. He is a member of IEEE and a fellow of IEICE.