Federated Learning-Based Intrusion Detection in the Context of IIoT Networks: Poisoning Attack and Defense

Nguyen Chi Vy[1,2], Nguyen Huu Quyen[1,2], Phan The Duy[1,2](✉), and Van-Hau Pham[1,2]

[1] Information Security Laboratory, University of Information Technology, Ho Chi Minh City, Vietnam
{18521681,18521321}@gm.uit.edu.vn, {duypt,haupv}@uit.edu.vn
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. The emerging of Federated Learning (FL) paradigm in training has been drawn much attention from research community because of the demand of privacy preservation in widespread machine learning adoption. This is more serious in the context of industrial Internet of Things (IIoT) with the distributed data resources and the sensitive local data in each data owner. FL in IIoT context can help to ensure the sensitive data from being exploited by adversaries while facilitating the acceptable performance by aggregating additional knowledge from distributed collaborators. Sharing the similar trend, Intrusion detection system (IDS) leveraging the FL approach can encourage the cooperation in building an efficient privacy-preserving solution among multiple participants owning the sensitive network data. But a rogue collaborator can manipulate the local dataset and send malicious updates to the model aggregation, aiming to reduce the global model's prediction accuracy rate. The reason for this case is that the collaborator is a compromised participant, or due to the weak defenses of the local training device. This paper introduces a FL-based IDS, named Fed-IDS which facilitates collaborative training between many organizations to enhance their robustness against diverse and unknown attacks in the context of IIoT. Next, we perform the poisoning attack against such an IDS, including label-flipping strategy and Generative Adversarial Networks (GANs). Then, a validation approach is utilized as a countermeasure of rejecting the malicious updates to protect the global model from poisoning attacks. The experiments conducted on Kitsune, a real-world attack dataset, demonstrate the high effectiveness of the validation function in Fed-IDS framework against data poisoning.

Keywords: Intrusion detection · IDS · Federated learning · Poisoning attack · GAN · Generative Adversarial Networks

1 Introduction

Recently, the rapid advances in the communication and internet fields have resulted in a huge increase in the network size and the corresponding data.

M. Yang et al. (Eds.): NSS 2021, LNCS 13041, pp. 131–147, 2021.
https://doi.org/10.1007/978-3-030-92708-0_8

Especially, heterogeneous Internet of Things (IoT) devices in the industrial context have led to a rapid growth in data volume generated during its operation. It leads to more challenges on network orchestration and security risk through data communication, and sharing [10]. Moreover, many IoT weakness and vulnerabilities has been exploiting by a diversity of malwares, adversaries, leading to serious challenges for device and network security in accurately detecting and preventing cyber threats [16]. To this end, the intrusion detection system (IDS) is a crucial component to prevent attack steps in the cyber kill chain, needing to be updated frequently to recognize malicious behaviors in the system or the network. To achieve better performance on unknown malicious traffic, such a system leverages the capability of machine learning (ML) for detecting abnormally action [20]. The benefits from ML in enhancing scalability and improving detection ability are proved in many studies in recent years [7,14]. In fact, these ML models produce results with high accuracy, thanks to the rapid increase in amount of collected network data and the sharing of known indicator of diverse sophisticated threats [8,9]. But the growth of network data is also a major challenge in implementing the centralization of data for training in the large-scale IoT network. In fact, centralized training strategy is often expensive cost for server and system capable of working and training with the huge amounts of data and keeping raw data secure are also a big problem [7]. Furthermore, data privacy is one of the most important problems in all ML applications, especially network traffic data [19]. At the moment, the lack of protection for sensitive and private data has become one of the main issues needing to be resolved while adopting ML in industrial Internet of Things (IIoTs) context [23].

To solve this concern, a novel method of training ML models, named Federated Learning (FL) is advocated in building a better privacy-preserving solution for real-life artificial intelligence adoption [2]. The main target of FL is reducing the pressure on the server, by creating a righteous model on the participant device holding the amount of data from global model fed from the aggregation server [24]. This method not only solves the problem of centralization of data, but also helps ensure data privacy since the training data do not leave the safe perimeter of organizations, or users. In particular, the new types of network attacks and vulnerabilities are increasingly diverse and more difficult to defend. The emergence of FL algorithms marks a step forward in encouraging the collaboration among many network attack data owners to develop an up-to-date detector from others. FL enables local agents to collaboratively learn a shared prediction model while keeping all training data on its premises, decoupling the ability to applying ML from the need to store the training dataset in the cloud or a centralized repository. It helps to release concerns and threats to individual and organizational data privacy. Beside the outstanding benefits that FL method brings, there are still some disadvantages, threats, and vulnerabilities [12,15]. Therein, the training data is heterogeneous and not well validated on different devices can greatly affect the global ML model. Additionally, poisoning attacks can reduce the accuracy performance of a global model if it lacks a verification mechanism of local updates from distributed clients [21].

In general, many studies has been focused on the FL adoption in intrusion detection schemes with high-rate detection accuracy. For instances, Thien Nguyen et al. [18] proposed a FL-based IDS in which a security gateway of each network participates in the collaborative training to automatically detecting threatens to IoT devices. Likewise, Liang Zhao et al. [11] developed FL approach for training DL-based model for IDS in industrial cyber-physical systems (CPSs). Mohamed et al. also proposed Fed-TH [1], a federated deep learning for hunting cyber threats in industrial CPSs. However, recent studies found that the FL-based IDS framework is also suffered poisoning attack. By experiments, they showed that the functionalities of the global model could be damaged through attacker's well-crafted local updates, like the works of Thien et al. [17]. Poisoning attack against FL model can be conducted by flipping labels [4] and Generative Adversarial Networks (GANs)-based data generation [25,26]. To address these problems, we first give a comprehensive empirical analysis of poisoning attack by flipping labels and Generative Adversarial Networks (GANs)-based strategies to illustrate the differences in predicting attacks of our FL-based IDS model under adversarial environments. Then, to prevent the client model from being poisoned during the training of each client's model, a validation function using Local Outlier Factor (LOF) is integrated into the model aggregation to compare the current model's performance with the accepted models in previous training rounds to further process. Our main contributions are summarized as follows:

- First, our deep learning model, named Fed-IDS is built relying on the structure modification of DeepFed [11] and validated with FL approach on real-world dataset of network intrusions.
- Second, we figure and compare the attack detection model's performance when updating poisoned data from a rogue training collaborator by both strategies of flipping labels and adding fake data generated by Generative Adversarial Networks (GANs).
- Third, a validation function, which is developed and integrated into FL scheme as a verification process before a client's training model is aggregated into the global IDS model.

The remainder of this paper is organized as follows. In Sect. 2, related works on data poisoning attacks, attack detection system for IIoT context are briefly introduced. Also, we discuss the federated learning approach for preserving sensitive data for training ML models and the vulnerabilities of FL. In Sect. 3, we describe the research methodology adopted in this study. The implementation, metrics, and performance evaluation results are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related Work

In this section, we briefly review deep learning model, the vulnerabilities and model-poisoning data which are associated with federated learning-based intrusion detection methods.

Initially, N.D. Thien et al. [18] proposed DïoT, an autonomous self-learning distributed system for detecting compromised IoT devices. This system was benchmarked on the dataset collected by several smart homes devices like cameras and routers. Beibei Li et al. proposed DeepFed [11], a federated learning scheme for IDS in industrial cyber-physical systems (CPSs), which conducts collaborative training a deep learning model on each security agents. The model is structured with a combination of convolutional neural networks (CNN) and gated recurrent units (GRUs). This approach helps multiple industrial CPSs to build a comprehensive intrusion detection model and ensure data privacy of involved organizations. Besides that, Fed-TH [1] was proposed by leveraging the characteristics of FL to hunting cyber threats against industrial CPSs.

Despite its ability to protect user privacy-sensitive data, FL still has the vulnerabilities that Nader Bouacida et al. [5] classified adversarial attacks into two categories based on their goals: untargeted attack aiming to reduce the global model accuracy, targeted attack aiming to change model's behavior on a specifically targeted subtask while sustaining good overall accuracy. With the rise of federated learning scenario, many works explore how the poisoning data affects the global model through model updates from a rogue client. The main target of poisoning attack aims to degrade the accuracy of the target model by poisoned data. This data can originate from label flipping strategy or injecting fabricated malicious data when performing local training aiming to misclassify prediction label. Jiale Zhang et al. [26] presented a poison data generation method, named Data_Gen, based on the generative adversarial networks (GAN). The authors also proposed a generative poisoning attack model, named PoisonGAN, against the federated learning-based image classifier on MNIST, F-MNIST, and CIFAR-10 datasets.

Recent studies have shown that FL is vulnerable to poisoning attacks that inject a backdoor into the global model. Sebastien et al. [3] have proposed a novel defense secure FL against backdoor attacks and named it Backdoor detection via Feedback-based Federated Learning (BaFFLe). The diverse datasets from the various clients were used to feedback to the FL process for detecting the poisoning updates from participants. Through empirical evaluation using the CIFAR-10 and FMNIST datasets, BaFFLe also gave the accuracy of 100% and false-positive rate below 5% on detecting poisoning attacks. Likewise, Vale Tolpegin et al. [22] performed poisoning attacks on datasets, which lead the FL-based systems have false-positive. The authors also proposed a method to avoid changing the labels in the datasets by validating the machine learning models before updating into the global models.

In summary, there are several works attempting to adopt FL in many applications, whereas a few studies investigate how poisoning attack can affect the global FL-based model and find the measure to defend. Nevertheless, it lacks the comprehensive study on the performance of FL-based IDS against rogue agents that feed the erroneous data to degrade the efficacy of global model. Recently, I. BeiBei et al. [11] proposed a federated deep learning scheme (DeepFed) for intrusion detection in industrial cyber-physical systems. But they did not consider the security of the FL approach, where malicious users can interfere the training process with poisoned data. Also, Zhang et al. [26] presented a generative poisoning attack against FL (PoisonGAN) which exposes 2 attack methods: labels flipping and generating poisoned data. However, they conduct the poisoning attack on image classification without investigating the effectiveness of defense method on their approach against poisoning attacks.

3 Methodology

This section gives the overview of our FL strategy for making a collaboration between different parties to build a more robust IDS without leaking raw data. After that, we introduce the method of performing poison attack against this FL-based IDS and its countermeasure.

3.1 Federated Learning

In this part, we introduce the FL model for IDS which consist of one global server and many collaborating machines, as shown in Fig. 1

- The server is responsible for aggregating the local models from each machine collaborates to build the global model. Then, the updated global model will be transmitted to the collaborators to continue training and improving the global model. Note that, the server is built on a trust infrastructure to provide the reliability for involved networks.
- The collaborators are responsible for locally training on its raw data and sending the trained parameters up to the global server. These parameters are the weight of the collaborator's training model. A collaborator is a training server representing for IDS builder in each network joined in the FL scheme.

And the workflow of this model can be summarized as follows:

- First, the server sends the weight and gradient of a pre-trained model or a generated model randomly to the collaborators.
- Based on the set of parameters received from the server, the collaborating machines proceeds training based on the amount of data which each machine is holding. In this method, D_i is the dataset at the i-th collaborator and w_i is the set of parameters after training.
- The set of weights w_i and size of the dataset is sent to the server when the collaborator finishes the training process.
- And the new model's set weights will be calculated by Formula 1.
- Finally, the server sends the updated model to collaborating machines for continuous use, evaluation, and train.

To establish the aggregate formula that the server does, we assume that there are N machines participating in collaboration during the training. And we only consider the case that the server must receive enough N models to execute

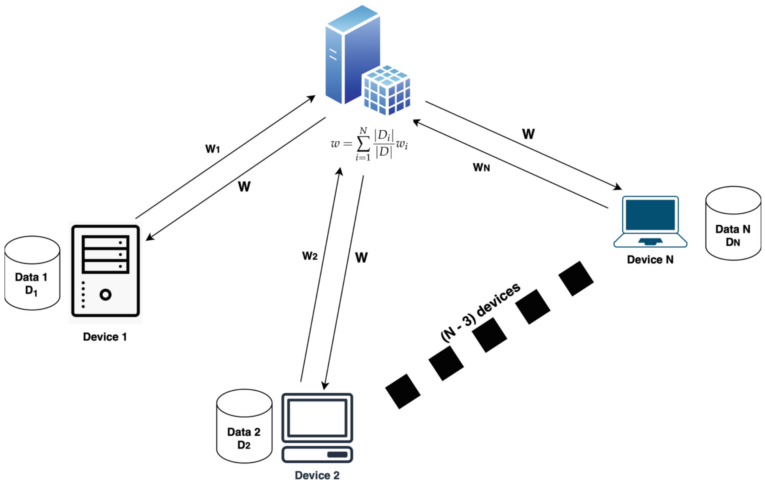

Fig. 1. The architecture of federated learning.

aggregating and updating new model for collaborative machines. Thus, the aggregation formula is calculated as in (1):

$$w = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i \tag{1}$$

In the Formula 1, w is the set of weights in new model in the global server. And $|D|$ which is the total size data of N collaborators is calculated as in (2).

$$|D| = \sum_{i=1}^{N} |D_i| \tag{2}$$

3.2 Training on Collaborative Machines

In this study, our training model, Fed-IDS is built based on DeepFed [11] as shown in Fig. 2 which is a combination of deep learning network and federated learning. The Fed-IDS architecture used in this work has been removed Shuffle layer and 1 GRU layer to fit the dataset and minimize calculation overhead while maintain the efficiency. The input x contains features that indicate whether this is an attack or not. x is passed through CNN and GRU model simultaneously. When passing through GRU x will be extracted into the features as consequential information with time and turned into v as in (3).

$$v = GRU(x) \tag{3}$$

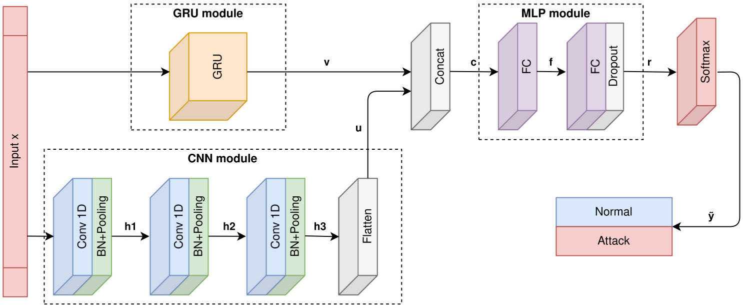

Fig. 2. The architecture of deep learning model - Fed-IDS.

When x comes to CNN module, it goes through 3 convolutional blocks, each includes a Convolutional 1D layer, a Batch Normalization layer, and a Pooling layer. The features are extracted in Convolutional 1D layer using convolution multiplication. Batch Normalize aims to standardize the values which were received from the Convolutional 1D layer at a moderate level, neither too large nor not too small. This is meaningful in retaining the features when going through many layers, many multiplications and avoiding arithmetic overflow. For example, when the number is too large and losing weight, since many small numbers multiplying together will result the zero value. And the last layer in each block is MaxPooling. Three convolutional blocks are denoted as $ConvBlock_1$, $ConvBlock_2$ and $ConvBlock_3$ respectively as in (4).

$$\begin{aligned} h_1 &= ConvBlock_1(x) \\ h_2 &= ConvBlock_2(h_1) \\ h_3 &= ConvBlock_3(h_2) \end{aligned} \tag{4}$$

The result after passing the $3^r d$ convolution block will be flattened by the Flatten layer as in (5).

$$u = Flatten(h_3) \tag{5}$$

The Concat layer concatenates the result u and v from GRU model and CNN model as in (6).

$$c = Concat(u, v) \tag{6}$$

This result c goes through the MLP module consisting of 2 Fully Connected layers and a Dropout layer to avoid overfitting. 2 Fully Connected layer were denoted as FC_1 and FC_2 respectively as in (7).

$$f = FC_1(c), r = Dropout(FC_2(f)) \tag{7}$$

Finally, the result r passes through the Softmax layer to form the probability $\widetilde{y}$ as in (8).

$$\widetilde{y} = Softmax(r) \tag{8}$$

The objective of this model is to optimize the cross-entropy loss function between the vector ouput $\widetilde{y}$ and vector y which contains the actual label values.

3.3 Poison Attack

Poison Attack is a type of attack that seeks to damage the FL model or reduce its predictability by interfering with the training process. This work performs two strategies of performing poisoning attack, including Label Flipping and Generative Adversarial Networks (GANs)-based Synthetic Data.

Poison Attack with Labels Flipping. Labels Flipping is a type of Data Poisoning Attack where an adversary relabels some samples in a training set to degrade the performance of the resulting classifier. Figure 3 depicts the interfering scheme of a rogue training collaborator.

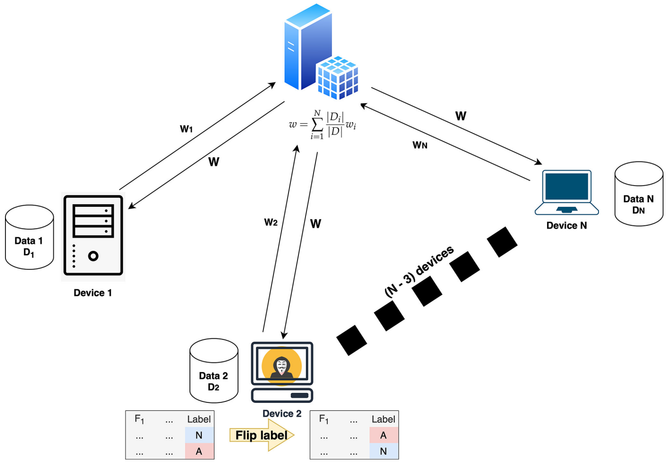

Fig. 3. Poison attack against Fed-IDS using flip-labels.

Poison Attack with Generative Adversarial Networks (GAN). In this strategy, the attacker pretends as a collaborator joining in the FL process and uses fake dataset generated to train the local model, then uploads the generated local parameters to the server. After aggregating from this collaborator, the global model will be poisoned. The overview of poisoning attack using GAN-based poisoned data generation is illustrated on Fig. 4.

To generate lots of artificial data aiming to poison the FL-based IDS, we design a GAN architecture, shown in Fig. 5. It has 2 main networks: Generator and Discriminator. The Discriminator is responsible for distinguishing between the actual data and the data generated by the Generator. Meanwhile, Generator perform the task of crafting new data to bypass the recognition of Discriminator.

The Generator has the same architecture as the CNN module of Fed-IDS scheme, consisting of three convolutional blocks $ConvBlock_1$, $ConvBlock_2$,

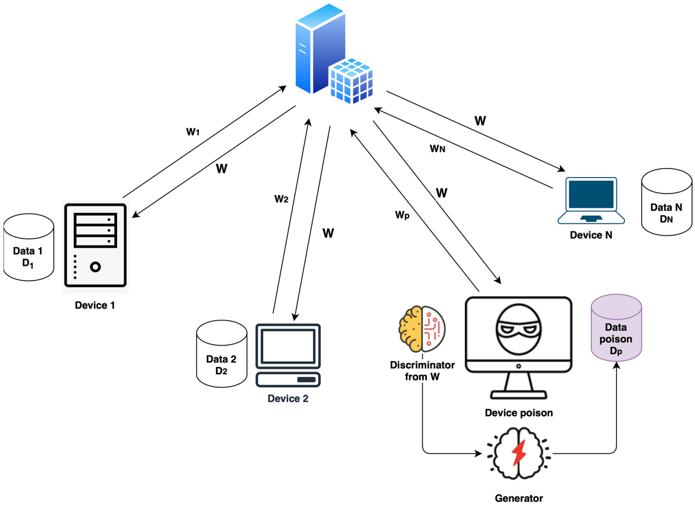

Fig. 4. Poisoning attack against Fed-IDS using GAN.

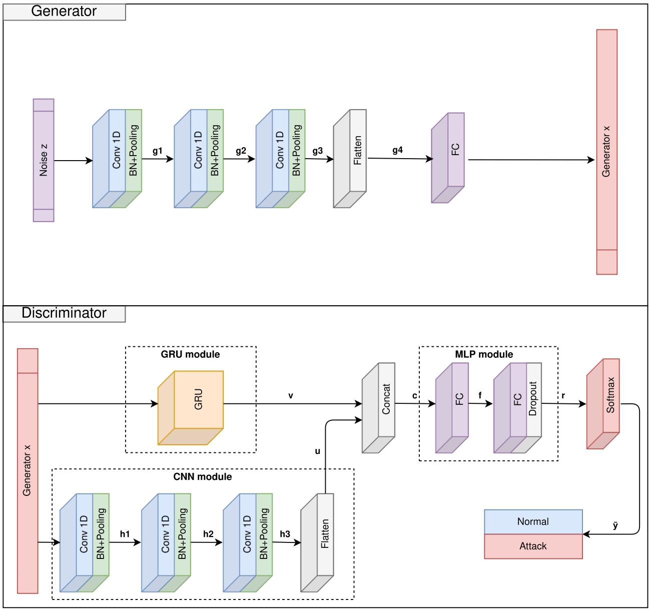

Fig. 5. The GAN-based poisoned data generation structure used in poisoning attack against Fed-IDS.

$ConvBlock_3$ and a Flatten layer. The generator G will take a random noise vector z and generate data x_{gen} as in (9).

$$x_{gen} = G(z) \tag{9}$$

The adversaries take advantage of a predictable model retrieved from the global server as Discriminator to mimic the capability of FL-based IDS in producing attack alerts. The distinguishing result of Discriminator is a vector representing the probability that a record of network traffic flow is attack or normal label. The vector has the form of $[normal, attack]$. And the sum of $normal$ and $attack$ is always 1. And the record of network traffic flow is Normal label if $normal > attack$, is Attack if $normal < attack$. Especially, when $normal = attack$, the network flow data can still be an attack label.

We assume $distance$ is the difference between $normal$ and $attack$ $distance = |normal - attack|$. If $distance$ is gradually approaching zero, it means that the distinguishing function in Discriminator is wondering if traffic data is an attack label. Conversely, the large $distance$, the more certainty that this is an attack label or normal label. From the above analysis, we propose the loss function of GAN as in (10):

$$loss_{generator} = \frac{N}{\sum_{i=1}^{N} (D(G(z_i))_{normal} - D(G(z_i))_{attack})^2} \tag{10}$$

D is the Discriminator that will check N generated by generator G, with z_i is the noise vecto i-th and $i = 1, 2, 3, \ldots, N$ $D(G(z_i))_{normal}$ is the probability that $G(z_i)$ is normal and $D(G(z_i))_{attack}$ is the probability that $G(z_i)$ is attacked. Finally, we just optimize loss function $loss_{generator}$ to find the generating function G corresponding to the discriminant function D. Then, we use G to generate the fake dataset for training. Algorithm 1 describes the training scheme of GAN for crafting new poisoning data records.

3.4 Collaborative Model Validation

As mentioned above, protecting FL from Poisoning Attack is a challenging task. To overcome this challenge, we aim to break the connection between the adversaries and the FL model by determining that a trained local model is good and appropriate for global aggregation.

Our defensive method for FL-based intrusion detection is built based on BaFFLe [3], a strategy for backdoor detection in FL-based image classifiers. To keep this defensive method highly effective, we make sure to have l safe models that are aggregated without being attacked to validate local models before updating to the global server. And D is a dataset saved in server which is used in validation process for anti-poisoning attack against FL-based IDS. We assume that $\mathcal{S}$ is defined as the global model aggregated on the server. And after l times aggregation, we have a set containing $\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_l$. With each S_i in l safe models above, there are two metrics of errors based on data and prediction results. The error based on data with formula $err_D(\mathcal{S}_i)^{y \to X}$ is interpreted as amount of data in D labeled y and misclassified by model $\mathcal{S}_i$. And the error based on predicted results with formula $err_D(\mathcal{S}_i)^{X \to y}$ is interpreted as amount

Algorithm 1. Generate fake data using GANs for poisoning attack

```
Input:
    Global model $S$;
    List of noise vector $\mathcal{Z}_{noise}$;
    Epochs $E$
Output: Fake data $D_{poison}$
$D \Leftarrow S$                                   ▷ Assign descriminator as global model
$G = Generator()$                                  ▷ Init generator
$e \Leftarrow 1$
while $e \leq E$ do                                ▷ Training with $E$ epochs
    Calculate $loss_{gen}$
    Optimize $G$
    $e \Leftarrow e + 1$
end while
$\mathcal{X}_{fake} = G(\mathcal{Z}_{noise})$
$D_{poison} = []$                                  ▷ Init fake data
for each $x \in \mathcal{X}_{fake}$ do
    $y = D(x)$                                     ▷ Get label
    $D_{poison}$.add($data : x, label : y$)        ▷ Add new record
end for
return $D_{poison}$
```

of data in D that model $\mathcal{S}_i$ incorrectly assigns to class y. Then, compute the two differences in errors between model $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ as in (11):

$$\begin{aligned} v^s(\mathcal{S}_i, \mathcal{S}_{i+1}, D, y) &= err_D(\mathcal{S}_i)^{y \to X} - err_D(\mathcal{S}_{i+1})^{y \to X} \\ v^t(\mathcal{S}_i, \mathcal{S}_{i+1}, D, y) &= err_D(\mathcal{S}_i)^{X \to y} - err_D(\mathcal{S}_{i+1})^{X \to y} \end{aligned} \tag{11}$$

Where Y is the set of labels of data, then we have 2 different vectors respectively as in (12):

$$\begin{aligned} \vec{v^s} &= [v^s(\mathcal{S}_i, \mathcal{S}_{i+1}, D, y)]_{y \in Y} \\ \vec{v^t} &= [v^t(\mathcal{S}_i, \mathcal{S}_{i+1}, D, y)]_{y \in Y} \end{aligned} \tag{12}$$

And $\mathbf{v}_i$ which characterizes the error difference between 2 models $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ is identified as in (13):

$$\mathbf{v}_i = [\vec{v^s}, \vec{v^t}] \tag{13}$$

With l safe models above, we have $\mathbf{v}$ as the set of the errors difference between 2 models as in (14):

$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_{l-1}] \tag{14}$$

Each subsequent model C from other IDS collaborators is uploaded to server is used to compute $\mathbf{v}_l$ based on $\mathcal{S}_l$. Then, the local model C is evaluated to receive permission to participate in the global model aggregation process by the Local Outlier Factor (LOF) [6] function. The verification process of each local model before aggregated into global models as parameter updates is summarized in Algorithm 2.

Algorithm 2. Verify a local IDS model before aggregation

```
Input:
    Array $\mathbf{v}$ are calculated from $l$ safe models;
    Verify model $\mathcal{C}$;
Output:
    0 or 1. 0 is not updated, 1 is updated
Calculate $\mathbf{v}_l$
$\mathbf{v}.\mathbf{add}(\mathbf{v}_l)$                          ▷ Add $\mathbf{v}_l$ into $\mathbf{v}$
for each $(v, i) \in \mathbf{v}$ do
    $lof[i] = LOF_{\lceil \frac{l}{2} \rceil}(v; \mathbf{v}/\{v\})$      ▷ Calculate LOF with $\lceil \frac{l}{2} \rceil$ neighbors
end for
$avg = mean(lof[1], lof[2]...lof[l-1])$
if $lof[l] < avg$ then
    return 1
else
    return 0
end if
```

4 Experimental Evaluation

In this section, the experiments are conducted to evaluate the performance of Fed-IDS scheme in normal condition, under poisoning attack, and the case of attack with defensive measure. First, we give data resource description and partitioning. Then, we focus on experimental settings, including the environmental settings, baseline studies, and performance metrics. Finally, we carry out a series of experiments to compare the performance of Fed-IDS model in basic case without attack, with attacked case, and with defense case.

4.1 Data Preprocessing

To experimental testing, we utilize Kitsune Network Attack Dataset [13], a collection of 9 network attack datasets captured from an IP-based commercial surveillance system or a network full of IoT devices. Each dataset contains millions of network packets and different cyberattacks. In this data resource, all collected records were labeled by two type states, including: Normal and Attack. And each piece of network data in this data resource contains 115 features and 1 label. And the column chart as in Fig. 6 shows the distribution of the two types of labels of each dataset in Kitsune.

5 Kitsune Dataset

We assess Fed-IDS on Active Wiretap as one of the datasets where the distribution of the two types of labels is quite similar. In our experiments, the Active Wiretap dataset is divided into two major parts, 75% for training and 25% for testing. The training part is divided into equally sized partitions to each collaborator for local model training. Note that all trained models are tested on the same testing data.

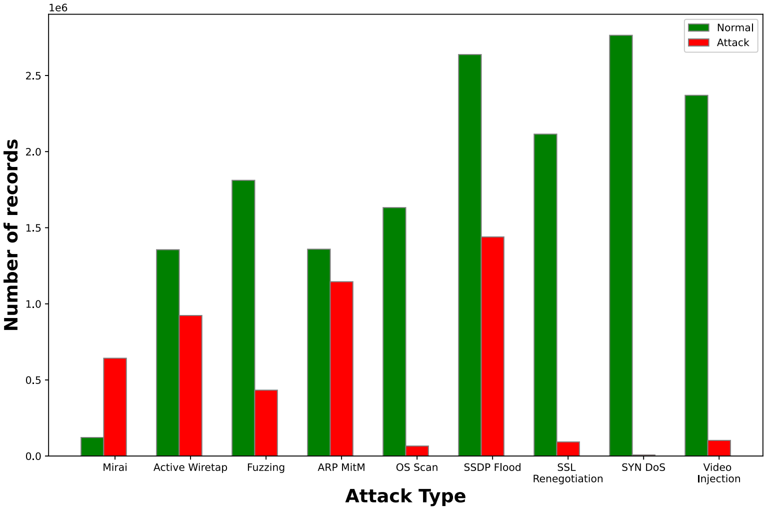

Fig. 6. Kitsune dataset distribution.

5.1 Experimental Settings

Environmental Setup. We implemented Fed-IDS as a federated learning algorithm by using the TensorFlow framework. The designed CNN-GRU model is implemented using Keras. Our experiments are conducted on Google Colab Pro with the Intel Xeon Processor CPU 2 cores 2.3 GHz and the Tesla T4 GPU with 27 GB RAM.

Baseline Studies. In this work, we compare the performance of Fed-IDS model in basic case without attack, with attacked case, and with defended case.

Performance Metrics. We use four metrics as follows: *Accuracy* is the ratio of right classifiers and total; *Precision* is the ratio of right predictions having attack label and total predictions belong to attack class; *Recall* is the right predictions having attack label over the sum of right predictions having attack label and misclassified belong to normal class; *F1-score* is calculated by two times the product of precision and recall over the sum of precision and recall.

5.2 Evaluation Result

In our experiments, we validate Fed-IDS model with above-mentioned baseline studies. We completed a total of 5 experiments, each experiment has the same circumstances with the numbers of agents K = 3, 5, 7 and 9 respectively. The numerical results which are shown in Table 1, Table 2 and Table 3 illustrate the performance of FL models, in terms of the accuracy and F-score, under four

Table 1. Performance of Fed-IDS without attack.

K	R	Accuracy	Precision	Recall	F1-Score
3	6	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987
	10	0.958631	0.907371	0.999974	0.951425
	12	0.99996	0.999926	0.999974	0.99995
5	6	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987
	12	0.999989	1.000000	0.999974	0.999987
7	6	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987
	12	0.999989	1.000000	0.999974	0.999987
9	6	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987
	12	0.999989	1.000000	0.999974	0.999987

Table 2. The results of Labels Flipping and Counterfeit Data Attack.

K	R	Labels Flipping				Counterfeit Data by GANs			
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
3	6	0.405152	0.405152	1.000000	0.576667	0.407654	0.406166	0.999974	0.577689
	8	0.970938	0.96929	0.95864	0.963936	0.407433	0.406076	0.999974	0.577598
	10	0.405152	0.405152	1.000000	0.576667	0.407308	0.406026	0.999974	0.577547
	12	0.405152	0.405152	1.000000	0.576667	0.405168	0.405159	1.000000	0.576673
5	6	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667
	8	0.405152	0.405152	1.000000	0.576667	0.594430	0.024000	0.000026	0.000052
	10	0.406913	0.405867	1.000000	0.57739	0.413366	0.408506	0.999974	0.580051
	12	0.405152	0.405152	1.000000	0.576667	0.594188	0.015464	0.000026	0.000052
7	6	0.412162	0.407969	0.999424	0.579417	0.592741	0.00495	0.000026	0.000052
	8	0.405152	0.405152	1.000000	0.576667	0.405653	0.405353	0.999974	0.576866
	10	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667
	12	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667
9	6	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667
	8	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667
	10	0.432805	0.414464	0.968982	0.580591	0.405152	0.405152	1.000000	0.576667
	12	0.405152	0.405152	1.000000	0.576667	0.405152	0.405152	1.000000	0.576667

Table 3. The results of Labels Flipping and Counterfeit Data Attack with Validation.

K	R	Labels Flipping				Counterfeit Data by GANs			
		Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
3	6	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	12	0.955455	0.900963	0.999974	0.94789	0.999989	1.000000	0.999974	0.999987
5	6	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	8	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987	0.972012	1.000000	0.930920	0.964224
	12	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
7	6	0.999988	1.000000	0.99997	0.999985	0.999989	1.000000	0.999974	0.999987
	8	0.970969	0.952534	0.977032	0.964628	0.999989	1.000000	0.999974	0.999987
	10	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987
	12	0.847266	0.993818	0.626922	0.768842	0.993895	0.985180	0.999974	0.992522
9	6	0.999989	1.000000	0.999974	0.999987	0.982936	0.959607	0.999974	0.979375
	8	0.979997	0.952974	0.999974	0.975909	0.999989	1.000000	0.999974	0.999987
	10	0.910621	0.819278	0.999974	0.900652	0.999989	1.000000	0.999974	0.999987
	12	0.999989	1.000000	0.999974	0.999987	0.999989	1.000000	0.999974	0.999987

different scenarios of agents with the numbers of rounds communication R = 6, 8, 10 and 12, respectively. It can be easily seen that two types of attacks have affected to the performance of Fed-IDS model without defensive measure. And the defense measure has also given the effective to recognize and remove poisoned model before uploading. As the number of communication rounds R increases from 6 to 12, the performance of Fed-IDS model stabilizes when R is sufficiently large. Although, the results of evaluating in FL model without attacked demonstrate predictability with the accuracy and F-score approximately 99.9989% and 99.9987% respectively, the accuracy and F-score dropped sharply at 40.5152% and 40.5152% respectively when model training is attacked with both 2 types: labels flipping and generating counterfeit data by GANs. However, the validated model uploaded to global server can significantly improve the predictability of the aggregated global model. And the validation results in both attack strategies obtain the accuracy and F-score of 99.9989% and 99.9987% respectively. It is clear that Fed-IDS model with validation approach can resolve the problems of Poison Attack against FL-based intrusion detection.

6 Conclusion

In this paper, we have introduced and evaluated Fed-IDS, a federated deep learning model for intrusion detection in the context of IIoT networks. We also evaluated Fed-IDS in the context of being attacked and defending from attacks. First, we built a FL model based on DeepFed [11] for multiple machines participating in local collecting and training to build and aggregate a comprehensive intrusion detection model. Then, we built GAN architecture to generate counterfeit data to attack our FL model. And we also attack our model with labels flipping, which is another data poison attack to compare and evaluate how much data poison affects the accuracy of our model. In addition, we combined a BaFFLe-based method validation [3] with FL model to enhance the performance of our model. The experiments on Kitsune Network Attack Dataset have demonstrated that the FL model is vulnerable and validate function are adoptable for covering and defending the model from attacks. It is worth noting that FL model with validation is very suitable for building a federated intrusion detection system in the context of IIoT networks. Future research directions will focus on encrypt parameters before sending to global model for escaping be divulged information by analyzing uploaded parameters in the context of FL for IDS in IIoT networks.

Acknowledgement. This research is funded by University of Information Technology – Vietnam National University Hochiminh City under grant number D1-2021-28.

Phan The Duy was funded by Vingroup Joint Stock Company and supported by the Domestic Master/ PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA), code VINIF.2020.TS.138.

References

1. Abdel-Basset, M., Hawash, H., Sallam, K.: Federated threat-hunting approach for microservice-based industrial cyber-physical system. IEEE Trans. Ind. Inform. **18**(3), 1 (2022)
2. Aledhari, M., et al.: Federated learning: a survey on enabling technologies, protocols, and applications. IEEE Access **8**, 140699–140725 (2020)
3. Andreina, S., et al.: BaFFLe: backdoor detection via feedback-based federated learning, November 2020
4. Adversarial label-flipping attack and defense for graph neural networks. In: 2020 IEEE International Conference on Data Mining (ICDM) (2020). IEEE Trans. Ind. Inform
5. Bouacida, N., Mohapatra, P.: Vulnerabilities in federated learning. IEEE Access **9**, 63229–63249 (2021). https://doi.org/10.1109/ACCESS.2021.3075203
6. Breunig, M., et al.: LOF: identifying density-based local outliers, vol. 29, pp. 93–104, June 2000
7. da Costa, K.A.P., et al.: Internet of Things: a survey on machine learning-based intrusion detection approaches. Comput. Netw. **151**, 147–157 (2019). ISSN 1389-1286
8. Hindy, H., et al.: A taxonomy of network threats and the effect of current datasets on intrusion detection systems. IEEE Access **8**, 104650–104675 (2020)
9. Kenyon, A., Deka, L., Elizondo, D.: Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. Comput. Secur. **99**, 102022 (2020). ISSN 0167-4048
10. Khan, L.U., et al.: Federated learning for Internet of Things: recent advances, taxonomy, and open challenges. IEEE Commun. Surv. Tutor. **23**(3), 1 (2021)
11. Li, B., et al.: DeepFed: federated deep learning for intrusion detection in industrial cyber-physical systems. IEEE Trans. Ind. Inform. **17**(8), 5615–5624 (2021)

12. Lyu, L., Yu, H., Yang, Q.: Threats to federated learning: a survey (2020). arXiv:2003.02133 [cs.CR]
13. Mirsky, Y., et al.: Kitsune: an ensemble of autoencoders for online network intrusion detection. In: The Network and Distributed System Security Symposium (NDSS) 2018 (2018)
14. Mishra, P., et al.: A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun. Surv. Tutor. **21**(1), 686–728 (2019)
15. Mothukuri, V., et al.: A survey on security and privacy of federated learning. Future Gener. Comput. Syst. **115**, 619–640 (2021). ISSN 0167-739X
16. Neshenko, N., et al.: 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), 2702–2733 (2019)
17. Nguyen, T.D., et al.: Poisoning attacks on federated learning-based IoT intrusion detection system. In: Workshop on Decentralized IoT Systems and Security (DISS) @ NDSS Symposium 2020 (2020)
18. Nguyen, T.D., et al.: DÏoT: a federated self-learning anomaly detection system for IoT. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 756–767 (2019)
19. Rahman, S.A., et al.: Internet of Things intrusion detection: centralized, on-device, or federated learning? IEEE Netw. **34**(6), 310–317 (2020)
20. Sommer, R., Paxson, V.: Outside the closed world: on using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy, pp. 305–316 (2010)
21. Sun, G., et al.: Data poisoning attacks on federated machine learning (2020). arXiv:2004.10020 [cs.CR]
22. Tolpegin, V., et al.: Data poisoning attacks against federated learning systems, July 2020
23. Wang, X., et al.: Towards accurate anomaly detection in industrial Internet-of-Things using hierarchical federated learning. IEEE Internet of Things J. 1 (2021)
24. Yang, Q., et al.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**, 1–19 (2019). ISSN 2157-6904
25. Zhang, J., et al.: PoisonGAN: generative poisoning attacks against federated learning in edge computing systems. IEEE Internet of Things J. **8**(5), 3310–3322 (2021)
26. Zhang, J., et al.: Poisoning attack in federated learning using generative adversarial nets. In: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 374–380 (2019)