Future Generation Computer Systems 134 (2022) 154–169

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

SecFedNIDS: Robust defense for poisoning attack against federated learning-based network intrusion detection system

Zhao Zhang [a], Yong Zhang [a,b,*], Da Guo [a], Lei Yao [a], Zhao Li [a]

[a] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[b] Beijing Key Laboratory of Work Safety Intelligent Monitoring, Beijing University of Posts and Telecommunications, Beijing, 100876, China

ARTICLE INFO

Article history:
Received 9 November 2021
Received in revised form 6 April 2022
Accepted 9 April 2022
Available online 18 April 2022

Keywords:
Network intrusion detection
Federated learning
Poisoning attacks
Defensive mechanism
Poisoned model detection
Poisoned data detection

ABSTRACT

Federated learning-based network intrusion detection system (FL-based NIDS) has demonstrated tremendous potential in protecting the security of IoT network. It enables learning an effective intrusion detection model from massive traffic data collaboratively without data privacy leakage. However, FL-based NIDS has exhibited inherent vulnerabilities on the poisoning attacks launched by malicious clients. The poisoning attacks aim to corrupt the intrusion detection model and impair its protection capability, by injecting the poisoned traffic data into the local training dataset. We build a secure FL-based NIDS that is robust for the poisoning attacks, namely SecFedNIDS. Firstly, we propose the model-level defensive mechanism based on poisoned model detection. Specifically, we propose the gradient-based important model parameter selection method to provide the effective low-dimensional representations of the uploaded local model parameters, and then we propose the online unsupervised poisoned model detection method to identify the poisoned models and reject them to join in the global intrusion detection model. Subsequently, we design the data-level defensive mechanism based on poisoned data detection. Notably, we propose a novel poisoned data detection method based on class path similarity, to filter out the poisoned traffic data and avoid them participating in subsequent local training. We adopt layer-wise relevance propagation to extract the class path of clean traffic data, and transmit the class paths to the poisoned clients to help distinguish the poisoned traffic data. Results show that SecFedNIDS with the proposed model-level defense boosts the accuracy by up to 48% under the poisoning attacks on UNSW-NB15 dataset and 36% on CICIDS2018 dataset, and the proposed data-level defense further improves its accuracy by up to 13% on CICIDS2018 dataset.

1. Introduction

Artificial Intelligence of Things (AIoT), combining Artificial Intelligence and Internet of Things, is an emerging trend to provide intelligent communication and efficient data processing among IoT devices. But the access of massive intelligent terminals in AIoT network has exposed the enormous security risks. The deployment of network intrusion detection system is an effective way to protect network security. Recently, due to the large-scale network environment, federated learning [1] (FL), as the emerging distributed deep learning paradigm, has been widely applied in network intrusion detection system in IoT network [2–5]. FL provides a means to obtain an effective intrusion detection model, by training the traffic data from massive IoT devices collaboratively without the IoT data privacy leakage. Specifically, the local clients perform the local training and upload their local intrusion detection models to the central server, the server aggregates all the local models to obtain the global intrusion detection model. Besides, owning to the heterogeneous nature of IoT network, FL-based NIDS will confront non-independent identically distributed (Non-IID) traffic data across devices. FL still has a certain ability of processing Non-IID data, which can ensure the effectiveness of FL-based NIDS in realistic scenario.

However, FL-based NIDS has exhibited inherent vulnerabilities on the poisoning attacks launched by malicious clients [6], since the central server has no access to the local datasets and cannot govern the malicious behaviors of the local clients in FL system. The attackers seek to corrupt the intrusion detection model and impair its protection capability of network security, through injecting the poisoned traffic data into the local training dataset. In this paper, we consider two common poisoning attacks against FL-based NIDS, including label flipping attack and clean label attack. The label flipping attack randomly flips the label of the traffic sample to another without changing the traffic features, e.g., the attackers change the labels of the malicious traffic data as the benign ones, causing the intrusion detection model trained on

* Corresponding author at: School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
E-mail address: yongzhang@bupt.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.future.2022.04.010
0167-739X/

them to misclassify the malicious traffic as the benign traffic. The clean label attack does not change the data label, but modifies the traffic features adversarially. Note that, to ensure the functionality and intactness of traffic data, the modification of traffic features must comply with the traffic domain constraints, e.g., the features extracted from the backward traffic flow packets generated by the victim network cannot be modified [7]. Thus, in consideration of the traffic domain constraints, we combine the C&W attack algorithm [8] to generate the adversarial traffic sample to implement the clean label attack.

We build the secure and robust FL-based NIDS to defend against the poisoning attacks, namely SecFedNIDS. Firstly, we first develop the robust model-level defensive mechanism for the poisoning attacks against FL-based NIDS. The newest research about the model-level defense is based on the poisoned model detection methods [9–12], which deploys an offline detection model at the central server side to detect the poisoned models and reject them to join in the global model aggregation. But the existing research on the detection-based defense methods still face two major challenges in the application for FL-based NIDS: (1) The intrusion detection model based on deep neural network equips with a large number of model parameters, but the poisoned model detection method is incapable of dealing with these high-dimensional data and the existing random model parameter selection method [9] is not effective enough. (2) The traffic data collected from realistic IoT environment are usually heterogeneous and time-varying. The data distribution shift will bring in the distribution shift between the local model parameters accumulated before the detection (for training the offline detection model in advance) and the local model parameters obtained during the detection, leading to disable the offline detection model in realist environment. To tackle the two major challenges, we first propose the gradient-based important model parameter selection method to obtain the effective low-dimensional representation of the local model parameters for the poisoned model detection. Then, we propose the online unsupervised poisoned model detection method based on stochastic outlier selection, without training the offline detection model in advance.

The model-level defensive mechanisms adopts the direct rejection of the poisoned models to mitigate the negative effect, which will also cause the decline of the amount the training traffic data and affect the convergence speed and generalization ability of the global model, especially on Non-IID data [10]. To further enhance the defense performance of SecFedNIDS under the poisoning attacks, we propose the data-level defense based on poisoned data detection, filtering out the poisoned traffic data and retraining the local model with the remaining clean data to rejoin in the global model. Current poisoned data detection methods [13–15] focus on the centralized learning system and require to directly access all the training data, but these methods are unsuitable for FL-based NIDS due to the data privacy issue. Additionally, each client may not contain the complete data information for detecting the poisoned data under the Non-IID data setting. To cope with these issues, we transmit the surrogate data of the clean traffic data instead of the original data to the poisoned client, which can preserve traffic data privacy and help to distinguish the poisoned traffic data. Concretely, we apply the exiting layer-wise relevance propagation (LRP) method [16] to extract the class path as the surrogate data, which is constituted by the critical neurons in deep neural network that make significant contributions toward the model decision of the traffic samples. It has been proven that the samples from the same class usually activate the similar paths and different classes activate different paths in deep neural network [17]. Inspired by it, we propose the poisoned data detection method based on class path similarity.

The main contributions are summarized as follows:

- We model two poisoning attacks against FL-based NIDS in consideration of traffic domain constraints, including label flipping attack and clean label attack. Particularly, we adapt the C&W attack algorithm to generate the adversarial traffic sample to implement the clean label attack.
- To defend against poisoning attacks, we propose the secure and robust FL-based NIDS, SecFedNIDS, including the model-level defensive mechanism and the data-level defensive mechanism.
- For the proposed model-level defense, we propose the gradient-based important model parameter selection method to provide the effective low-dimensional representations of the uploaded local model parameters, and then we propose the online unsupervised poisoned model detection method.
- Regarding to the proposed data-level defense, we propose a novel poisoned data detection method based on class path similarity, where the class path is extracted by the layer-wise relevance propagation method.

The remainder of this paper is organized as follows. A brief review of related works is provided in Section 2. Section 3 details the system architecture of FL-based NIDS and the attack model against it. In Section 4, we introduce the proposed defensive mechanisms for poisoning attacks. The experiment settings and results are shown in Section 5. Section 6 comes to conclusion remarks.

2. Related work

2.1. FL-based NIDS

In virtue of the privacy-preserving capabilities and the low communication costs, federated learning has been widely applied in network intrusion detection system for IoT networks [2–5]. Fan et al. [2] proposed the FL-based intrusion detection framework for 5G IoT. This framework built the network intrusion detection model based on convolutional neural network (CNN) and aggregated various local intrusion detection models through federated learning to train a powerful intrusion detection model. It was verified on CICIDS2017 dataset and was more effective than traditional ML methods. The network intrusion detection model combining FL and CNN was proposed in [3], called CNN-FL. It can train the global model for multiple participants without sharing private data. Experimental results on NSL-KDD dataset show that the proposed CNN-FL model achieves higher performance than other methods in binary classification and multi-classification. Li et al. [4] proposed an intrusion detection system in Industrial Internet of Things (IIoT) based on federated deep learning, DeepFed. It combined CNN and gated recurrent unit to learn local representations and utilized FL to leverage the data resources from multiple industrial owners. FL-based attention gated recurrent unit for intrusion detection system in wireless edge networks was proposed in [5]. It was evaluated with the IID and Non-IID data on KDD'CUP 99 dataset, CICIDS2017 Dataset, and WSN-DS Wireless Network Dataset.

However, these researches focus on discussing and evaluating the performance of FL-based NIDS compared to the advantages of centralized systems, and do not further analyze their robustness in the face of poisoning attacks.

2.2. Poisoning attack against FL system

Federated learning systems are vulnerable to poisoning attacks from malicious clients, as the central server of an FL system cannot control the behavior of local clients or directly access

the local datasets. According to the attack modes of malicious participants, poisoning attacks can be divided into data poisoning attacks and model poisoning attacks [18]. Data poisoning attack: malicious participants modify local datasets through injecting the poisoned data into local datasets, e.g., through label flipping. Training with these poisoned data will disturb the decision boundary and then generate the poisoned model updates. These poisoned models will upload to the central server and ultimately affect the accuracy of the aggregated global model. Model poisoning attack: malicious participants generate random local model weights through predefined rules without modifying local datasets. The direct manipulation on model weights can also acquire the poisoned model updates, so that causes substantial drops in the performance of the global model.

In this paper, we focus on data poisoning attacks against FL-based NIDS, for the reason that each client in FL-based NIDS requires to collect training traffic data from the local network and offers the attackers great opportunities to inject poisoned traffic data into the local datasets. It has also been verified in [6] that the attackers in FL-based IoT intrusion detection system can gradually poison the detection model by only utilizing compromised IoT devices (and not gateways/clients) to inject poisoned traffic data into local training datasets. The data poisoning attack against FL-based NIDS is simple-to-implement and highly-effective. On the contrary, the model poisoning attacks against FL-based NIDS are extremely difficult to carry out in practice, since the attacker needs to control the whole well-protected IDS to modify the local model weight in targeted manners, but actually the attackers do not have such strong ability.

2.3. Defense for poisoning attack against FL

2.3.1. Model-level defense

Recently, the existing model-level defense methods for poisoning attack against FL system fall into two simple categories: robust aggregation-based [19–22] and poisoned model detection-based [9–12].

The robust aggregation-based approaches are to use a more robust model aggregation algorithm instead of using the model average aggregation, which can be broadly divided into two categories. One class of these approaches is to select a representative local model and use it to estimate the global model. Among them, Krum [19] selected the local model parameters with the shortest distance from other clients as the global model' parameters. Medoid [20] was proposed to estimate the global model by using the medoid local model. The other class is to estimate the global model based on all the model updates from clients. Geomed [21] adopted the geometric median of all local model parameters as the global model's parameters. The trimmed mean of all local model parameters was utilized as the global model in TrimMean [22]. The robust aggregation-based approaches are easy to implement and do not rely on additional datasets, but their performances are generally limited with Non-IID distributed settings.

The poisoned model detection-based methods are the latest poisoning attack defense methods for federal learning, which deploy a powerful detection model on the central server side to detect the poisoned model and reject it to join the global model aggregation. The variational autoencoder (VAE) anomaly detection model for poisoned model detection in FL system was proposed in [9], to detect label flipping attack, sign flipping attack, and additive noise attack. Experiments demonstrate that it has better defense ability against poisoning attack than Krum and Geomed. [10] proposed a binary anomaly detection model based on deep neural network to detect intentional and unintentional malicious models in FL system, it removed the detected malicious models from aggregation to alleviate their negative effects. In [11], an isolated forest-based malicious model detection mechanism, D2MIF, was proposed for federated learning of authorized AIoT, and it was verified on MNIST and Fashion MNIST dataset. [12] proposed a robust federated Industrial Internet of Things architecture for detecting Android malicious applications, and proposed GAN network (A3GAN) defense algorithm for detecting malicious poisoned models at the server side to avoid aggregation anomalies.

2.3.2. Data-level defense

There has been a large amount of research on the poisoned data detection for data-level defense [13–15], but these studies have mainly focused on the defense of centralized learning system. Paudice et al. [13] proposed the poisoned data detection method based on k-Nearest-Neighbors (k-NN) to mitigate the effect of label flipping attacks. [14] proposed a novel Deep k-NN defense for clean label poisoning attacks. The experiments on CIFAR-10 dataset showed that the proposed strategy could detect over 99% of poisoned samples generated by the clean label attacks. Chen et al. [15] proposed the defense strategy based on Generative Adversarial Networks (GANs) for detecting the data poisoning attack, called as De-Pois. It employed the mimic model to distinguish the poisoned data from the testing samples by comparing the difference between the mimic model's output and a properly determined detection boundary. These poisoned data detection methods require to have access to all the training data in the detection procedure, but it is unable to access all the local clients' training traffic data in FL-NIDS due to the privacy protection of IoT data. Hence, we seek to achieve the poisoned data detection without privacy leakage.

3. Poisoning attacks against FL-based NIDS

In this section, we provide a detailed description of the system architecture for the federated learning-based network intrusion detection system (FL-based NIDS) and introduce the attack model against FL-based NIDS.

3.1. System architecture of FL-based NIDS

We consider a typical federated learning-based network intrusion detection system consisting with N local intrusion detection clients and a central server. It can collaboratively train an effective intrusion detection model using the traffic data from a large number of IoTDs without data privacy leakage, as shown in Fig. 1. **The local intrusion detection client** is generally the security gateway that can monitor and collect the traffic data from IoTDs in the local network and train its own local intrusion detection model. **The central server** is typically the central entity or the cloud server that can aggregate all the local intrusion detection models.

Each client owns a private local training dataset D_i, where D_i may contain the traffic data of the normal behavior and various network attacks that collected from the local IoT network. During round t, each client performs the local training on D_i and produces the local model M_i^{t+1}. Subsequently, the client transmits the local model parameters M_i^{t+1} to the central server. At the central server side, the global model M^{t+1} is obtained by the average aggregation of all the local model, and then is broadcast to each client for the local model parameter initialization at the next round. The process of federated learning can be formulated as:

$$\textbf{Local client}: \quad M_i^{t+1} = M_i^t - \alpha \nabla L_i(M_i^t, D_i) \tag{1}$$

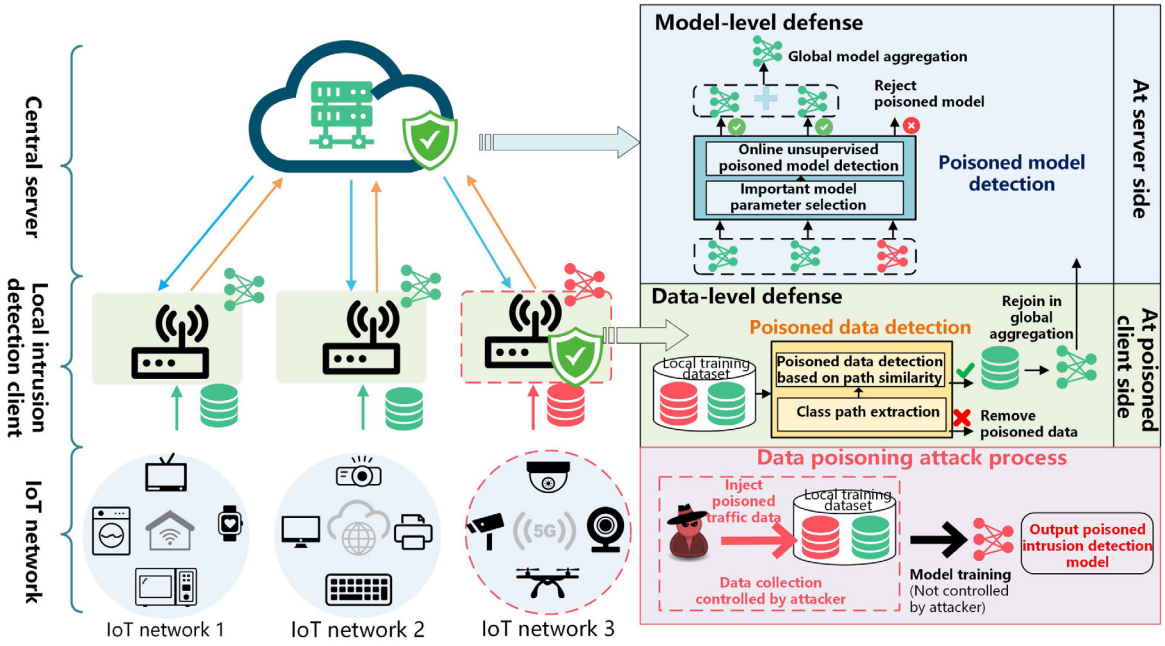


Fig. 1. The whole framework of SecFedNIDS with the proposed defensive mechanisms for the poisoning attacks.

$$\textbf{Central server}: \quad M^{t+1} = \sum_{i=1}^{N} M_i^{t+1} \tag{2}$$

where α is the learning rate, $\nabla L_i(M_i^t, D_i)$ is the gradient of local optimization loss, and c_i represents the proportion of the local training dataset D_i in the whole dataset.

3.2. Attack model

As the malicious behaviors of the local clients are invisible to the central server in FL system, FL-based NIDS has exhibited inherent vulnerabilities on the poisoning attacks launched by malicious clients [6]. In this paper, our attack model considers the poisoning attacks against FL-based NIDS. Assuming that the central server is honest and not compromised, and that the number of the malicious clients controlled by attackers does not exceed 50% of the total number of clients. Note that the hypotheses about the attack model apply in cases where system designers seek to improve the robustness of their NIDS, as suggested in [6].

Attackers' goal: The attacker aims to corrupt the global intrusion detection model and make it provide misjudgments for the specific traffic data, e.g., the intrusion detection model will detect the malicious traffic as the benign traffic so that it cannot provide sufficient protection for network security.

Attackers' capability: The attacker can access and modify the traffic data in the training dataset on their own clients, but cannot manipulate the local model training process. Notably, to preserve the functionality and intactness of the traffic data, the modification of traffic data needs to comply with the traffic domain constraints [7]. Unlike the pixels in the image that can be modified arbitrarily, some specific features of the traffic flow cannot be modified, e.g., the features extracted from the backward traffic flow packets generated by the victim network cannot be controlled over.

Attackers' strategy: The attackers take into consideration two most common poisoning attacks against the FL-based NIDS, including the label flipping attack and the clean label attack. The two poisoning attacks are simple-to-implement and highly-effective in practical environment. For the label flipping attack, it can randomly flip the labels of the traffic samples in the training dataset to the other labels without changing the traffic features, e.g., the attackers change the labels of the malicious traffic data as the benign ones, causing the intrusion detection model trained on them to misclassify the malicious traffic as the benign traffic. For the clean label attack, it does not change the labels of the training data, but adversarially changes the training traffic features to corrupt the obtained intrusion detection model [23]. That is, training with these adversarial traffic samples will change the judging boundary of the intrusion detection model and further result in the misjudgments of the clean traffic data during inference. In this paper, we adapt the existing adversarial sample crafting algorithm, Carlini and Wagner (C&W) attack [8], to generate the adversarial traffic sample that respect the traffic domain constraints. Specifically, we generate the adversarial traffic sample by minimizing the adversarial disturbance and maximizing the classification errors, while taking account into these traffic features that cannot be modified, which can be expressed as,

$$\min_{\delta}(||\delta \odot I||_2) + c \cdot f(x + \delta \odot I) \tag{3}$$

where δ is the adversarial disturbance, $\odot$ represents element-wise vector multiplication, and I is the mask of the unmodifiable features, that is, it is 0 if encountering with the unmodifiable features, otherwise it is 1. Note that the unmodifiable features in our experiments display in Section 5.1. The second term denotes as the maximum probability of the intrusion detection model misclassifying the sample as non-target classes, calculated by $f(x) = \max(\max\{M(x)_i : i \neq t\} - M(x)_t)$. For the generated adversarial traffic sample $x + \delta$, we reserve their original labels, so it is called as clean label attack. For example, we generate the adversarial samples for the malicious traffic and keep the malicious labels.

Eventually, the attackers inject the poisoned traffic data generated by the poisoning attacks into the local training dataset, and then conduct the model training to output the poisoned intrusion detection model. These poisoned local models will upload to the central server to further corrupt the global intrusion detection model. The data poisoning attack process is given in Fig. 1. We do note that the attackers just compromise the data collection procedure not the model training.

4. Defense mechanism

In this work, we aim to build a secure FL-based NIDS that is robust for the poisoning attacks, namely SecFedNIDS. The whole framework of SecFedNIDS presents in Fig. 1.

Firstly, we propose the model-level defense at the server side. Specifically, we propose the poisoned model detection method to identify whether the uploaded local model is a poisoned model. Once the poisoned models are detected, the model-level defensive mechanism will reject them to join in the global intrusion detection model aggregation to mitigate the negative influence of poisoning attacks.

Secondly, we propose the data-level defense at the local client side. To be specific, at the poisoned client side, we propose the poisoned data detection method to filter out the poisoned training traffic data, so as to avoid the negative influence of the poisoned data participating in the next round of model training. Different from the model-level defense that directly reject the participations of all local training data at the poisoned clients, the data-level defense filters out the poisoned data and retrains the local model with the remaining clean data to rejoin in the global model aggregation. It can further improve the generalization ability and convergence speed of the global model on Non-IID data.

4.1. Poisoned model detection for model-level defense

In this section, we propose the poisoned model detection method to achieve the model-level defense in FL-based NIDS. Current detection-based defense methods require training an offline anomaly detection model with large amounts of local models' parameters accumulated before the detection. Although the existing detection-based methods have achieved success in defending poisoning attacks, the applications of these detection-based methods in the scenario of FL-based NIDS still confront two main challenges. The first challenge is derived from a large number of model parameters in the deep neural network-based intrusion detection model, which cannot be processed directly by the poisoned model detection method. Considering that the existing random model parameter selection method is not efficient enough, we first propose the gradient-based important model parameter selection method for the poisoned model detection, obtaining the effective low-dimensional representation of the local model parameters. The second challenge is that the collected traffic data from the realistic IoT environment usually are heterogeneous and time-varying. The distribution shift of training data will bring in the distribution shift between the local model parameters accumulated before the detection and the local model parameters obtained during the detection, thus it is probably to disable the offline detection model in realistic scenario. To solve the second challenge, we propose the online unsupervised poisoned model detection method based on stochastic outlier selection algorithm, without training a poison model detection model in advance.

4.1.1. Gradient-based important model parameter selection

Due to the inability of processing the high-dimensional model parameters, it is necessary to acquire the effective low-dimensional representation for the uploaded intrusion detection model parameters. In this paper, we propose the gradient-based important model parameter selection method, choosing the important model parameters as the dimensionality-reduced model parameters for the subsequent poisoned model detection. Since the gradient represents the model parameter changes related to the model prediction result, it can be used to evaluate the importance factor of model parameter, which are widely applied in continual learning [24] and network pruning [25]. Moreover, the gradient updates are also used for the global model aggregation, thereby the gradient-based important parameters are also conducive to the subsequent poisoned model detection.

Firstly, we calculate the importance factor of the local model parameter based on the gradient. Specifically, considering the change in local training loss caused by the infinitesimal parameter update δ_i^t at round t, it can be estimated by the gradient and is written as,

$$L(M_i^t + \delta_i^t) - L(M_i^t) \approx \sum_{k=1}^{|M|} g_{i,k}^t \Delta M_{i,k}^t \tag{4}$$

where $|M|$ is the number of the model parameters, $\Delta M_{i,k}^t$ represents the kth parameter of the ith local model update in round t, $g_{i,k}^t$ denotes as the kth parameter of the gradient in round t, and $g_{i,k}^t \Delta M_k^t$ represents the contribution of the kth parameter update to the change in the local training loss. Then, by adding up all infinitesimal changes in all iterations of round t, the total amount of the local training loss change in the whole parameter space can be obtained by,

$$\Delta L_i^t = \sum_{k=1}^{|M|} \int_t g_{i,k}^t \Delta M_{i,k}^t dt = \sum_{k=1}^{|M|} -w_{i,k}^t \tag{5}$$

where $w_{i,k}^t$ denotes as the contribution of the kth parameter to the total local training loss change. We introduce the minus sign in front of $w_{i,k}^t$, as we are typically interested in the loss decline. At last, considering that the importance factor of each parameter of the local model is determined by how much it contributed to the decline of the local training loss, the importance factor of the kth parameter of the ith local model $\Omega_{i,k}^t$ can be estimated by $w_{i,k}^t$,

$$\Omega_{i,k}^t = \max\left(\frac{w_{i,k}^t}{(\Delta_{i,k}^t)^2 + \xi}, 0\right) \tag{6}$$

where $\Delta_{i,k}^t$ denotes as the cumulative changes of the kth parameter, ξ is the damping factor that avoids the denominator to be zero. The importance factors of all the parameters of the ith local model is denoted as $\Omega_i^t = \{\Omega_{i,1}^t, \ldots, \Omega_{i,k}^t, \ldots, \Omega_{i,|M|}^t\}$.

Secondly, we define the important model parameters of the ith local model by considering the top K largest values of Ω_i^t. Concretely, we introduce the bitmask $Mi\_mask_i^t$ to mark the important model parameters in the ith local model, each element of it is expressed as,

$$Mi\_mask_{i,k}^t = \begin{cases} 1, & if\ k \in TopK(\Omega_i^t) \\ 0, & else \end{cases} \tag{7}$$

where $TopK(\cdot)$ is to obtain the indexes of the top K largest values, and 1 indicates that the kth parameter is one of the TopK important model parameters, otherwise, it is 0.

Thirdly, we aggregate all the local models' information to obtain the global important model parameters, due to the fact that the important model parameters of all the local models are typically inconsistent. In detail, each client uploads the local model parameters M_i^t to the central server, along with its bitmask $Mi\_mask_i^t$. And then, the central server aggregates all the local models' $Mi\_mask_i^t$ to obtain the global bitmask M_mask^t. The global bitmask is to mark the global important model parameters, each element of it is given by,

$$M\_mask_k^t = \begin{cases} 1, & if\ k \in TopK(\sum_{i=1}^{N} Mi\_mask_i^t) \\ 0, & else \end{cases} \tag{8}$$

Lastly, based on the global bitmask M_mask^t, we obtain the dimensionality-reduced local model parameters, $\hat{M}_i^t = NonZero(M_i^t \odot M\_mask^t)$. For all the local models, these dimensionality-reduced local model parameters are used for subsequent unsupervised poisoned model detection.

4.1.2. Online unsupervised poisoned model detection

After obtaining the low-dimensional local model parameters, we perform the poisoned model detection. Unlike the existing detection-based defense methods that employ an offline model to detect the uploaded local models in each round, our method adopts the online detection for the uploaded local models, that is, each round corresponds to a specific detection model. The online detection model is obtained by the local model parameters uploaded in each round. Generally, the uploaded local models in each round consist of the poisoned models and the clean models, among which the distributions of the poisoned model parameters vary from those of the clean model. Thus, we propose the online unsupervised poisoned model detection, by adopting the existing probability-based unsupervised anomaly detection algorithm, Stochastic Outlier Selection (SOS) algorithm [26]. As SOS does not require a large amount of training data and is easy to calculate, it is suitable for the scenario of online unsupervised poisoned model detection.

We take all the dimension-reduced local models uploaded in round t as the inputs of SOS algorithm. This algorithm employs the concept of affinity to quantify the relationship among the uploaded local model parameters. Based on this relationship, when a local model has insufficient affinity with all other models, it is most probably a poisoned model. For the dimension-reduced local model $\hat{M}_i^t$, the probability that $\hat{M}_i^t$ belongs to the poisoned model is calculated by,

$$P(\hat{M}_i^t \in PM) = \prod_{j, j \neq i} (1 - B_{ji}) \tag{9}$$

where PM represents the poisoned model. B_{ji} denotes as the normalized affinity between $\hat{M}_i^t$ and $\hat{M}_j^t$, given by,

$$B_{ij} = \begin{cases} \dfrac{\exp(-d(\hat{M}_i^t, \hat{M}_j^t)^2 / 2\sigma_i^2)}{\sum_l^N \exp(-d(\hat{M}_i^t, \hat{M}_l^t)^2 / 2\sigma_i^2)}, & if\ i \neq j \\ 0, & if\ i = j \end{cases} \tag{10}$$

where d is the distance function and σ_i^2 represents the variance of $\hat{M}_i^t$. σ_i^2 is determined by the adaptive method and controlled by the complexity factor h.

According to the probabilities of all the local models $\{P_1^t, P_2^t, \ldots, P_N^t\}$ and the predefined outlier threshold, the SOS algorithm outputs the labels that predict whether the local model is the poisoned model $\{l_1^t, l_2^t, \ldots, l_N^t\}$, where $l_i^t = 0$ if the local model is detected as the poisoned model; otherwise, it is 1. **Algorithm** 1 presents the proposed poisoned model detection method.

To conclude, the proposed model-level defensive mechanism rejects the poisoned intrusion detection models to join the global intrusion detection model aggregation, mitigating the negative effect of the poisoning attacks. The newly obtained global model can be expressed as,

$$M^t = \sum_{i=1}^{N} l_i^t c_i M_i^t \tag{11}$$

where l_i is the poisoned model prediction results, c_i is the proportion of the local training dataset D_i in the whole dataset.

Algorithm 1: Proposed poisoned model detection

Input: Local training datasets $\{D_1, D_2, \ldots, D_N\}$
Output: Poisoned model detection results $\{l_1^t, l_2^t, \ldots, l_N^t\}$

At the local clients:;
for i *in* N **do**
  Train with D_i to obtain the local model M_i^t;
  Calculate the importance factors Ω_i^t and its bitmask $Mi\_mask_i^t$ to mark the important model parameters by (4)–(7);
  Upload M_i^t and $Mi\_mask_i^t$ to the central server;
end
At central server:;
Aggregates all the local models' $Mi\_mask_i^t$ and obtain the global bitmask M_mask^t to mark the global important model parameters by (8);
Obtain the dimensionality-reduced local models $\{\hat{M}_1^t, \ldots, \hat{M}_N^t\}$ based on M_mask^t;
Input $\{\hat{M}_1^t, \ldots, \hat{M}_N^t\}$ to the SOS algorithm and output the poisoned model detection results;

4.2. Poisoned data detection for data-level defense

In this section, we propose the poisoned data detection method to identify the poisoned traffic samples that inject into the local training dataset at the poisoned client side. Current poisoned data detection methods [13–15] focus on the centralized learning system and require to directly access all the training data, however, due to the data privacy concerns, these methods are unsuitable for the FL-based NIDS scenario. Besides, since FL-based NIDS commonly faces with Non-IID data, each client may not contain the complete information of the clean traffic data, it is difficult for the poisoned client to distinguish the poisoned data and the clean data by only utilizing its own data. To cope with these issues, we transmit the surrogate data of the clean traffic data to the poisoned client, rather than transmitting the original data, which can help it achieve the poisoned data detection and preserve the data privacy. More specifically, we extract the class paths of the clean traffic data as the surrogate data. The class path is constituted with the critical neurons in deep neural network that make significant contributions toward the model decision of the traffic samples within the same class, shown in Fig. 2. It has been proved that the class path is conductive to distinguish the poisoned data and the clean data [17], as most of the adversarial perturbations are propagated and amplified along the critical neurons.

4.2.1. Class path extraction based on LRP

We apply the exiting Layer-wise Relevance Propagation (LRP) method [16] to extract the class paths of the traffic samples. LRP is used to traverse the whole deep neural network, and the decision values are redistributed based on the proportion of the contribution of each neuron in the layer. The layer-wise relevance propagation rule is simplified as,

$$R_i^l = \sum_j \frac{a_i^l f_{ij}}{\sum_{i'} a_{i'}^l f_{i'j}} R_j^{l+1} \tag{12}$$

where R_i^l is the relevance score of the ith neuron in the lth layer, f_{ij} denotes as the weight connected the ith neuron and the jth neuron, and a_i^l represents the activation of the ith neuron in the lth layer.

Firstly, we adopt LRP to extract the effective path of each traffic sample x. To be more specific, we calculate the relevance

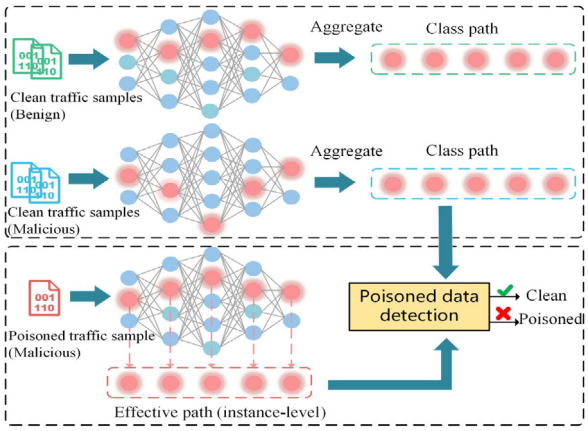


Fig. 2. Illustration of proposed poisoned data detection.

scores of all neurons R_x^l in the lth layer of the model by (12). Subsequently, we take the indexes of the TopK values of R_x^l as the critical neuron set in the lth layer for x, denoted as $\mathcal{N}_x^l$,

$$\mathcal{N}_x^l = TopK(R_x^l) \tag{13}$$

The critical neurons of all the layers constitute the effective path of traffic sample x, $\mathcal{N}_x = \{\mathcal{N}_x^1, \mathcal{N}_x^2, \ldots, \mathcal{N}_x^L\}$.

Secondly, considering that the effective paths of the samples within the same class typically show similar patterns, we aggregate the effective paths of the traffic samples within the same class to obtain the corresponding class path. More specifically, we introduce the bitmask $N\_mask_x^l$ to note whether the ith neuron in the lth layer of the traffic sample is the critical neuron,

$$N\_mask_x^{l,i} = \begin{cases} 1, & if\ i \in \mathcal{N}_x^l \\ 0, & else \end{cases} \tag{14}$$

By aggregating $N\_mask_x^l$ of the traffic samples in the same class and taking the TopK indexes, the class path in layer l can be obtained,

$$\mathcal{N}_c^l = TopK(\sum_{x \in x_c} N\_mask_x^l) \tag{15}$$

The class paths of all the layers constitute the class path for class c, $\mathcal{N}_c = \{\mathcal{N}_c^1, \mathcal{N}_c^2, \ldots, \mathcal{N}_c^L\}$.

Overall, according to the poisoned model detection results, we randomly select a small number of traffic samples from the local dataset of the clean clients, and calculate their effective path and then upload to the central server. The central server aggregates the uploaded paths to obtain all the class paths of the clean traffic samples $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_C\}$.

4.2.2. Poisoned data detection based on path similarity

The effective paths of the samples from the same class usually have high similarity with the corresponding class path [17]. Inspired by this fact, we propose the poisoned data detection method based on the path similarity. We first study the similarity between the effective path of the traffic sample x and its corresponding class path. The similarity between them can be calculated by the Jaccard coefficient,

$$J_x = J(\mathcal{N}_x, \mathcal{N}_c) = \frac{1}{L} \sum_{l=1}^{L} \frac{|\mathcal{N}_x^l \bigcap \mathcal{N}_c^l|}{|\mathcal{N}_x^l \bigcup \mathcal{N}_c^l|} \tag{16}$$

where $\mathcal{N}_x$ is the effective path of the traffic sample x and $\mathcal{N}_c$ denotes as the corresponding class path. $J(\cdot)$ is the Jaccard coefficient.

In each round, we calculate the similarity between the effective path of the clean traffic samples and its corresponding class path, and take the p percentile similarity as its threshold θ_c, where p is set to 95 in our experiments. The central server conveys the class paths of the clean traffic data and their corresponding thresholds to the poisoned clients. For the poisoned clients, we calculate the effective paths of all the samples, and then calculate the similarity between the path and the class path corresponding to their labels. If the similarity is lower than the predefined threshold, it is identified as poisoned samples. The implementation procedure of the proposed poisoned data detection method is summarized in **Algorithm** 2.

In conclusion, the proposed data-level defensive mechanism filters out the poisoned traffic data at the poisoned client, and retrains the local intrusion detection model with the remaining clean traffic data to rejoin in the global intrusion detection aggregation.

Algorithm 2: Proposed poisoned data detection

Input: Poisoned model detection results $\{l_1^t, \ldots, l_N^t\}$
Output: Poisoned data detection results
Acquire the clean clients $S_c = \{i | l_i^t = 1\}$ and the poisoned clients $S_p = \{j | l_j^t = 0\}$;
At the clean clients:;
for i *in* S_c **do**
  Randomly select a small subset of D_i as $\bar{D}_i$ $(|\bar{D}_i|/|D_i| = \eta\%)$;
  Obtain the effective paths$\{\mathcal{N}_x | x \in \bar{D}_i\}$ by (13);
  Upload $\{\mathcal{N}_x | x \in \bar{D}_i\}$ to the central server;
end
At central server:;
Obtain the class paths $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_C\}$ by (15);
Determine the detection thresholds $\{\theta_1, \theta_2, \ldots, \theta_C\}$;
Transmit the class paths and the thresholds to S_p;
At the poisoned clients:;
for j *in* S_p **do**
  Obtain the effective paths$\{\mathcal{N}_x | x \in D_j\}$ by (13);
  Calculate the path similarity J_x by (16);
  Compare J_x with θ_c to obtain the poisoned data detection results;
end

5. Experiments and analysis

5.1. Datasets and traffic domain constraints

We conduct all the experiments on UNSW-NB15 dataset and CICIDS2018 dataset. UNSW-NB15 dataset is often used to evaluate IoT intrusion detection system. It covers the most comprehensive attack scenarios, including the traffic data from nine network attacks and the benign type. Each row in the dataset has 49 features, containing multi-classification labels and binary classification labels. According to [27], to avoid the interference of the five-tuple information, we delete the related features and reserve 42 features for training the intrusion detection model. CICIDS2018 dataset collects the network traffic on a large test platform, covering 14 types of network attacks and the benign flows. It contains 79 packet-level and flow-level traffic features. Considering that some features contain too many zeros [28], we remove 11 features (Dst Por, Protocol, Timestamp, Flow Duration, Fwd URG Flags, Bwd URG Flags, CWE Flag Count, Fwd Byts/ B Avg, Fwd Pkts/ B Avg, Fwd Blk Rate Avg, Bwd Byts/ B Avg) to keep the dataset consistent and clean, remaining 68 features for intrusion

Table 1
The network attack types of UNSW-NB15 dataset and CICIDS2018 dataset.

Dataset	Traffic type	Number	Traffic type	Number
UNSW-NB15	Benign	2 218 761	Reconnaissance attack	13 987
	Generic attack	215 481	Analysis attack	2677
	Exploits attack	44 525	Backdoors attack	2329
	Fuzzers attack	24 246	Shellcode attack	1288
	DoS attack	16 353	Worms attack	174
CICIDS2018	Benign	13 390 249	Infilteration attack	160 639
	DDoS HOIC attack	686 012	DoS SlowHTTPTest attack	139 890
	DDoS LOIC HTTP attack	576 191	DoS GoldenEye attack	41 757
	DoS Hulk attack	461 912	DoS Slowloris attack	11 069
	Bot attack	286 191	DDoS LOIC UDP attack	1730
	FTP BruteForce attack	193 354	BruteForce Web attack	396
	SSH Bruteforce attack	187 589	BruteForce XSS attack	151
			SQL Injection attack	53

Table 2
The unmodified features of UNSW-NB15 dataset and CICIDS2018 dataset.

Dataset	Unmodified feature name
UNSW-NB15	Dbytes, dttl, dloss, Dload, Dpkts, dwin, dtcpb, dmeansz, Djit, Dintpkt, is_sm_ips_ports
CICIDS2018	Tot Bwd Pkts, Bwd Pkt Len Max, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Bwd IAT Tot, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Bwd PSH Flags, Bwd Header Len, Bwd Pkts/s, Bwd Seg Size Avg, Bwd Pkts/b Avg, Subflow Bwd Pkts, Subflow Bwd Byts, Init Bwd Win Byts

detection. Table 1 summarizes the number of each traffic type on the two datasets. As suggested in [29], we select the five most frequent categories as the experimental data on UNSW-NB15 dataset, including Benign, Generic, Exploits, Fuzzers, and DoS, and choose the seven frequent categories on CICIDS2018 dataset, including Benign, DDoS HOIC, DDoS LOIC HTTP, DoS Hulk, Bot, FTP BruteForce, and SSH Bruteforce. Similar to the data partitioning method in [4], each dataset is divided into two major parts: 70% of the dataset for the training part and the remaining 30% for the testing part. The training part is further divided into several partitions and delivered to each client for training the local intrusion detection model, and the testing part is to evaluate the global intrusion detection model. To mitigate the overfitting problem, dropout and regularization can be adopted in the model training [30].

When crafting the poisoned samples for the clean label attack, we take into account the traffic domain constraints that cannot modify some specific features, e.g., the features extracted from the backward traffic flow packets generated by the victim network are unable to be controlled over. Features that cannot be modified in UNSW-NB15 dataset and CICIDS2018 dataset are listed in Table 2.

5.2. Experimental settings

We consider two distributed data settings for FL-based NIDS: **IID data** and **Non-IID data**. For UNSW-NB15 dataset, we perform the anomaly detection task (binary classification) with IID data, since it is possible for each client to gather both benign and malicious traffic. With regard to CICIDS2018 dataset, we conduct the intrusion detection task (multi-class classification) on Non-IID data. Among it, each client owns different traffic types for local training, as the client is tough to collect the traffic data of all types of network attack.

For the attack model, we select 100 clients to learn the global network intrusion detection model during each round, among which 40 clients are malicious clients under poisoning attacks. We adopt the expanding poisoned model parameters strategy to maximize the success of poisoning attacks, as suggested in [11]. Two attack scenarios are considered:

Single round attack scenario: It indicates that poisoning attack is launched only in one round, e.g., injecting the poisoning attacks in the third round (denoted as $T_{attack} = 3$). T_{attack} is the attack round. In our experiments, we launch the poisoning attacks in round 3, 5, and 10, respectively. Generally, the third round is the early stage of federated training process, the tenth round represents the late stage wherein the global model tends to converge. The experiments can provide insights for investigating the defense performance when launching the poisoning attack in different stages.

Multiple round attack scenario: It means that the malicious clients launch poisoning attacks during multiple rounds. We consider continually launching the poisoning attacks since the first round on UNSW-NB15 dataset and since the fourth round on CICIDS2018 dataset.

In the FL-based NIDS, the local intrusion detection models are based on convolutional neural network. The CNN model adopts two 1D convolutional layers, each of which is followed by a 1D max-pooling layer, and the number of convolutional filters is 16 and 32. Two fully connected layers are then added for binary classification and multi-classification. The kernel size of each convolutional layer is fixed as 3, the stride is set as 1, the pooling size is 4, and the stride is 2. Additionally, as for the implementation of Algorithm 2, we set the proportion of the selected subset $\bar{D}_i$ in the whole local dataset D_i (η%) as 5% in our experiments. We implement the federated learning algorithm with the PyTorch framework and Python3.6, and the source codes about the implementations of the proposed SecFedNIDS are released online.[1]

5.3. Benchmarks

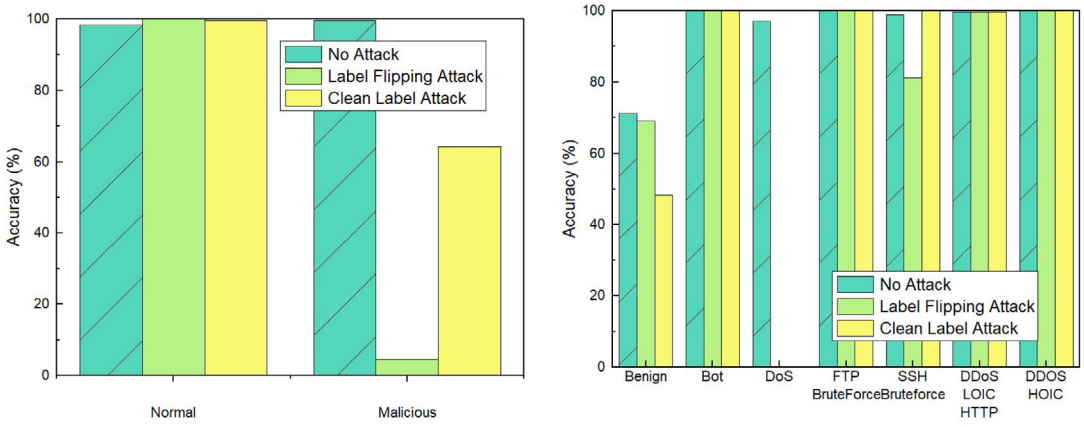
We compare the performance of the proposed defensive mechanisms with the following defense methods:

Krum [19] selects the local model parameters with the shortest distance from other N-M-2 clients as the global model' parameters. N is the number of all local models and M is the number of poisoned models.

Geomed [21] adopts the geometric median of all local model parameters as the global model' parameter.

VAE [9] relies on an offline detection model that employed in the server side to detect and reject the anomalous local models.

[1] https://github.com/zhangzhao156/SecFedNIDS.



(a) UNSW-NB15 Dataset (b) CICIDS2018 Dataset

Fig. 3. The accuracy of FL-based NIDS (with FedAvg) under no attack, the label flipping attack, and the clean label attack.

It requires to pretrain with pre-collected clean model parameters and is based on the random selection of model parameters.

The implementations of these methods are based on the descriptions in their papers. Note that these defense methods are also the first time to apply in FL-based NIDS.

5.4. Evaluation metrics

We define two evaluation metrics to assess the defensive performance of SecFedNIDS under poisoning attacks, namely, the accuracy and the target task accuracy.

Accuracy (Acc) refers to the overall accuracy of the global intrusion detection model on the test dataset.

Target class accuracy (Target Acc) is to quantify the accuracy of the target class. The target class is the category in which the attackers inject the modified traffic data. The poisoning attack that launched in our experiments is the targeted attack that aims to degrade the performance of the target class and not affect other classes. We choose the malicious traffic as the target class on UNSW-NB15 dataset and the Dos attack as the target class for CICIDS2018 dataset.

Besides, considering that the proposed poisoned model detection is essentially the binary classification task, **the poisoned model detection accuracy** is adopted to evaluate its effectiveness. As suggested in [14], we adopt **the poisons remove rate** and **the clean data remove rate** to evaluate the proposed poisoned data detection method. The poison remove rate is the recall rate of the poisoned data, and the clean data remove rate indicates the misjudgment of the clean data.

5.5. Results and discussion

5.5.1. Effect of poisoning attack on FL-based NIDS

To investigate the effect of the poisoning attacks against FL-based NIDS, we evaluate the performances of the global intrusion detection model obtained by FedAvg (the baseline federated learning algorithm) under three scenarios, including no attack (ideal scenario), launching the label flipping attack, and launching the clean label attack.

Firstly, we provide the detection accuracy of each traffic type considered in UNSW-NB15 dataset and CICIDS2018 dataset, under the three scenarios. For UNSW-NB15 dataset, we perform the anomaly detection task (binary classification) and launch the poisoning attacks targeted at the malicious traffic. Fig. 3(a) shows that under no attack scenario, the detection accuracies of the normal traffic and the malicious traffic obtained by FL-based NIDS are both close to 100%. However, the accuracy value of the malicious traffic has been drastically reduced to less than 10% by using the label flipping attack and to 64% with the clean label attack, while the accuracy of the normal is not affected. It indicates that FL-based NIDS under the poisoning attacks will misjudge the malicious traffic as the normal to a large extent. Similarly, for CICIDS2018 dataset, we perform an intrusion detection task (multi-classification), and Dos attack is selected as the target class. After launching the two types of poisoning attacks, the detection accuracies of Dos dropped to 0%, given in Fig. 3(b), indicating that the FL-based NIDS is unable to identify Dos attack. The substantial drops in accuracy confirm the negative effect of the poisoning attacks against FL-based NIDS once again.

Secondly, we list the overall accuracy of FedAvg on the test dataset under three scenarios on Tables 3 and 4. Results show that, under the label flipping attack and the clean label attack, the accuracy of FedAvg significantly decreases by up to 47% on UNSW-NB15 dataset and 30% on CICIDS2018 dataset. It also validates the poisoning attacks' effectiveness in degrading the performance of FL-based NIDS.

In summary, the poisoning attacks bring out the negative impact on the performance of FL-based NIDS and impair its protection capability. In this paper, we build a secure FL-based NIDS that is robust for the poisoning attacks, namely SecFedNIDS. Next, we present the systematic evaluations of SecFedNIDS with the proposed model-level defense and the proposed data-level defense respectively.

5.5.2. Defense performance of the proposed model-level defense mechanism

(1) Performance evaluation of SecFedNIDS under single round attack scenario. Tables 3 and 4 summarize the performances of SecFedNIDS with the proposed model-level defense mechanism when launching the poisoning attack in round 3, 5, and 10, respectively. Besides, we also compare it with the existing model-level defense methods, including the robust aggregation-based methods, Krum [19] and Geomed [21], and the detection-based method, VAE [9]. The compared results are also given in Tables 3 and 4.

First, it is apparent that, for both datasets, SecFedNIDS boosts the accuracies significantly compared with FedAvg under the two poisoning attacks, which achieves the equivalent performance as FedAvg in the ideal scenario (without poisoning attacks). Specifically, the accuracy improvement under the poisoning attacks is achieved up to 48% on UNSW-NB15 dataset and 36% on CICIDS2018 dataset. It is worth mentioning that the accuracy improvement of SecFedNIDS is attributed to the significant accuracy

Table 3
Performances of SecFedNIDS with the proposed model-level defense under the single-round poisoning attack on UNSW-NB15 dataset.

		NoAttack	Label flipping attack in T_{attack}					Clean label attack in T_{attack}				
		FedAvg (ideal)	FedAvg	Krum	Geomed	VAE	SecFedNIDS	FedAvg	Krum	Geomed	VAE	SecFedNIDS
$T_{attack} = 3$	Acc	97.15	50.85	52.01	67.58	50.00	**98.63**	52.5	51.78	**98.89**	98.88	98.88
	TargetAcc	95.79	1.73	4.07	35.34	0.00	**98.82**	6.69	5.96	**99.58**	99.54	99.54
$T_{attack} = 5$	Acc	98.11	51.48	92.55	79.14	50.00	**96.42**	76.65	92.81	**98.87**	**98.87**	**98.87**
	TargetAcc	98.02	2.98	86.35	58.65	0.00	**94.52**	67.42	87.75	**99.58**	**99.58**	**99.58**
$T_{attack} = 10$	Acc	98.98	52.27	98.99	91.16	50.00	**99.00**	81.92	82.00	**98.98**	85.75	**98.98**
	TargetAcc	88.53	4.57	99.39	83.54	0.00	**99.52**	64.19	64.34	99.52	72.19	**99.53**

Table 4
Performances of SecFedNIDS with the proposed model-level defense under the single-round poisoning attack on CICIDS2018 dataset.

		NoAttack	Label flipping attack in T_{attack}					Clean label attack in T_{attack}				
		FedAvg (ideal)	FedAvg	Krum	Geomed	VAE	SecFedNIDS	FedAvg	Krum	Geomed	VAE	SecFedNIDS
$T_{attack} = 3$	Acc	15.32	14.29	28.51	14.29	14.29	**28.51**	14.23	31.34	21.29	14.29	**36.86**
	TargetAcc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
$T_{attack} = 5$	Acc	81.32	51.81	58.75	73.67	66.91	**87.84**	72.17	71.99	67.39	65.41	**81.29**
	TargetAcc	97.06	0.00	0.00	0.00	0.00	**96.98**	0.00	0.00	0.00	0.00	**97.03**
$T_{attack} = 10$	Acc	95.25	78.53	74.04	81.29	74.13	**95.07**	78.28	81.54	81.41	81.23	**95.40**
	TargetAcc	97.04	0.00	0.00	0.00	0.00	**96.98**	0.00	0.00	0.00	0.00	**97.03**

improvement of target class. For example, it can be observed from Table 3 that, under the label flipping attack, the target task accuracy (the detection accuracy of the malicious traffic) of SecFedNIDS has increased by 91% to 97% compared with those of FedAvg. Likewise, for CICIDS2018 dataset, the target task accuracy (the detection accuracy of Dos) of SecFedNIDS has increased by up to 97% with $T_{attack} = 5$ and $T_{attack} = 10$, shown in Table 4. These results verify the defensive abilities of SecFedNIDS and indicate that the proposed model-level defense is robust for the poisoning attacks.

Second, the proposed model-level defense method achieves higher accuracies in most of cases than these benchmark defense methods, whether on UNSW-NB15 dataset or CICIDS2018 dataset. Besides, Table 3 shows that for UNSW-NB15 dataset, Krum and Geomed improve the accuracy values compared with FedAvg under the poisoning attacks, but Table 4 shows that for CICIDS2018 dataset, the accuracy improvement of Krum and Geomed compared with FedAvg is negligible, their target accuracies are still 0%. Note that we perform the anomaly detection task with IID data on UNSW-NB15 dataset and conduct the intrusion detection task with Non-IID data on CICIDS2018 dataset. These results confirm the protection of the robust aggregation-based defense methods against the poisoning attack on IID data and their limitation on Non-IID data.

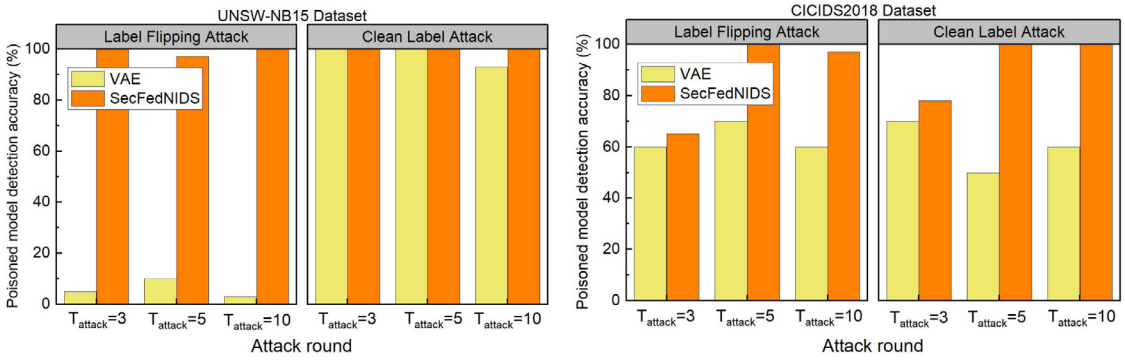
(2) Effectiveness evaluation of the proposed poisoned model detection method under single round attack scenario. Considering the proposed model-level defense mechanism is based on the proposed poisoned model detection method, we additionally study the effectiveness of the proposed poisoned model detection method under single round attack scenario. The proposed poisoned model detection method consists of the online unsupervised poisoned model detection method and the gradient-based important model parameter selection method, and thus we investigate the effect of the two components respectively.

Firstly, to evaluate the effectiveness of the proposed online unsupervised poisoned model detection method, we compare it versus with the existing offline detection-based method, VAE. The poisoned model detection accuracy of the proposed method and VAE are shown in Fig. 4. It is observed that the poisoned model detection accuracies of our method are higher than those of VAE on both datasets. Therefore, it is clear that our online detection-based method is capable of detecting poisoned model more accurately than the offline detection-based method.

Secondly, in order to investigate the effectiveness of the proposed gradient-based important model parameter selection method, we compare it with the PCA-based model parameter selection algorithm. Considering that PCA is an efficient dimensionality reduction method, the compared PCA-based method applies PCA to implement the model parameter selection and the other experimental settings keep consistent with our method. Experiments are carried out under the label flipping attack on UNSW-NB15 dataset. The poisoned model detection accuracies of the proposed method and the PCA-based method are given in Fig. 5. It can be found that the poisoned model detection accuracies of our method are superior than those of the PCA-based method. Especially, the poisoned model detection accuracies of our method reach 100%. The observable accuracy improvements reveal that the dimensionality-reduced model parameters obtained by the proposed gradient-based important model parameter selection method are more targeted and conducive to the poisoned model detection.

(3) Performance evaluation of SecFedNIDS under multiple round attack scenario. To further evaluate the performance of SecFedNIDS with the proposed model-level defense method, we consider the multi-round attack scenario that continually launches the poisoning attacks. From the results shown in Figs. 6 and 7, we can see several achievements. First, the accuracy curves of our method are above the curves of FedAvg, which indicates that SecFedNIDS with the proposed model-level defense method can improve its performance under the multiple rounds of poisoning attacks. Second, on the whole, our method outperforms the other model-level defense methods under the multiple rounds attack scenario, except for the results on Fig. 6(b). Third, different from the results on UNSW-NB15 dataset shown in Fig. 6, the accuracy curves of all the methods have fluctuations on CICIDS2018 dataset shown in Fig. 7. The reason is that our experiments adopt the IID data setting on UNSW-NB15 dataset and the Non-IID data settings on CICIDS2018 dataset. For Non-IID data, there exist the large model distribution shift among the local models, and the global model obtained after aggregation has certain instability. This fluctuation also aggravates the challenge of multi-round poisoning attack defense with Non-IID data.

(4) Effectiveness evaluation of the proposed poisoning model detection method under multiple round attack scenario. We also evaluate the effectiveness of the proposed poisoning model detection method under multiple round attack scenario, and provide the poisoning model detection accuracies of



(a) UNSW-NB15 Dataset (b) CICIDS2018 Dataset

Fig. 4. The poisoned model detection accuracies of SecFedNIDS (online detection-based) and VAE (offline detection-based).

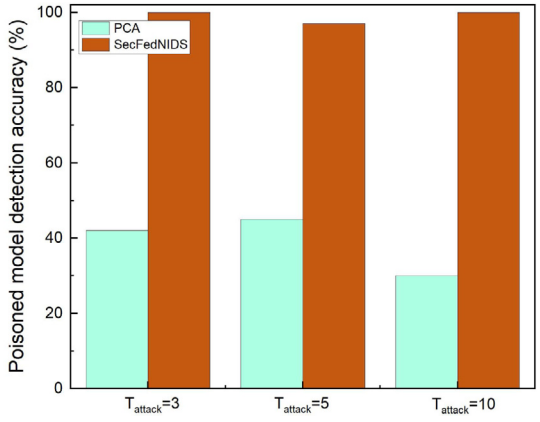


Fig. 5. The poisoned model detection accuracies of SecFedNIDS (gradient-based) and the PCA-based method.

our method and those of VAE. Experimental results under label flipping attack on UNSW-NB15 dataset and CICIDS2018 dataset are shown in Fig. 8(a) and (b), respectively. For the UNSW-NB15 dataset, the poisoning model detection accuracies are high, especially at 100% in the early stage. The poisoning model detection accuracies of VAE are very low, even below 50% in some attack rounds.

(5) Impact of the poisoned sample number on validity. To further analyze the proposed model-level defense method, we also study the impact of the poisoned sample number on the defense validity. Fig. 9 depicts that the performance of SecFedNIDS with different percentages of poisoned samples in the target class. Results demonstrate that SecFedNIDS can maintain high accuracies regardless of injecting a small percentage of poisoned samples or a large percentage of poisoned samples. On the one hand, the federated learning system is intrinsically robust to a few poisoned samples but cannot defend against extensive poisoned samples. As shown in Fig. 9, on UNSW-NB15 dataset, the accuracy of FedAvg (federated learning baseline method) does not change with less than 40% of the poisoned samples. Especially on CICIDS2018 dataset, the accuracy of FedAvg tends to drop significantly until the poisoned samples increased to more than 60%. It is worth mentioning that when a few poisoned samples are injected into the local training dataset, the generated poisoned models are usually indistinguishable from the clean models, but the harm of these poisoned models is easy to be neutralized by the remaining clean models, so that the performance of FL system is not impacted. On the other hand, the poisoned models trained with a large number of poisoned samples generally show the distinguishable patterns from the clean models, which ensures that the proposed poisoned model detection method is capable to detect and filter out these poisoned models effectively, and thus the proposed SecFedNIDS can still maintain high accuracies when encountering extensive poisoned samples.

5.5.3. Defense performance of the proposed data-level defense mechanism

The proposed data-level defense mechanism relies on the proposed poisoned data detection method, we first evaluate the feasibility and effectiveness of the proposed poisoned data detection method under the two poisoning attacks.

(1) Analysis of the class path similarity. As the proposed poisoned data detection method is based on the class path similarity, we analyze the class paths extracted by LRP to evaluate its feasibility.

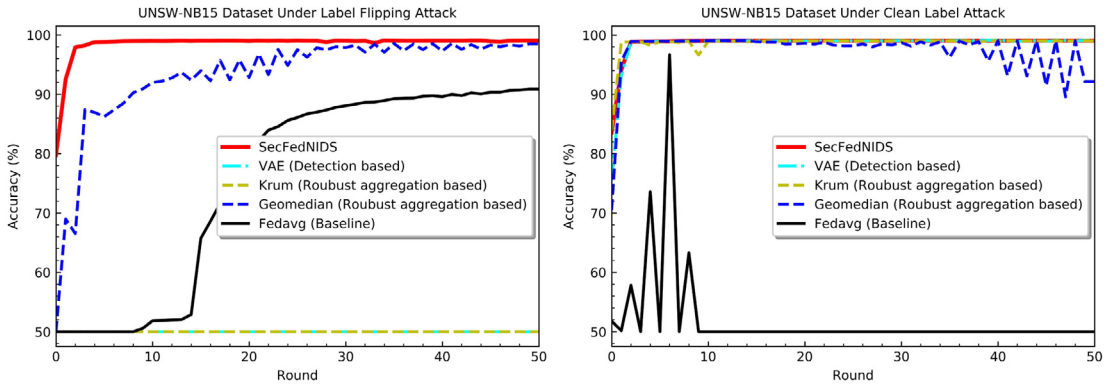
Firstly, we extract the effective paths of the clean traffic samples based on LRP, and present the instance-level path similarity. We randomly selected five instances from each traffic type in UNSW-NB15 dataset and CICIDS2018 dataset. The experimental results are shown in Fig. 10, the path similarities between the samples within the same class are fairly high, whereas the path similarities between the samples across different classes are low. From the observed results, it reveals that there exists the specific class path for the samples from the same class, thus the corresponding class path can be obtained by aggregating the effective paths of the samples within the same class.

Secondly, we calculate the path similarity between the effective paths of these traffic samples and the class paths that correspond to their labels, including the clean traffic samples and the poisoned traffic samples generated by label flipping attack (LFA) and lean label attack (CLA). From Fig. 11, both for the two poisoning attacks, the path similarity between the effective paths of the clean traffic samples and their corresponding class paths is high, while the path similarity of the poisoned traffic samples is relatively low. According to these observations, the similarity between the effective path of the sample and the class path corresponding to its label can be utilized to detect whether the traffic sample is the poisoned one. If the similarity is low, it is highly likely to be the poisoned traffic sample.

Overall, these results suggest that the class path extracted by LRP can provide a venue for the poisoned data detection.

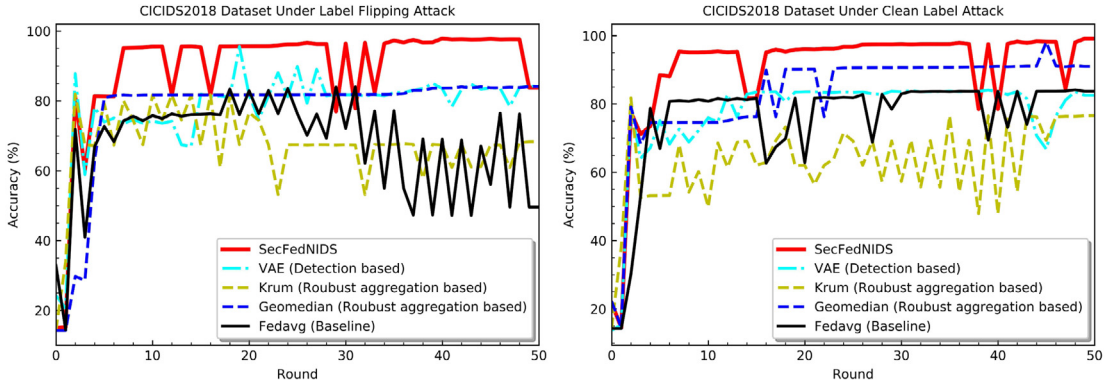
(2) Effectiveness evaluation of the proposed poisoned data detection method. We evaluate the performance of the proposed poisoning data detection method, through detecting the poisoned traffic samples generated by the label flipping attack and the clean label attack.

These results provided in Tables 5 and 6 are quite revealing in several ways. First, the proposed poisoned data detection method



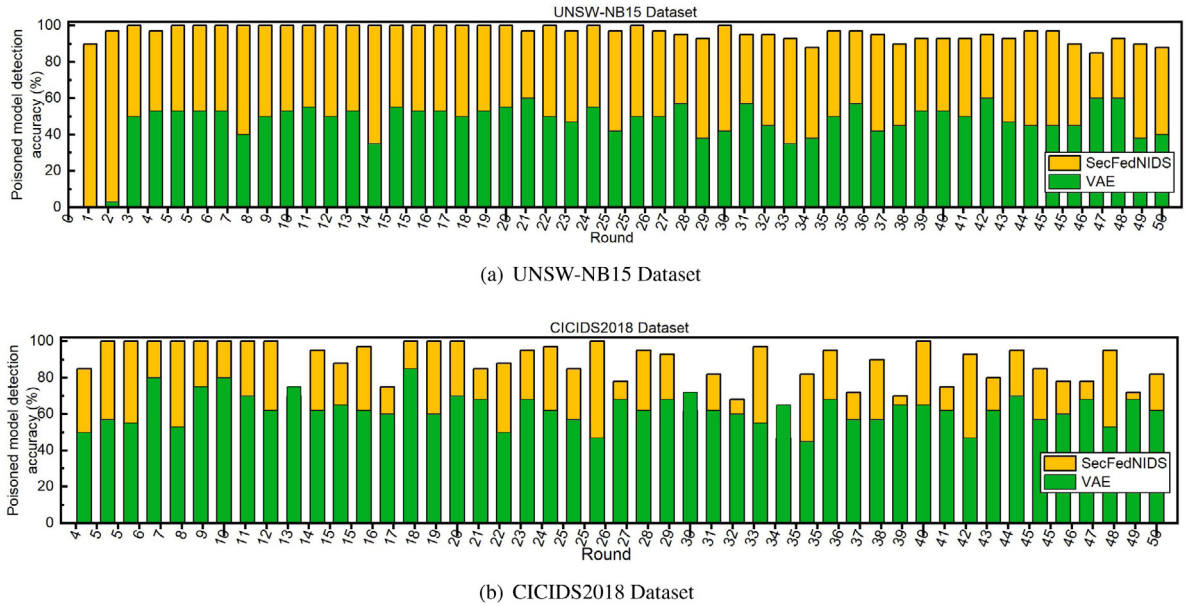
(a) Label flipping attack (b) Clean label attack

Fig. 6. Performances of SecFedNIDS with the proposed model-level defense under the multiple-rounds poisoning attack on UNSW-NB15 dataset.



(a) Label flipping attack (b) Clean label attack

Fig. 7. Performances of SecFedNIDS with the proposed model-level defense under the multiple-rounds poisoning attack on CICIDS2018 dataset.

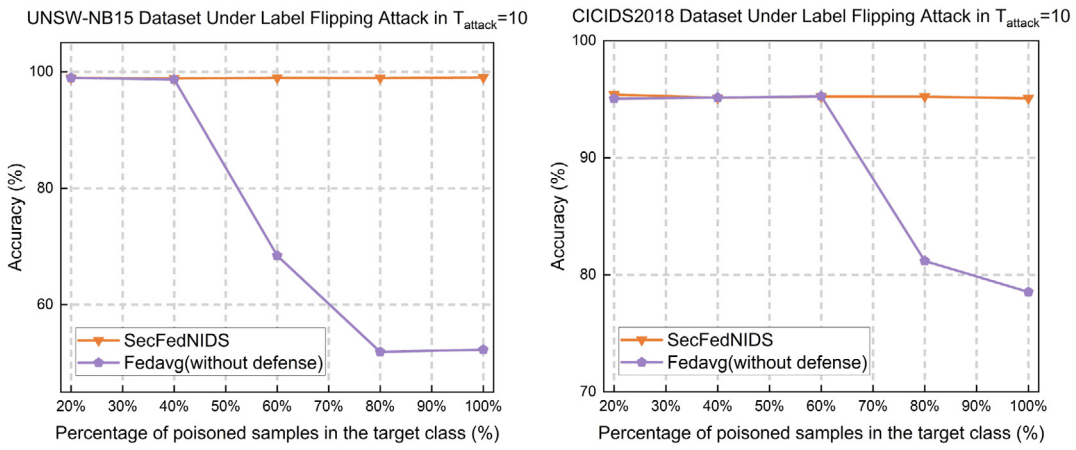


(a) UNSW-NB15 Dataset

(b) CICIDS2018 Dataset

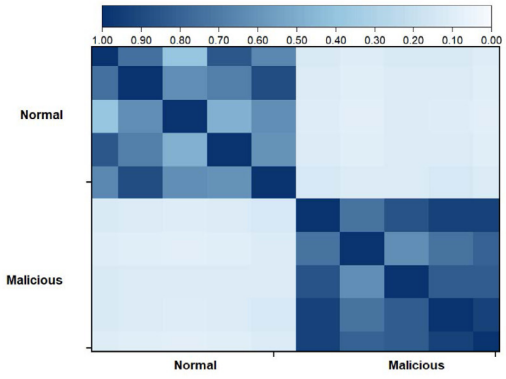
Fig. 8. The poisoned model detection accuracies of SecFedNIDS and VAE under the multiple-rounds poisoning attack.

can achieve high poisons remove rates and low clean data remove rate. Especially on CICIDS2018 dataset, the poisons remove rate can reach up to 100% and the clean data remove rate can reach lower than 1%. It indicates that all poisoned traffic samples can be identified and has low misjudgment on clean data, which also highlights the effectiveness of the proposed poisoned data detection method. Considering that the proposed poisoned data detection method is based on the threshold determined by the clean traffic samples, the clean traffic samples will inevitably be misjudged as the poisoned samples, but the low misjudgments

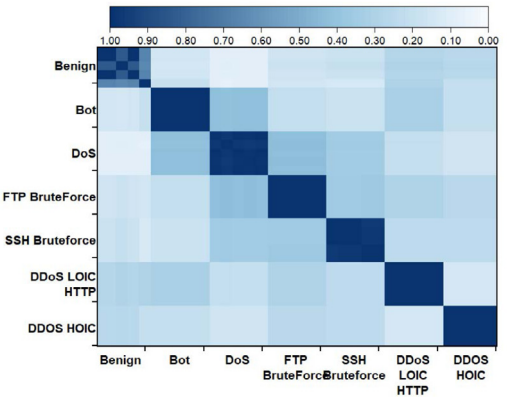


(a) UNSW-NB15 Dataset (b) CICIDS2018 Dataset

Fig. 9. Impact of the percentage of the poisoned samples on the performance of SecFedNIDS and FedAvg.

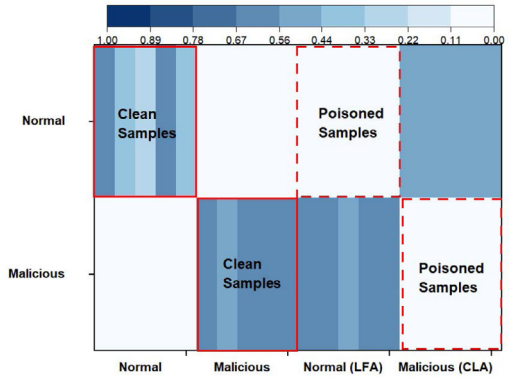


(a) UNSW-NB15 Dataset

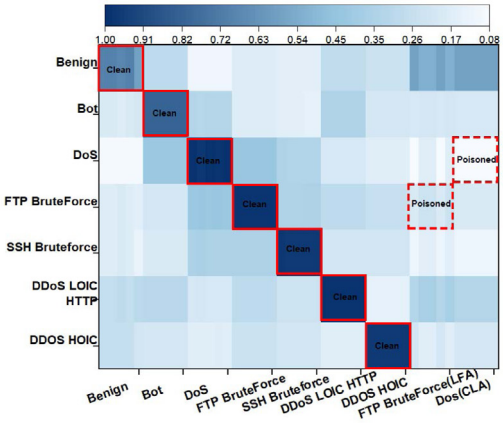


(b) CICIDS2018 Dataset

Fig. 10. The similarity analysis between the effective paths of the samples.



(a) UNSW-NB15 Dataset



(b) CICIDS2018 Dataset

Fig. 11. The similarity analysis between the effective paths of the samples and the class paths corresponds to their labels.

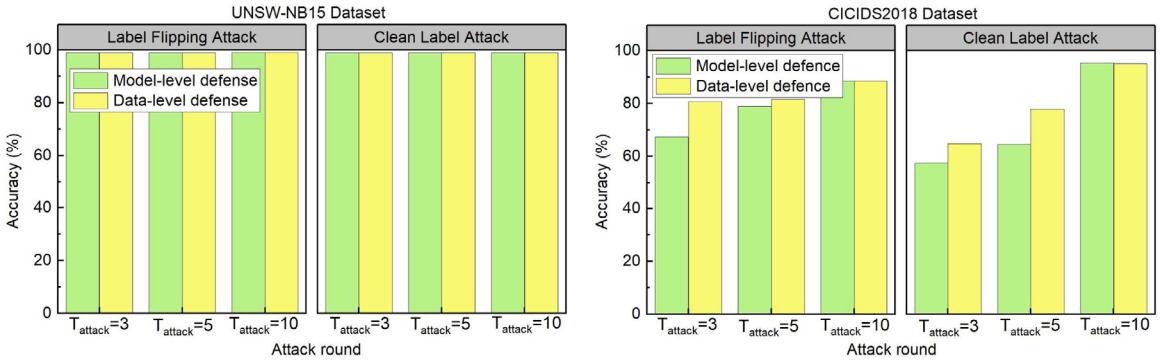
of the clean data are also acceptable in actual scenarios. Second, the results also demonstrate that the poisoned data rates have little influence on the effectiveness of the proposed poisoned data detection method. Even when the poisoned data account for 50% of the clean data, our method can still maintain high poisons remove rates. Third, from Table 6, the poisons remove rates are not impacted by the attack rounds on CICIDS2018 dataset. Lastly, it is remarkable that the proposed poisoned data detection method reaches higher detection accuracies for detecting the poisoned traffic sample generated by the clean label attack compared with those for detecting the label flipping attack on UNSW-NB15 dataset. This is because the clean label attack is launched by injecting the adversarial samples that aim to make the model misjudgment. The proposed class path can amplify the adversarial disturbances, resulting in better distinguishing between the effective paths of poisoned data and those of clean

Table 5
Performances of the proposed poisoned data detection method on UNSW-NB15 dataset.

Attack type	Poison rate	$T_{attack} = 3$		$T_{attack} = 5$		$T_{attack} = 10$	
		Poisons remove (%)	Clean data remove (%)	Poisons remove (%)	Clean data remove (%)	Poisons remove (%)	Clean data remove (%)
Label filipping attack	0.1	87.2	6.14	92.6	5.83	98.7	6.53
	0.2	87.4	6.42	97.7	6.52	97.2	6.11
	0.3	89.5	6.93	95.8	5.38	98.3	4.21
	0.4	86.5	4.57	98.4	4.66	98.5	6.77
	0.5	87.9	5.31	96.5	5.91	98.5	5.40
Clean label attack	0.1	100.0	4.67	94.0	6.45	100.0	6.08
	0.2	90.7	6.08	100.0	3.66	100.0	6.91
	0.3	99.1	5.06	90.7	5.88	100.0	4.53
	0.4	100.0	5.82	96.4	2.37	99.1	5.53
	0.5	97.5	5.28	100.0	4.96	100.0	4.71

Table 6
Performances of the proposed poisoned data detection method on CICIDS2018 dataset.

Attack type	Poison rate	$T_{attack} = 3$		$T_{attack} = 5$		$T_{attack} = 10$	
		Poisons remove (%)	Clean data remove (%)	Poisons remove (%)	Clean data remove (%)	Poisons remove (%)	Clean data remove (%)
Label flipping attack	0.1	100.0	8.04	100.0	1.63	100.0	0.96
	0.2	100.0	1.00	100.0	2.09	100.0	1.32
	0.3	100.0	1.15	100.0	1.08	100.0	1.50
	0.4	100.0	0.76	100.0	2.56	100.0	2.58
	0.5	100.0	1.84	100.0	1.42	100.0	1.51
Clean label attack	0.1	100.0	1.63	100.0	1.04	86.5	1.79
	0.2	100.0	0.85	100.0	1.49	100.0	1.56
	0.3	100.0	1.24	100.0	1.37	85.9	1.87
	0.4	100.0	2.37	100.0	1.51	86.1	0.61
	0.5	100.0	0.94	100.0	1.05	100.0	1.00



(a) UNSW-NB15 Dataset (b) CICIDS2018 Dataset

Fig. 12. Performances of the proposed model-level defense method and the proposed data-level defense method.

data, thereby the detection of clean data attack can be more efficient.

(3) Performance evaluation of SecFedNIDS with the proposed data-level defense. Fig. 12 reports the performances of SecFedNIDS with the proposed data-level defense on the two datasets. Meanwhile, in Fig. 12, we also compare it versus with the performance of SecFedNIDS with the proposed model-level defense, so as to validate the further improvement of the proposed data-level defense method on the proposed model-level defense method.

According to Fig. 12(a), SecFedNIDS with the proposed data-level defense has the equivalent performances as SecFedNIDS with the proposed model-level defense on UNSW-NB15 dataset. As we perform the anomaly detection (binary classification) task on UNSW-NB15 dataset with IID distributed data settings. The global model converge quickly in the early stage when trained with IID data, and thus directly discarding the poisoned client will not affect the performance of the global model. From Fig. 12(b), for both label flipping attack and clean label attack on CICIDS2018 dataset, the performance of SecFedNIDS with the proposed data-level defense is superior than that of SecFedNIDS with the proposed model-level defense when $T_{attack} = 3$ and $T_{attack} = 5$, and the accuracy improvement of the proposed data-level defense is achieved by up to 13%. This is because we perform the intrusion detection (multi-classification) task on CICIDS2018 dataset with Non-IID data. For Non-IID data, the model itself is not easy to convergence in the early stage. More training data joining in distributed training will assist to enhance the performance of the global model, thus the data-level defense mechanism poses an advantage over the model-level defense mechanism on Non-IID data.

6. Conclusion

In this work, we build a secure and robust FL-based NIDS called SecFedNIDS to defend against the poisoning attack. We first propose the poisoned model detection method for the model-level defense, including the online unsupervised poisoned model

detection method and the gradient-based important model parameter selection method. To further enhance the defensive performance, we propose the poisoned data detection method for the data-level defense. Specifically, we apply layer-wise relevance propagation to extract the class path and achieve the poisoned data detection based on path similarity. Our experiments verify the effectiveness of the proposed poisoned model detection method and the proposed poisoned data detection method in defending against the poisoning attacks.

CRediT authorship contribution statement

Zhao Zhang: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Yong Zhang:** Methodology, Writing – review & editing, Project administration, Funding acquisition. **Da Guo:** Writing – review & editing, Supervision. **Lei Yao:** Writing – review & editing, Visualization, Investigation. **Zhao Li:** Writing – review & editing, Investigation.

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.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61971057.

References

[1] B. McMahan, E. Moore, D. Ramage, et al., Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.
[2] Y. Fan, Y. Li, M. Zhan, et al., Iotdefender: A federated transfer learning intrusion detection framework for 5 g iot, in: 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), IEEE, 2020, pp. 88–95.
[3] Z. Ying, W. LiBao, C. JunJun, et al., Network anomaly detection based on federated learning, J. Beijing Univ. Chem. Technol. 48 (2) (2021) 92.
[4] B. Li, Y. Wu, J. Song, et al., DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems, IEEE Trans. Ind. Inf. 17 (8) (2020) 5615–5624.
[5] Z. Chen, N. Lv, P. Liu, et al., Intrusion detection for wireless edge networks based on federated learning, IEEE Access 8 (2020) 217463–217472.
[6] T.D. Nguyen, P. Rieger, M. Miettinen, et al., Poisoning attacks on federated learning-based IoT intrusion detection system, in: Proc. Workshop Decentralized IoT Syst. Secur.(DISS), 2020, pp. 1–7.
[7] M.J. Hashemi, G. Cusack, E. Keller, Towards evaluation of nidss in adversarial setting, in: Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, 2019, pp. 14–21.
[8] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 Ieee Symposium on Security and Privacy (Sp), IEEE, 2017, pp. 39–57.
[9] S. Li, Y. Cheng, W. Wang, et al., Learning to detect malicious clients for robust federated learning, 2020, arXiv preprint arXiv:2002.00211.
[10] C. Ma, J. Li, M. Ding, et al., Federated learning with unreliable clients: Performance analysis and mechanism design, IEEE Internet Things J. (2021).
[11] W. Liu, H. Lin, X. Wang, et al., D2MIF: A malicious model detection mechanism for federated learning empowered artificial intelligence of things, IEEE Internet Things J. (2021).
[12] R. Taheri, M. Shojafar, M. Alazab, et al., FED-IIoT: A robust federated malware detection architecture in industrial IoT, IEEE Trans. Ind. Inf. (2020).
[13] A. Paudice, L. Muñoz-González, E.C. Lupu, Label sanitization against label flipping poisoning attacks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Cham, 2018, pp. 5–15.
[14] N. Peri, N. Gupta, W.R. Huang, et al., Deep k-NN defense against clean-label data poisoning attacks, in: European Conference on Computer Vision, Springer, Cham, 2020, pp. 55–70.
[15] J. Chen, X. Zhang, R. Zhang, et al., De-pois: An attack-agnostic defense against data poisoning attacks, IEEE Trans. Inf. Forensics Secur. 16 (2021) 3412–3425.
[16] S. Bach, A. Binder, G. Montavon, et al., On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (7) (2015) e0130140.
[17] Y. Qiu, J. Leng, C. Guo, et al., Adversarial defense through network profiling based path extraction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4777–4786.
[18] P. Kairouz, H.B. McMahan, B. Avent, et al., Advances and open problems in federated learning, 2019, arXiv preprint arXiv:1912.04977.
[19] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, et al., Machine learning with adversaries: Byzantine tolerant gradient descent, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 118–128.
[20] C. Xie, O. Koyejo, I. Gupta, Generalized byzantine-tolerant sgd, 2018, arXiv preprint arXiv:1802.10116.
[21] Y. Chen, L. Su, J. Xu, Distributed statistical machine learning in adversarial settings: Byzantine gradient descent, in: Proceedings of the ACM on Measurement and Analysis of Computing Systems, Vol. 1, (2), 2017, pp. 1–25.
[22] D. Yin, Y. Chen, R. Kannan, et al., Byzantine-robust distributed learning: Towards optimal statistical rates, in: International Conference on Machine Learning, PMLR, 2018, pp. 5650–5659.
[23] V. Tolpegin, S. Truex, M.E. Gursoy, et al., Data poisoning attacks against federated learning systems, in: European Symposium on Research in Computer Security, Springer, Cham, 2020, pp. 480–501.
[24] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, PMLR, 2017, pp. 3987–3995.
[25] P. Molchanov, A. Mallya, S. Tyree, et al., Importance estimation for neural network pruning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11264–11272.
[26] J.H.M. Janssens, F. Huszár, E.O. Postma, et al., Stochastic Outlier Selection, techreport 2012-001, Tilburg centre for Creative Computing, 2012.
[27] S.M. Kasongo, Y. Sun, A deep learning method with wrapper based feature extraction for wireless intrusion detection system, Comput. Secur. 92 (2020) 101752.
[28] R. Faek, M. Al-Fawa'reh, M. Al-Fayoumi, Exposing bot attacks using machine learning and flow level analysis, in: International Conference on Data Science, E-Learning and Information Systems 2021, 2021, pp. 99–106.
[29] Z. Zhang, Y. Zhang, D. Guo, et al., A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks, Int. J. Mach. Learn. Cybern. 12 (6) (2021) 1649–1665.
[30] X. Ying, An overview of overfitting and its solutions, J. Phys.: Conf. Ser. 1168 (2) (2019) 022022, IOP Publishing.

Zhao Zhang is currently pursuing the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include network intrusion detection, network security, and deep learning. She has authored or coauthored more than 5 papers. Her current work focuses on federated learning on network intrusion detection.

Zhang Yong received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 2007. He is a Professor with the School of Electronic Engineering, BUPT. He is currently the Director of Fab.X Artificial Intelligence Research Center. He is the Deputy Head of the mobile internet service and platform working group, China communications standards association. He has authored or coauthored more than 80 papers and holds 30 granted China patents. His research interests include Artificial intelligence, wireless communication, and Internet of Things.

Guo Da received his Ph.D. degree in electrical engineering from Beijing University of Posts and Telecommunications, and he is currently a senior engineer at that institution. His research interests are in AI, mobile communications, P2P networks.

Yao Lei is working toward the master's degree from the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include machine learning and deep learning. His current work focuses on network traffic detection and prediction based on deep learning.

Zhao Li is working toward the master's degree from the Beijing University of Posts and Telecommunications, Beijing, China. His research interests is in deep learning and graph neural network. His current work focuses on deep learning on network traffic detection and prediction.