Federated Learning-Based IDS Against Poisoning Attacks

Mengfan Xu[1(✉)] and Xinghua Li[2]

[1] School of Computer Science, Shaanxi Normal University, Xi'an 710061, Shaanxi, China
[2] School of Cyber Engineering, Xidian University, Xi'an 710071, Shaanxi, China
xhli1@mail.xidian.edu.cn

Abstract. With the implementation of the General Data Protection Regulation (GDPR), the federated learning scheme has become a hot topic in the field of private computing. However, existing federated learning scheme can only encrypt the models to ensure the privacy of the data, but can not guarantee the correctness of the uploaded models, which will lead to a significant decrease in the detection performance of the global model. In this paper, we propose a federated learning-based intrusion detection scheme (IDS) against poisoning attacks. Specifically, we first design an anti-poisoning attacks algorithm based on the encryption model. Then we define the anti-attack strategy and objective function. To achieve high detection performance for the availability and concealment of attack, we introduce the poisoning rate into the objective function. The privacy preservation for local data sources also be provided while the IDS model based on knowledge sharing among islands is constructed. We leverage the Paillier public key cryptosystem to prevent data leakage for each entity. The results of security analysis show that our scheme can meet the security requirements of local data sources. In addition, the experiment results demonstrate that the proposed scheme can significantly improve the robustness of the detection model, and its accuracy rate can reach 83.11% even after being poisoned, which means the detection performance has not significantly decreased compared with non-poisoning attacks scheme.

Keywords: Federated learning · Privacy computing · Poisoning attacks · Intrusion detection system · Homomorphic encryption

1 Introduction

With the continuous improvement of network attack methods, the traditional single-point-based detection schemes have a serious overfitting issue on the detection model [3,20,29] because data exists as islands in different local clients which resulting in a small amount of data. Therefore, an intrusion detection system based on multi-source local data came into being. Gartner, a well-known

W. Shi et al. (Eds.): SPNCE 2021, LNICST 423, pp. 331–345, 2022.
https://doi.org/10.1007/978-3-030-96791-8_25

international security agency, proposed the Managed Detection and Response Service (MDR) in 2016[1]. This service aims to overcome the above-mentioned overfitting issue. By integrating multiple local data sources, security experts can find potential threats, and configure a rule base and security protection strategy [11,14,16]. There are lots of machine learning-based researches have focused on above issues. However, the General Data Protection Regulation (GDPR) is introduced by the European Union and effective on May 25, 2018, clearly stipulates a total ban on the utilization of automated model decisions [9], which means that simply "**Roughly**" exchanging cross-domain data to train machine learning models will be illegal. To solve this issue, Google proposed the federated learning in 2016 [17,18,22]. The federated learning process as shown in Fig. 1, each island uploads the encrypted model to the cloud. After the cloud model aggregates the encrypted model uploaded by each island to train a more powerful detection model, the aggregation model is returned to each island to update. Because the model is transmitted and trained before encryption, no party can obtain real data information, which solves the problem of privacy disclosure. This architecture builds a global model without violating data privacy regulations.

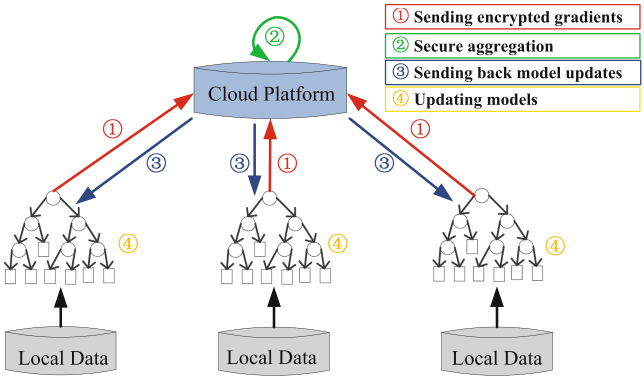


Fig. 1. Architecture for a federated learning system.

However, attackers in intrusion detection systems usually have background knowledge, including controlling models or training data and test data [7]. When the island is breached, the federated learning can only encrypt the model to ensure that the privacy of the data is not leaked. However, it cannot guarantee the correctness of the uploaded model, which makes it difficult to detect the poisoning attacks initiated during the latency phase, such as Stuxnet, OceanLotus, etc. [1,15,23]. To change the original distribution of the training data and causes the learning algorithm to change logically to threaten the target model [6,10,27], the attackers modify, delete, or inject malicious data on the training data of the target entity.

[1] https://www.paladion.net/buyers-guide-to-managed-detection-and-response.

To detect the above-mentioned poisoning attacks, researchers have done numerous researches. However, the existing anti-poisoning attacks schemes [21, 25,26,30] screen training data to improve the anti-attack capability in plaintext, which can not be used for federated learning. Specifically, if the island is breached and poisoned, as well as a large difference between the island data and the features of the verification set, the performance of the local encryption model (weak island model) will be reduced. Because the federated learning only uploads local encryption models, the cloud cannot determine whether local data has been modified, which leads the poisoning attacks cannot be detected. Furthermore, the current researches against poisoning attacks focused on reducing the availability of attacks, which is only considering the reduction of model errors, but ignoring the concealment of the attacks. It is impossible to accurately characterize actual attacks, such as Stuxnet, OceanLotus which have both availability and concealment features. Meanwhile, the local models are uploaded in federated learning further provides conditions for latency attacks. It is challenging to make the model take into account the availability and concealment of resistance to attacks.

In this paper, to address the above-mentioned issues, we propose a federal learning-based intrusion detection scheme against poisoning attacks called FLIDS, which can achieve secure data sharing between islands. In contrast, the model can effectively resist attacks. To summarize, the main contributions are as follows:

- A cryptographic model against poisoning attacks is designed. To avoid poisoning attacks in the island, the optimal local models are iteratively selected by calculating the performance residuals of the global model. Finally, a strong, robust intrusion detection model is aggregated. While ensuring the privacy-preserving for each local data source, the robustness of the global model against poisoning attacks is improved.
- An anti-poisoning attacks model is proposed based on federated learning. We first define the anti-attack strategy and the objective function. Furthermore, to detect the poisoning attacks with availability and concealment, we introduce the poisoning rate into the objective function.
- The experiments results demonstrate that the proposed scheme accuracy rate can still reach 83.11% with poisoning attacks, which is only a slight decrease compared with the detection performance without poisoning attacks.

The remaining parts of this paper are organized as follows. Section 2 shows the related work associated with our framework. Then, Sect. 3 presents some preliminary cryptographic background and Sect. 4 describes the problem formulations which include system model, threat model, and security goals. The concrete constructions of our schemes are demonstrated in Sect. 5. Later on, Sect. 6 shows the proof of security and performance analysis. Finally, the concluding remark of this whole paper is summarized in Sect. 7.

2 Related Work

2.1 Federated Learning

Nguyen et al. [13] construct federated learning on wireless networks as a convex structure optimization issue. They describe how mobile computing delay and delay affects UE energy consumption, system parameters between the learning time and learning precision balance. By all the closed forming solution of the sub-issue, the global optimal solution is achieved. Wang et al. [28] study the privacy leakage of federated learning and propose a general reconfiguration attack, which enables malicious servers not only to reconstruct actual training samples, but also to destroy the privacy of target clients. The proposed attack does not affect the standard training process and has obvious advantages over the existing attack mechanism. Brisimi et al. [2] propose a federated learning model, which can utilize the distribution between different data source of EHR data to predict future hospitalization in patients with heart disease. the proposed original dual division (cPDS) clustering algorithm can solve the issue of sparse support vector machine (SVM), which only using the small amount of features. It is beneficial to the interpretability of classification decision. Hu et al. [12] propose a new reasoning framework for urban environment perception Federated Reinforcement Learning (FRL), which inherited the basic idea of federated learning and utilized regional features in training process to improve reasoning accuracy.

However, the existing federated learning researches ensure that data privacy is not leaked through the encryption model, but can not guarantee the correctness of the upload model. Then the local data source is breached, it is difficult to detect the poisoning attacks, such as Stuxnet, OceanLotus, which is launched in the latent stage.

2.2 Poisoning Attacks

Saeed et al. [21] studied the antagonistic risk and robustness of classifiers, which are associated with metric sets in metric spaces. It is proved that any classifier with initial constant error is susceptible to antagonistic perturbation if the measurement probability space of the test instance is centralized. Suciu et al. [25] introduced a general framework for evaluating actual attacks against machine learning systems. They propose a targeted poisoning attacks which is designed to bypass existing defenses. The results show that this method is suitable for four classification tasks of three classifiers. Zhao et al. [30] transform the optimal poisoning attacks calculation issue in the multi-task relational learning model into a two-level programming which can adapt to the selection of arbitrary target task and attack task. They propose an effective algorithm for calculating the optimal attack strategy. The results show that the multi-task relational learning model is very sensitive to poisoning attacks, and the attacker can significantly degrade the performance of the target task by directly poisoning the target task. Matthew et al. [26] proposed an optimization framework specifically designed for linear regression and proved its effectiveness in a series of data sets and models.

In addition, a fast statistical attack is introduced, and a new anti-attack method is designed accordingly, which has strong flexibility against all poisoning attacks.

However, the existing researches on poisoning attacks improves the robustness of the system model by filtering plaintext data, so it is not suitable for federated learning based on encryption model. In addition, there is no research on the anti-poisoning attacks and anti-attack strategy for intrusion detection system.

3 Preliminaries

First, we introduce the cryptosystem based on additive homomorphism used in this scheme [24].

KeyGen: Given the security parameters, calculate the public key pk = (M, g) and the private key sk = λ, where the private key sk can be randomly divided into $sk^{(1)}$ and $sk^{(2)}$

$\text{Enc}_{pk}(m)$: Given plaintext m, use the public key pk to compute the encrypted data $[\![m]\!]$.

$\text{Dec}_{sk}([\![m]\!])$: given the ciphertext $[\![m]\!]$, use the key sk to decrypt the plaintext m.

$\text{SDec}_{sk^{(i)}}([\![m]\!])$: given the ciphertext, the partial key sk_i is used to calculate the partially decrypted ciphertext.

$[\![m]\!]^{(i)} = [\![m]\!]^{(sk^{(i)})} mod M^2$

WDec($\{[\![m]\!]^{(1)}, [\![m]\!]^{(2)}\}$): given a set of partially decrypted ciphertexts, the plaintext m is obtained.

Since this cryptosystem is based on additive homomorphism, it has the following properties:

$[\![m_a + m_b]\!] = [\![m_a]\!] \bullet [\![m_b]\!]$

$[\![-m]\!] = [\![m]\!]^{M-1}$

In addition, in order to ensure the effective operation of this scheme on ciphertext, this paper adopts the encrypted rational number comparison operation protocol proposed in reference [19]. The specific security calculation is as follows:

Here, we introduce security comparison (SCOM):

SCOM($[\![m_a]\!], [\![m_b]\!]$): Outputs a comparison result R to determine the size, if R = 0, $[\![m_a]\!] \geq [\![m_b]\!]$, otherwise, $[\![m_a]\!] < [\![m_b]\!]$.

4 Problem Formulation

In this section, the system model, threat model and anti-attack strategy are introduced respectively.

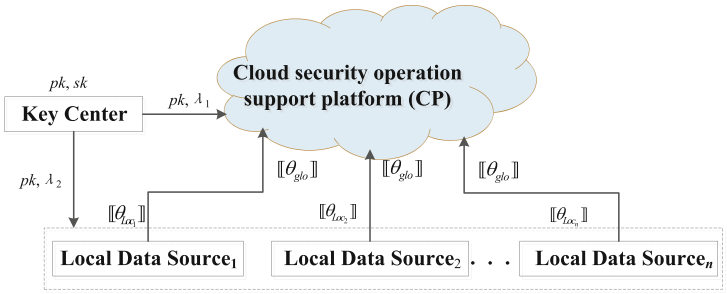


Fig. 2. System model.

4.1 System Model

As shown in Fig. 2, our system model includes n Local Data sources (LDS), Cloud platform (CP) and Key Distribution Center (KC).

KC: The trusted KC is responsible for the distribution and management of all keys in the system.

LDS: Each local data source has a local data set and is willing to contribute its detection model to build an accurate global detection model. Therefore, LDS should encrypt detection models trained on local datasets before sharing them with CP.

CP: It is a semi-honest cloud server with enough storage space to provide local data source security and can resist the global model building of poisoning attacks.

In order to initialize the password parameters in the scheme, the key of each domain is generated by a fully trusted key distribution center. The detailed process of key distribution is as follows:

- The key distribution center generates a key pair (pk, sk) and splits the key sk into λ_1 and λ_2.
- The key distribution center generates key pairs for the local data source (pk, λ_2).
- Key distribution center generates a key pair (pk, λ_1) for cloud server (CP).

If the ciphertext in this document is not specified, it indicates that the ciphertext is encrypted under the public key pk, for example: $[\![x]\!]$ represents $[\![x]\!]_{pk}$.

4.2 Threat Model

In this attack model, we assume that both CP and LDS are honest but curious entities that strictly follow predefined protocols but try to learn more data from other entities. Therefore, this paper introduces an attacker Adv with the following abilities:

- Availability attack. An attacker can generate malicious data to maximize the model's error rate or cause denial of service, making the model unavailable.
- Latency attack. In order to maintain continuous control or continuous access to useful information, attackers can conceal and steal information for a long time without being detected.

In addition, we assume that CP and any LDS are two independent semi-trusted entities that cannot collude.

5 Our Scheme

In this paper, an intrusion detection scheme against poisoning attacks based on federated learning is proposed to protect sensitive information in different networks and greatly improve the ability (robustness) of the global model against poisoning attacks. The overall process of the scheme is shown in Fig. 3. The scheme is mainly composed of attack policy and anti-attack policy. The local model of multi-source data is encrypted and uploaded to the cloud, and the global model is aggregated in ciphertext form in the cloud for local.

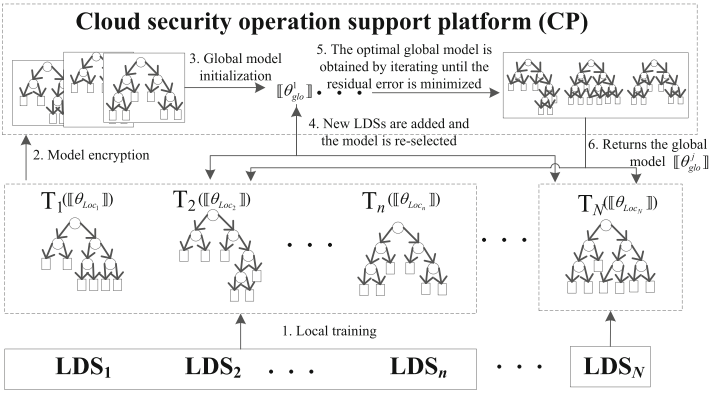


Fig. 3. The process of our scheme.

5.1 Attack Strategy

In order to fight against poisoning attacks more effectively, we first assume the attacker's ability and strategy to fit the actual attack behavior. Similar to existing studies [21,25,26,30], in this paper, we assume that the attacker has sufficient background knowledge of the attacked system, that is, the attacker can know training data D_{tr}, feature set F and model parameter θ. In addition, such as Stuxnet virus, Hailotus and other APT attacks, in the intrusion detection system, attackers need to be available and concealed at the same time in order to steal data or wait for an opportunity to damage the target network for a long time.

Firstly, the attacker hopes that the model after poisoning will be as different as possible from the original model. We define a function E to represent the available effect of local attacks, and its calculation formula is as follows:

$$E_{Loc} = ||\theta_{Loc_p} - \theta_{Loc}||_2^2 \tag{1}$$

The greater the E_{Loc}, the better the availability of local attacks.

Secondly, attackers prefer to hide attacks to avoid detection. We define a function C_{Loc} to represent the hiding effect of local attacks, and its calculation formula is as follows:

$$C_{Loc} = ||x_p - x_c||_2^2 \tag{2}$$

The smaller the C_{Loc}, the better the concealment of local attacks.

To sum up, we use W_{Loc} to represent the attacker's local objective function, whose strategy can be expressed as:

$$\arg\min_{x_p} W_{Loc} = \alpha C_{Loc}(x_p) - (1-\alpha) E_{Loc}(\theta_{Loc_p}) \tag{3}$$

$$s.t. \theta_{Loc_p} \in \arg\min_{\theta} L(D_{tr} \cup x_p, \theta) \tag{4}$$

It should be pointed out that the above problems are two-layer optimization problems [4]. The optimization of x_p in Eq. (1) is called the upper level problem, the optimization of θ_p in Eq. (3) is called the lower level problem, and L in Eq. (4) is the minimization of the objective function of the learning algorithm during training. In this paper, local poisoning rate is introduced to make the model fit the actual attack better. $\alpha = 0.5$ indicates that the attacker considers both hiding and effect of the attack, and $\alpha = 0$ indicates that the attacker only wants to execute effective attacks without considering concealment.

Similarly, we introduce the global poison rate beta, and the availability and concealment of global attacks are defined as Eqs. (5) and (6).

$$E_{glo} = ||\theta_{glo_p} - \theta_{glo}||_2^2 \tag{5}$$

$$C_{glo} = ||\theta_{Loc_p} - \theta_{Loc}||_2^2 \tag{6}$$

Combined with Eqs. (7) and (8), we use W_{glo} to represent the global objective function of the attacker, whose strategy can be expressed as:

$$\arg\min_{\theta_{Loc_p}} W_{glo} = \beta C_{glo}(\theta_{Loc_p}) - (1-\beta) E_{glo}(\theta_{glo_p}) \tag{7}$$

$$s.t. \theta_{glo_p} \in \arg\min_{\theta} L(\theta_{Loc} \cup \theta_{Loc_p}, \theta) \tag{8}$$

5.2 Anti-attack Strategy and Algorithm

In practical scenarios, it is basically safe to use original training data for intrusion detection model construction, and attackers are usually unable to manipulate it [5]. However, many intrusion detection systems require additional training data to update the model to enhance its adaptability, and this process provides an intrudable path for attackers [8]. Therefore, this paper assumes that the global model is updated every time a new data source is added. In order to secure the

sharing of source data, it is necessary to build a global model by aggregating benign encryption models uploaded from local data sources. In the real world, however, it is difficult for the cloud to directly distinguish between benign and malicious models.

To solve this problem, n local models with the best performance were selected from N local data sources and aggregated to obtain the global model $[\![\theta_{glo}]\!]$ and evaluate different local data iteratively until its detection performance remained constant. Evaluate different local data in combination with Eqs. (9), (10) and (11). After that, the P_i is encrypted and uploaded to the cloud.

$$P_i = Acc_i + DR_i \tag{9}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$DR = \frac{TP}{TP + FP} \tag{11}$$

where the larger P_i indicates, the better detection performance of the global model $[\![\theta_{glo}]\!]$ on the ith local data. $[\![RES_{P_i}]\!]$ is used to calculate the residuals of global model detection performance before and after iteration. Meanwhile, in order to deal with the attack proposed in Sect. 5.1, we define the following anti-attack target function:

$$min[\![RES_{P_i}]\!], RES_{P_i} = (\sum_{k=1}^{n} P_i - \sum_{k=1}^{n} P_{i-1})^2 \tag{12}$$

$$s.t.[\![P_i]\!] \in \{[\![P_1]\!], [\![P_2]\!], ..., [\![P_n]\!]\}, \min\{[\![P_1]\!], [\![P_2]\!], ..., [\![P_n]\!]\} > \max\{[\![P_{n+1}]\!], [\![P_{n+2}]\!], ..., [\![P_N]\!]\} \tag{13}$$

Similar to Eqs. (3) and (4), the above problems are two-layer optimization problems. The optimization of $[\![RES_{P_i}]\!]$ in Eq. (12) is called the upper level problem, and the optimization of P in Eq. (13) is called the lower level problem.

The global model is constructed iteratively, and the local model subset with the best performance in each iteration is aggregated. Assume that the number of original local models is n, and $\alpha * n$ is the number of poisoned data sources. The total number of local data sources in federated learning is $N = n + \alpha * n$. This paper assumes that $\alpha < 1$, ensure that most local data sources are not poisoned. Ideally, we need to identify all p poisoned models and aggregate the global model from the remaining n benign models. However, it is apparently that the true distribution of local training data is unknown, making it difficult to accurately distinguish between benign and poisoned models. To address this challenge, we aggregate global models and test them locally, iteratively selecting the n submodels with the best performance (These models may also include poisoned models, but only local models that are close to benign and do not significantly affect the global model).

The iterative algorithm adopted in this paper is based on alternating minimization or expectation maximization algorithms [4]. At the beginning of the iteration, we have the encrypted local model $\{[\![\theta_{Loc_1}]\!], [\![\theta_{Loc_2}]\!] ..., [\![\theta_{Loc_n}]\!]\}$ uploaded by n data sources, which is aggregated to get the global model$[\![\theta_{glo}]\!]$. When new data sources are added to the model and need to be updated, all local data sources are evaluated as the detection model by $[\![\theta_{glo}]\!]$, and n models with the highest performance are selected for reaggregation and performance residuals $[\![RES_P]\!]$ are calculated. When the residual converges to a minimum, the process terminates. At this point, it is considered that CP has aggregated the detection model with the best performance for use by local data sources.

In order to realize the safe calculation of residuals $[\![RES_P]\!]$, this paper first designs the safe maximum array operation based on SCOM security comparison, and selects the maximum n number from N numbers. The specific algorithm is as follows:

Algorithm 1: Secure Max Array

Input: $[\![P_1]\!], [\![P_2]\!], ..., [\![P_N]\!]$
Output: Max array $\{[\![P_1]\!], [\![P_2]\!], ..., [\![P_n]\!]\}$

```
1 Initialize encrypted set Set_p ;
2 for N = n to N − n do
3 |   max ← [[0]];
4 |   for i = 1 to N do
5 |   |   s ← SCOM([[P_i]], [[P_max]]) if s ← 0 then
6 |   |   |   max = [[P_i]];
7 |   |   Setp ← max;
8 return Set_p /*Set_p={[[P_1]], [[P_2]], ..., [[P_n]]} ;
```

Secondly, SecureRES security calculation is designed as follows:

SecureRES: LDS and CP calculate the performance of the global model on different local data sources through SCOM security operations. n data sources with maximum performance were selected by SMA security operation, and finally calculated.

6 Evaluation

This section evaluates the security and detection performance of the proposed scheme. First, we perform a security analysis of the proposed scheme to prove that the scheme can achieve the security objectives outlined in Sect. 4.2. Secondly, the experimental environment and data preprocessing process are introduced in detail. Then, the scheme is compared with the existing work. Finally, we analyze the influence of different parameters on the detection performance of the scheme.

6.1 Proof of Security

Now we present the proof of FLIDS security in the semi-honest model.

Theorem 1. *If Paillier public key cryptosystem is a semantically secure public key encryption scheme, then FLIDS is secure in the presence of a semi-honest adversary.*

Proof. We will prove the theorem by considering, in turn, the case where each of the parties has been corrupted. In each case, we invoke a simulator with the corresponding party s input and output. Our focus is in the case where party A wants to engage in the computation of the intersection. If party A does not want to proceed with the protocol, the views can be simulated in the same way up to the point where the execution stops.

Case 1. Corrupted CP. In this case, we show that we can construct a simulator Sim_{CP} that can produce a computationally indistinguishable view. In the real execution, the CP's view, $View_{CP}(\Lambda, M_N)$ is as follows:

$$\{\Lambda, r_{CP}, M_N, P_N, P_n, M_n\} \tag{14}$$

In the above view, r_{CP} is the outcome of internal random coins of the cloud. $M_N = \{m_i | i \in [1, N]\}$ is the set of N local submodels which are sent by the LDSs to the CP. $W_N = \{P_j | j \in [1, N]\}$ are calculated by formula (9), (10), (11) which are also sent to the CP. P_n is the output of Algorithm 1 and the SCOM and SMA are both proved using the ideal-real paradigm. The security proof of SCOM can be found in [19]. M_n is the first N valid set of local sub-models which is selected by index P_n.

To simulate this view, Sim_{CP} does the following: it creates an empty view and appends to it Λ and uniformly at random chosen coins r'_{CP}. N local sub-models are randomly selected and encrypted by the Paillier public key cryptosystem to form the local model set M'_N of LDSs. Then, taking the M'_N as the input of Algorithm 1 to generate simulated copy P'_N, P'_n, M'_n. Finally, the simulator appends P'_N, P'_n, M'_n to the view. Therefore, $Sim_{CP}(\Lambda, M_N) = \{\Lambda, r'_{CP}, M'_N, P'_N, P'_n, M'_n\}$.

We argue that the information sequences generated by simulation is computationally indistinguishable from the real view. The input parts are identical (i.e., both are Λ), the random coins are both uniformly random, and so they are indistinguishable. The element M'_N in Sim_{CP} is randomly selected and encrypted by using the Paillier public key cryptosystem, which is consistent with the element M_N in real view. P'_N, P'_n, M'_n are similar to M'_N which are ciphertext encrypted by the Paillier public key cryptosystem. In this paper, they rely on the assumption of the existence of a semantically secure additive homomorphic encryption scheme. Therefore, we construct a simulator Sim_{CP} that can produce a computationally indistinguishable view, i.e. $Sim_{CP}(\Lambda, M_N) = View_{CP}(\Lambda, M_N)$.

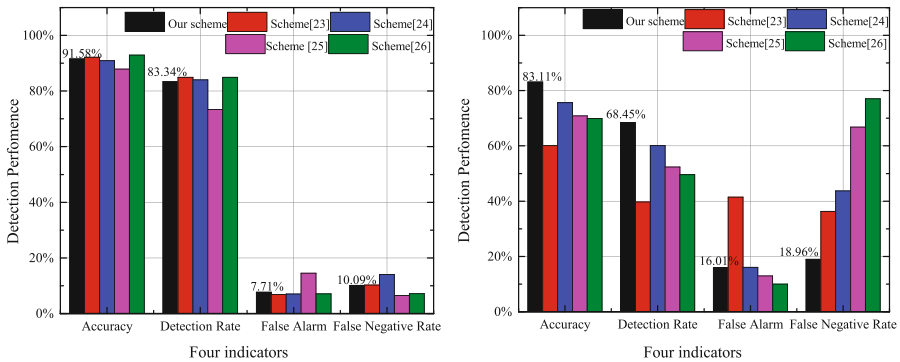
Case 2. Corrupted LDS. The proof process is similar to *Case 1.*

Combining the above, we conclude the algorithms are secure and complete our proof.

6.2 Experimental Environment

The CTU-13 data set published by Technical University of Prague, Czech Republic was used to evaluate the detection performance based on temporal association analysis. CTU-13 dataset includes 14 features such as *StartTime*, *Dur* and *Proto* and 1 category *Label* in CTU-13 data set. The dataset, released in 2014, contains 13 files with consecutive 7 days of network traffic data, with an average of 2.6 million pieces of data per file. The experimental environment was PC (i5-4590 main frequency 3.3 GHz, memory 4 GB, operating system Win7 64-bit), and the experimental tools were Java and Python 3.0. In addition, real data sets are used to verify the performance of the proposed scheme. In this scheme, K = 1024 bits is selected to achieve the 80-bit security level.

6.3 Analysis of Experimental Results



(a) *Without poisoned* (b) *Poisoned*

Fig. 4. Comparison of the effect by using different federated learning algorithms on detection performance.

First, this scheme was compared with schemes [2,12,13,28], and tested on CTU-13 data set. As shown in Fig. 4, the false alarm rate and false negative rate of this scheme are similar to those of existing algorithms when there is no poisoning attack. This is because this scheme uses integrated classifier with better detection performance to generate global model. Meanwhile, due to the limitation of training data set size, existing deep learn-based algorithms cannot achieve higher detection accuracy. When there is a poisoning attack on an existing federated learning scheme, the performance dropped substantially. This scheme can still ensure stable detection performance when being poisoned, this is because the security aggregation algorithm in this scheme design, use of multiple iterations local model to calculate the residual choose the highest performance, the greatest degree of reducing the poisoning model the impact on the global model. Among

them, the detection performance of this scheme has a slight decline, mainly because the global model fails to completely eliminate the influence brought by the poisoning model, resulting in the decline of the final detection accuracy.

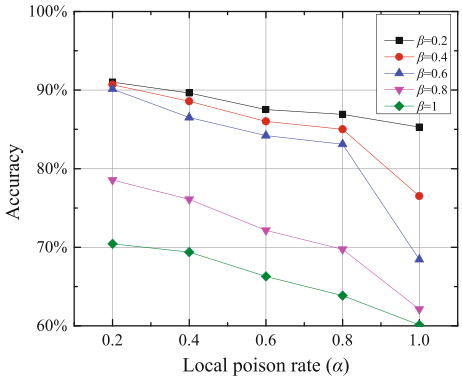


Fig. 5. The accuracy on different local poison rate and global poison rate.

With Sect. 5.1 attack strategy, can know different rate of local and global poisoning effect on the performance of the global model is different, as shown in Fig. 5, when $\alpha \leq 0.8$, $\beta \leq 0.6$, the global model testing accuracy stable at more than 83%, it shows that using this scheme the CP can iteratively select the optimal local model from LDS and aggregate it, when $\alpha = 1$, $\beta \geq 0.8$, the global model performance dropped substantially, This is because there are too many poisoned nodes at this time, and the local data is completely tampered, CP has been unable to obtain correct information from LDS. But at this time, it does not accord with the characteristics of concealment of actual attack. Therefore, $\alpha = 0.8$, $\beta = 0.6$ are selected in this paper, and it is considered that the robustness of the global model can be guaranteed when $\alpha \leq 0.8$, $\beta \leq 0.6$.

7 Conclusions

In this paper, we propose a federated learning-based intrusion detection scheme against poisoning attacks, called FLIDS, which achieves that secure data sharing between islands, ensures the privacy-preserving for each local data source, and improves the robustness of the global model against poisoning attacks. In addition, we design the resistance to poisoning attacks algorithm based on encryption model, and put forward a complete anti-attack model. The model defines the strategy and target function against the attack, while the poisoning rate is introduced to objective function. Then we make the model take into consideration with the availability and concealment in attack. The analysis showed that FLIDS is able to satisfy the proposed goals. Finally, we verify the validity and feasibility of the scheme. Experimental results show that the detection performance of this method on real data sets is significantly improved.

Acknowledgements. This work was supported by National Natural Science Foundation of China (Grant Nos. U1708262, U1736203).

References

1. Bohara, A., Noureddine, M.A., Fawaz, A., Sanders, W.H.: An unsupervised multi-detector approach for identifying malicious lateral movement. In: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), pp. 224–233. IEEE (2017)
2. Brisimi, T.S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I.C., Shi, W.: Federated learning of predictive models from federated electronic health records. Int. J. Med. Inform. **112**, 59–67 (2018)
3. Caruana, R., Lawrence, S., Giles, L.: Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping. Advances in Neural Information Processing Systems, pp. 402–408 (2001)
4. Csiszár, I.: Information geometry and alternating minimization procedures. Stat. Decis. **1**, 205–237 (1984)
5. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
6. Fu, Z., Huang, F., Ren, K., Weng, J., Wang, C.: Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data. IEEE Trans. Inf. Forensics Secur. **12**(8), 1874–1884 (2017)
7. Gajewski, M., Batalla, J.M., Mastorakis, G., Mavromoustakis, C.X.: A distributed ids architecture model for smart home systems. Clust. Comput. **22**(1), 1739–1749 (2019)
8. Gozde Bakirli, D.B.: DTreeSim: a new approach to compute decision tree similarity using re-mining. Turk. J. Electr. Eng. Comput. Sci. **25**, 108–125 (2017)
9. Greengard, S.: Weighing the impact of GDPR. Commun. ACM **61**(11), 16–18 (2018)
10. Grinshpoun, T., Tassa, T., Levit, V., Zivan, R.: Privacy preserving region optimal algorithms for symmetric and asymmetric DCOPs. Artif. Intell. **266**, 27–50 (2019)
11. Hermessi, H., Mourali, O., Zagrouba, E.: Deep feature learning for soft tissue sarcoma classification in MR images via transfer learning. Expert Syst. Appl. **120**, 116–127 (2019)
12. Hu, B., Gao, Y., Liu, L., Ma, H.: Federated region-learning: an edge computing based framework for urban environment sensing. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7. IEEE (2018)
13. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 19–35. IEEE (2018)
14. Jeong, G., Kim, H.Y.: Improving financial trading decisions using deep Q-learning: predicting the number of shares, action strategies, and transfer learning. Expert Syst. Appl. **117**, 125–138 (2019)
15. John, J.T.: State of the art analysis of defense techniques against advanced persistent threats. Future Internet (FI) and Innovative Internet Technologies and Mobile Communication (IITM) Focal Topic: Advanced Persistent Threats 63 (2017)
16. Joy, T.T., Rana, S., Gupta, S., Venkatesh, S.: A flexible transfer learning framework for Bayesian optimization with convergence guarantee. Expert Syst. Appl. **115**, 656–672 (2019)

17. Konečnỳ, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527 (2016)
18. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
19. Liu, X., Choo, K.K.R., Deng, R.H., Lu, R., Weng, J.: Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans. Dependable Secur. Comput. **15**(1), 27–39 (2016)
20. Luo, D., Ding, C., Huang, H.: Linear discriminant analysis: new formulations and overfit analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 25 (2011)
21. Mahloujifar, S., Diochnos, D.I., Mahmoody, M.: The curse of concentration in robust learning: evasion and poisoning attacks from concentration of measure. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4536–4543 (2019)
22. McMahan, H.B., Moore, E., Ramage, D., Arcas, B.A.: Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629 (2016)
23. Oprea, A., Li, Z., Yen, T.F., Chin, S.H., Alrwais, S.: Detection of early-stage enterprise infection by mining large-scale log data. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 45–56. IEEE (2015)
24. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
25. Suciu, O., Marginean, R., Kaya, Y., Daume III, H., Dumitras, T.: When does machine learning {FAIL}? Generalized transferability for evasion and poisoning attacks. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp. 1299–1316 (2018)
26. Van Opbroek, A., Achterberg, H.C., Vernooij, M.W., De Bruijne, M.: Transfer learning for image segmentation by combining image weighting and kernel learning. IEEE Trans. Med. Imaging **38**(1), 213–224 (2018)
27. Viejo, A., Sánchez, D.: Secure and privacy-preserving orchestration and delivery of fog-enabled IoT services. Ad Hoc Netw. **82**, 113–125 (2019)
28. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: user-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp. 2512–2520. IEEE (2019)
29. Xie, S., Gao, J., Fan, W., Turaga, D., Yu, P.S.: Class-distribution regularized consensus maximization for alleviating overfitting in model combination. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 303–312 (2014)
30. Zhao, M., An, B., Yu, Y., Liu, S., Pan, S.J.: Data poisoning attacks on multi-task relationship learning. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)