PDGAN: A Novel Poisoning Defense Method in Federated Learning Using Generative Adversarial Network

Chapter · January 2020

DOI: 10.1007/978-3-030-38991-8_39

Ying Zhao
Beijing University of Chemical Technology
70 PUBLICATIONS **475** CITATIONS
SEE PROFILE

Jiale Zhang
Nanjing University of Aeronautics & Astronautics
25 PUBLICATIONS **183** CITATIONS
SEE PROFILE

Di Wu
Deakin University
16 PUBLICATIONS **170** CITATIONS
SEE PROFILE

PDGAN: A Novel Poisoning Defense Method in Federated Learning Using Generative Adversarial Network

Ying Zhao[1], Junjun Chen[1](✉), Jiale Zhang[2], Di Wu[3,4], Jian Teng[1], and Shui Yu[3]

[1] College of Information Science and Technology,
Beijing University of Chemical Technology, Beijing 100029, China
{zhaoy,chenjj,tengj}@buct.edu.cn

[2] College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
jlzhang@nuaa.edu.cn

[3] School of Computer Science, University of Technology Sydney,
Sydney, NSW 2007, Australia

[4] Centre for Artificial Intelligence, University of Technology Sydney,
Sydney, NSW 2007, Australia
{Di.Wu,Shui.Yu}@uts.edu.au

Abstract. Federated learning can complete an enormous training task efficiently by inviting participants to train a deep learning model collaboratively, and the user privacy will be well preserved for the users only upload model parameters to the centralized server. However, the attackers can initiate poisoning attacks by uploading malicious updates in federated learning. Therefore, the accuracy of the global model will be impacted significantly after the attack. To address this vulnerability, we propose a novel poisoning defense generative adversarial network (PDGAN) to defend the poising attack. The PDGAN can reconstruct training data from model updates and audit the accuracy for each participant model by using the generated data. Precisely, the participant whose accuracy is lower than a predefined threshold will be identified as an attacker and model parameters of the attacker will be removed from the training procedure in this iteration. Experiments conducted on MNIST and Fashion-MNIST datasets demonstrate that our approach can indeed defend the poisoning attacks in federated learning.

Keywords: Federated learning · Poisoning defense · Generative adversarial network

1 Introduction

The traditional machine learning architecture [1] provides intelligent data analysis and automatic decision making that is known as machine-learning-as-a-service

S. Wen et al. (Eds.): ICA3PP 2019, LNCS 11944, pp. 595–609, 2020.
https://doi.org/10.1007/978-3-030-38991-8_39

[2], such as recommendation systems, keyboard input prediction, smart transportation, and health monitoring. However, such learning scenario requires its participants to outsource their private raw data to an unknown third party, causing a significant privacy concern where the sensitive data may be exposed to attackers [3]. Considering the above privacy issues, *federated learning* [4,5] has been explored recently, which has the natural ability to preserve the user data privacy by its unique distributed machine learning framework. The federated learning framework trains a global model across multiple participants in a distributed manner. Each participant can download the global model from the central server and train the model on their training datasets locally instead of outsourcing the sensitive training data in the traditional centralized training methods. Participants in federated learning only need to upload the model parameters (i.e., parameters of gradients and weights) generated from local training procedure, which provides a basic privacy guarantee. After receiving all the participant model updates, the central server will execute the federated average algorithm to update the global model further. The processes mentioned above will be performed iteratively until the global model tends to convergence.

However, the unique training scheme of the federated learning could be a double edge sword, where the central server cannot access participants private training data. This characteristic may be leveraged by the attackers to launch attacks such as poisoning attack. Poisoning attack [6] is a common attack method in the traditional centralized distributed system. The attackers in the poisoning attack [7,8] can change the learning model parameters by compromising partiality training data from other participants. Label-flipping [9] is a classic method to launch the poisoning attack, and it also can poison the federated learning. It requires the attacker to change the parameters of the target learning model in the training phase, and the poisoned model will be marked with some attacker targeted attributes. Then, the poisoned model will misclassify (take a classification task as an example) the attacker chosen inputs at the prediction stage. In federated learning, we notice that the poisoning attack can be easily initiated due to the following reasons. (1) Federated learning algorithm does not contain any authentication or identity verification mechanisms, so the trustworthy of participants cannot be guaranteed. (2) The participants' training data and training procedure are invisible to the server, so it is impossible to audit the accuracy of users' models from their updates. (3) Since the distributions of participants' training datasets are independent, which brings enormous difficulties for the anomaly detection of user's updates.

In this paper, we focus on the poisoning attack launched by the malicious participants in federated learning and try to defend such active attacks. To defend poison attacks, we deployed a generative adversarial network in the server to reconstruct user local training data and audit accuracy for each participant model using the generated data. The participant whose accuracy is lower than a predefined threshold is identified as the attacker.

1.1 Related Work

Known poisoning attack defense methods in centralized learning scenarios have been well explored. Secure multiparty computation and homomorphic additive cryptosystem [10–12] are two efficient tools to build training models while protecting training data privacy. However, these schemes introduce huge computation overhead to the participants and may bring a negative effect on model accuracy. Byzantine-tolerant machine learning methods [13,14] have been explored recently to guarantee the privacy of only Byzantine participants, which imitating the applicant in the scenario of federated learning. Besides, anomaly detection techniques have shown the significant advantages of detecting abnormal participants' behaviors. Aiming at detecting poisoning attacks in distributed learning models, mechanisms in [9,15,16] apply several algorithms (e.g., k-means and clustering) to check the participants' updates across communication rounds and remove the outliers.

Furthermore, some other defense methods [17,18], such as cosine similarity and gradients norm detection, were proposed to detect the gradients anomalies. However, the effectiveness of above-mentioned anomaly detection methods is quite low in the context of federated learning. That is mainly because the distributions of participants' training data in federated learning are considered as Non-IID (not independent and identically distributed), which means the participants' model updates are obviously different from each other.

1.2 Our Contributions

Aiming at solving the above problems, we propose a novel poisoning defense method. Briefly speaking, the contributions in our paper are threefold.

- We propose a detection scheme based on accuracy auditing, named poisoning defense generative adversarial network (PDGAN), to defend poisoning attacks in federated learning. The proposed scheme can be easily deployed in the real scenario.
- The PDGAN can use partial classes data to reconstruct the prototypical samples of participants' training data for auditing the accuracy of each participant's model.
- Experiments conducted on MNIST and Fashion-MNIST datasets demonstrate that our approach can indeed defend the poisoning attacks in federated learning.

1.3 Organization

The rest of this paper is organized as follows. In Sect. 2, we briefly introduce the basic knowledge of federated learning and generative adversarial nets. The overview of the poisoning attack in federated learning is presented in Sect. 3, and the construction of the proposed defense algorithm is detailed in Sect. 4. Extensive experimental evaluation is conducted in Sect. 5. Finally, Sect. 6 gives the conclusion and future work.

2 Preliminaries

In this section, we briefly review the preliminary knowledge of federated learning and the introduction of Generative adversarial nets (GAN) to facilitate understanding of our defense mechanism.

2.1 Federated Learning

Federated learning [4] is a centralized training framework which can preserve user privacy by its unique distribution learning mechanism. Unlike other collaborative learning methods, the participants in federated learning upload the model updates which generated by training learning model on participant private training data to the central server, and the central server will distribute the global models which share the same structure with participants models. In federated learning, all participants share the same learning objective and model structure, where the central server sends the current global model parameters to the selected participants m_t in each communication round t. Then, all the selected participants update their model and apply the model to train the local data. Each participant uploads the model updates after the local training procedure, where the uploaded model updates will be averaged and accumulated to the current central model. Equation 1 shows the updating procedure on the central model.

$$M_{t+1} = M_t + \frac{1}{m_t} \sum_{k=1}^{m_t} u_t^k, \tag{1}$$

where M_t is the current global model at the t-th iteration, and u_t^k represents the model updates uploaded by the k-th participant. A federated learning framework can achieve high satisfaction when the participants download the same model with the same initialization, which is averaged by the central model with all the valid uploads.

2.2 Generative Adversarial Nets

Generative adversarial networks [19] have shown the promised performance in computer vision as well as other research areas [20], which can generate high-quality fake images by training on the original images. The generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$) of GAN play an adversarial game, where the $\mathcal{G}$ firstly generates and $\mathcal{D}$ discriminate if the image is from the generator or original image set where the output of $\mathcal{D}$ can be represented as fake (0) or real (1). The performance of both $\mathcal{G}$ and $\mathcal{D}$ can be improved by several training epochs. Equation 2 shows the training target of GAN.

$$\begin{aligned} \min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim p_{data}(x)}[\log \mathcal{D}(x)] \\ + \mathbb{E}_{z \sim p_z(z)}[\log(1 - \mathcal{D}(\mathcal{G}(z)))], \end{aligned} \tag{2}$$

where $x \sim p_{data}(x)$ represents the original data distribution and $z \sim p_z(z)$ is the distribution of the random vector z. $\mathcal{D}$ and $\mathcal{G}$ will be trained by several epochs until the training procedure achieves the Nash equilibrium, while the $\mathcal{D}$ cannot discriminate the fake data from the real data.

3 Overview of Poisoning Attacks

In this section, we first introduce the threat model, including the learning scenario, the attacker's goals, and the attacker's capabilities. Then, we detailed discuss the poising attacks against federated learning and demonstrate the effectiveness of poisoning attacks through experimental evaluations.

3.1 Threat Model

Learning Scenario: As described in Sect. 2.1, we consider a federated learning scenario where multiples participants agree on a common learning objective and jointly train a global model on their localized training datasets. Besides, we assume that the distributions of all the participants' datasets are independent with each other, and the participant model updates will be averaged on the server side to achieve federated learning property. Without loss of generality, the main purpose of training a global model is to perform an image classification task in this paper.

Attacker's Goal: The attacker in the poisoning attack wants to conduct a specific global model that performs high accuracy on his chosen inputs while having less impact on overall accuracy. The attacker replaces the global model with a poison model that has the following two properties:

- Poisoning accuracy: the global model should behave good performance on the attacker-chosen poisoned inputs after the attacker's model updates were uploaded to the central server.
- Overall accuracy: the poisoned updates should have a less negative impact on overall accuracy, which means the poisoned global model cannot be discarded due to the attack behaviors.

Attacker's Capability: In a federated learning scenario, the attacker first pretends to be an honest user to participate in the learning system, while the main purpose of this attacker is to compromise the global model. Specifically, the attacker has the following capabilities:

- Knowledgeable: the attacker has a white-box access privilege to the global model, meaning that the model structure and parameters can be obtained.
- Active: the attacker is considered as an active insider because he can fully control the local training procedure and modify the model hyper parameters (e,g., epochs, learning rate).

3.2 Poisoning Attacks

Existing literatures [6,9,15] demonstrate that the poisoning attack can be easily launched by malicious participants in the context of federated learning. Here, we give a brief introduction about this poisoning attack and show how one attacker can successfully compromise the global model.

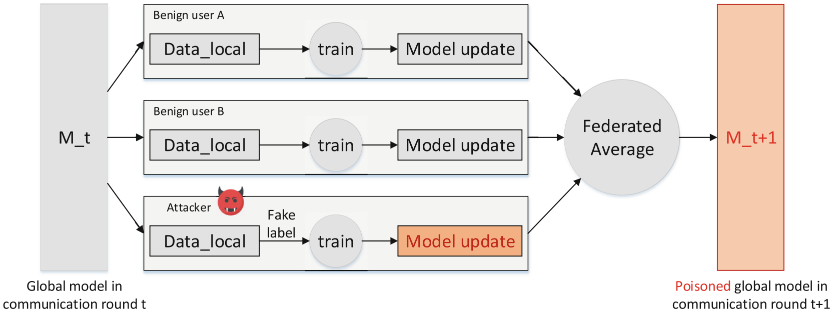
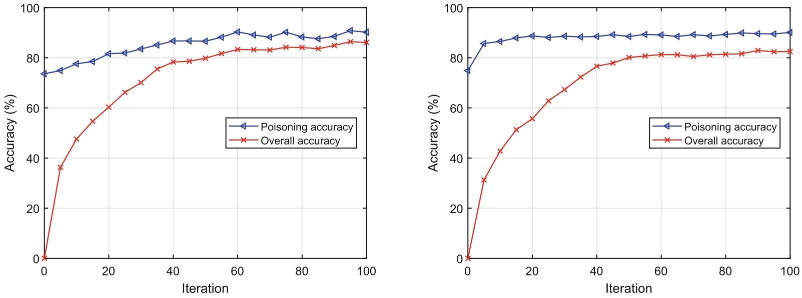


Fig. 1. Poisoning attack in federated learning



(a) Single attacker

(b) Three attackers

Fig. 2. Poisoning accuracy and overall accuracy under attacks

Figure 1 illustrates the detailed processes of the poisoning attack in federated learning. Considering there are three participants, and one of them is attacker, where all the participants agree on a common learning objective and model structure as depicted in the federated learning algorithm. The attacker's purpose is to compromise the global model by uploading the poisoned local model updates. To achieve this goal, he first changes the category label of his target class on his training dataset. Then, he feeds the modified datasets to the local model and computes the poisoned local updates. Note that these poisoned local updates can be scaled to speed up the attack process. Last, these local updates are sent to the central server. After model averaging, the global model can be contaminated by the poisoned model.

Figure 2(a) and (b) illustrate the poisoning accuracy and overall accuracy of image classification task on MNIST dataset. According to the results, we can see that the increasing rate of poisoning task accuracy on three attackers scenarios is faster than the single attacker scenario.

4 Defense Algorithm

In this section, we present the proposed defense approach PDGAN, which is specifically designed for poisoning defense in federated learning. To detect attackers and reduce the impact of the poisoning attack, the PDGAN reconstructs participants training data from their updates in the server and audits accuracy for each participant model using the generated data. The participant whose accuracy is lower than a predefined threshold will be identified as the attacker.

4.1 Overview of PDGAN

In poisoning attacks, attackers try to poison the global model by uploading malicious updates, where misclassification happens after the model has been poisoned on the server side. Anomaly detection methods of federated learning can be categorized into two classes: gradients distance detection and model accuracy auditing. The gradients distance detection method [9] detects anomaly by comparing the distances between gradients uploaded by different participants. For evading this detection, attackers [21] use the optimization algorithm to carry out stealthy poisoning attacks. The model accuracy auditing methods use an auxiliary dataset to identify attackers by checking the accuracy for each participant model.

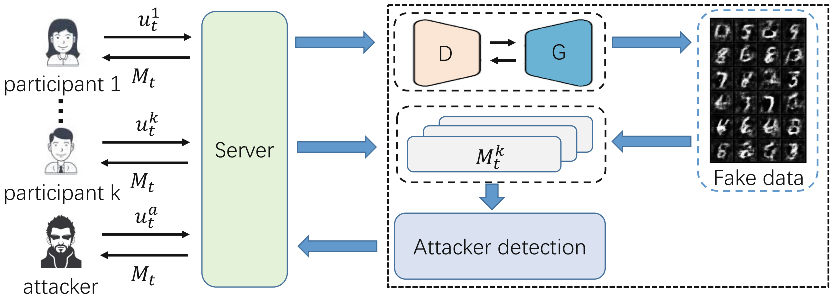


Fig. 3. Overview of the proposed PDGAN method in the federated learning.

Algorithm 1. PDGAN in Federated Learning

```
Data: Parameters updates $u_t^k$ from participant $k$ at iteration $t$; global
model $M_{t-1}$ at iteration $t-1$; Auxiliary data $X_{aux}$; Labellist $L$; Accuracy
threshold $\eta$
Initialize Generator $\mathcal{G}$
for Iteration t do
    Receive updates from the selected participants, $\{u_t^1, u_t^2,..., u_t^k\}$
    Generate $X_{fake}$ from $\mathcal{G}$
    Update the Discriminator by the participant updates,
    $\mathcal{D}_t = \mathcal{D}_{t-1} + \frac{1}{N}\sum_{k=1}^{N} u_t^k$
    Train $\mathcal{D}_t$ by $X_{aux}$ and $X_{fake}$, and Train $\mathcal{G}$
    if $t \geq d\_iter$ then
        for k=1 to N do
            Initialize participant classification model, $M_t^k = M_{t-1} + u_t^k$
            foreach x in $X_{fake}$ do
                $L[k][x] = M_t^k(x)$
            end
        end
        Assign labels for $X_{fake}$ based on $L$
        Calculate accuracy $a^k$ of each participant classification model on
        $X_{fake}$
        Initialize the sum of benign updates $S = 0$ and the number of
        benign participants $NC = 0$
        for k=1 to N do
            if $a^k \geq \eta$ then
                $S = S + u_t^k$
                $NC = NC + 1$
            end
        end
        $M_t = M_{t-1} + \frac{S}{NC}$
        Sent $M_t$ to all participants
    end
    else
        Federated learning averages updates to construct new global
        model and send the new global model to participants
    end
end
```

Figure 3 overviews the proposed poisoning defense mechanism in federated learning. We assume that there are k benign participants and that the $(k+1)$th participant is the attacker who uploads malicious updates u_t^a to the server for poisoning the global model. In federated learning, there is usually an auxiliary data to audit participant model accuracy. However, it is hard that the auxiliary data includes all classes data in the real scenario because participants train models locally and do not share their private training data. To solve this problem, we implement a GAN on the server to reconstruct participant training data. The PDGAN does not detect attackers at the beginning of the training but after some iterations, for the generative adversarial network needs to take some iterations to train itself. We set the proposed method to begin to detect attackers at iteration d_iter. After obtaining the generated data, the server builds a classification model M_t^k for each participant by using the updates u_t^k uploaded by each participant and the global model M_{t-1} of the previous iteration. We can only reconstruct the training data with GAN, but we can not access the data labels. Therefore, we feed the generated data to each participant model and then get the predicted results. We specify that the label with the most occurrences is the true label for each data. After obtaining the generated data and its' labels, the accuracy of participant model can be calculated. Therefore, the participants can be divided into two clusters, benign participants and attackers, by a predefined accuracy threshold η. If one participant is judged to be an attacker, the PDGAN will ignore its updates in this iteration. The Algorithm 1 shows the procedures of federated learning under the PDGAN.

4.2 Structure of PDGAN

The PDGAN involves two components, discriminator and generator, which are deployed on the server side. The target of PDGAN is to generate samples closed to participants private training data by alternately optimizing the discriminator and the generator. In the training phase, the generator generates fake data, and the discriminator discriminates these generated data from real data. According to [21] and [22], there is an auxiliary dataset in the server for evaluating the learning process in federated learning. In the real scenario, the auxiliary dataset is hard to include all classes data, so we assume that the auxiliary dataset only consists of some classes data, which is used by discriminator to achieve the real-fake task. In federated learning, the server averages gradients uploaded by participants to construct a new global model. Therefore, the global model contains information about real data. According to [22] and [23], we use the global model as the discriminator in our proposed method.

Table 1. Network structure of PDGAN

Discriminator	$32^2 \times 1 \xrightarrow{\textit{Conv (stride = 2), LeakyReLU, Dropout}}$ $16^2 \times 64 \xrightarrow{\textit{Conv (stride = 2), BatchNorm, LeakyReLU}}$ $8^2 \times 64 \xrightarrow{\textit{Conv (stride = 2), BatchNorm, LeakyReLU}}$ $4^2 \times 64 \xrightarrow{\textit{Conv (stride = 1), BatchNorm, LeakyReLU}}$ $2^2 \times 128 \xrightarrow{\textit{Conv (stride = 1), BatchNorm, LeakyReLU}}$ $4^2 \times 128 \xrightarrow{\textit{Conv (stride = 1), LeakyReLU}} 2^2 \times 128 \xrightarrow{\textit{AvgPool2d, FC, Softmax}} 11$
Generator	$100 \xrightarrow{\textit{Deconv, BatchNorm, LeakyReLU}}$ $4^2 \times 256 \xrightarrow{\textit{Deconv, BatchNorm, LeakyReLU}}$ $8^2 \times 128 \xrightarrow{\textit{Deconv, BatchNorm, LeakyReLU}} 32^2 \times 1 \xrightarrow{\textit{Tanh}} 32^2 \times 1$

Table 1 shows the network structure of the PDGAN. For discriminator, the kernel size of the first three convolutional layers and the last three convolutional layers are 4×4 and 3×3, respectively. Additionally, we use *BatchNorm* layers and *Dropout* layers to achieve good model performance. For the generator, the kernel size is 4 × 4, and the input is a random vector of length 100. The *LeakyReLU* and *Tanh* are used as activation functions in different layers.

5 Experiments

5.1 Datasets

We used two public datasets in experiments to evaluate the effectiveness of the proposed defense method. The details of the two datasets are shown as follow.

- *MNIST:* The MNIST dataset [24] is a gray-scale handwritten digits dataset that consists of 70,000 images with the size of 28 × 28. There are totally 10 classes digits from 0 to 9 in the MNIST. The dataset is divided into two subsets, 60,000 training images and 10,000 test images, which are commonly used for training and evaluating image classification systems.
- *Fashion-MNIST:* The Fashion-MNIST dataset [25] is another benchmark dataset for machine learning evaluation. The dataset consists of 10 classes fashion product images which are divided into 60,000 training samples and 10,000 test samples.

We resized the images from the two datasets to 32 × 32 in the experiments. All experiments are done by using PyTorch framework on a server with Intel Xeon W-2133 3.6 GHz CPU, Nvidia Quadro P5000 GPU with 16 G RAM and RHEL7.5 OS.

5.2 Experimental Setting

To evaluate the effectiveness of the proposed method, we set two scenarios in our experiments.

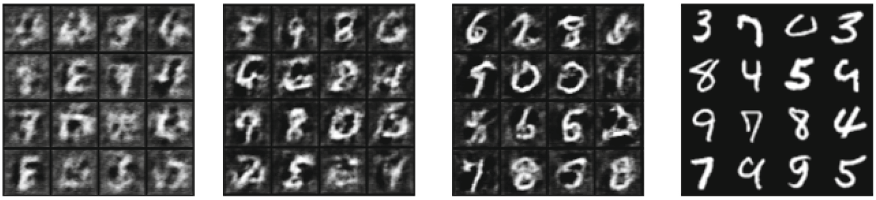
1. **Single attacker:** There are 10 participants in the federated learning where single attacker uploads malicious updates to the server and other 9 participants are benign.
2. **Multiple attackers:** In the real scenario, the attackers are fewer than the benign participants; Therefore, we set the number of attackers to 3 in the experimental setting of multiple attackers. For each iteration, there are 3 attackers and 7 benign participants in federated learning.

In the above experimental setting, training data is randomly assigned to each participant. For the *MNIST*, the target of attackers is to misclassify digit 1 to digit 7. For the *Fashion-MNIST*, the target of attackers is to misclassify T-shirt to Pullover.

We train the federated learning model as a classification task and define two metrics to evaluate the proposed method. One is the poisoning accuracy, which represents the success rate of classifying the poisoned samples to the attacker chosen classes. The second is the overall accuracy, which means the success rate of getting correct classification results with all samples.

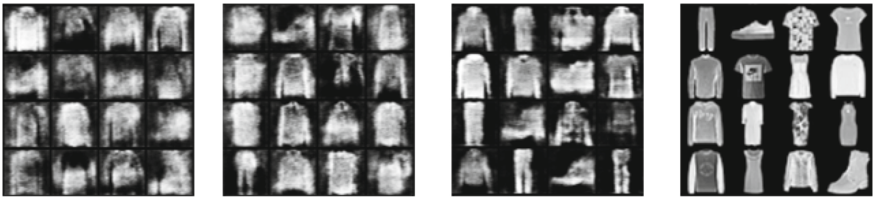
5.3 Data Reconstruction

In this section, we evaluate the proposed method's performance on reconstructing participants training data. We implement a GAN on the server side under the single attacker scenario, where 10 participants attend the training procedure in each iteration. Figure 4 shows the reconstruction results of MNIST dataset in different iterations. In this experimental setting, the auxiliary data in the server includes two classes data (digit 0 and digit 4), which is as the real data to feed the discriminator. After 400 iterations, the generated images are not blurring and can be used as the auditing data for detecting the attacker. Figure 5 shows the reconstruction results of Fashion-MNIST dataset. The auxiliary data includes 3 classed data (dress, coat, sandal). Because the images of Fashion-MNIST are more complex than the MNIST, it takes more iterations to generated not blurring images.



(a) 100 iteration (b) 200 iteration (c) 400 iteration (d) real data

Fig. 4. MNIST reconstruction performance

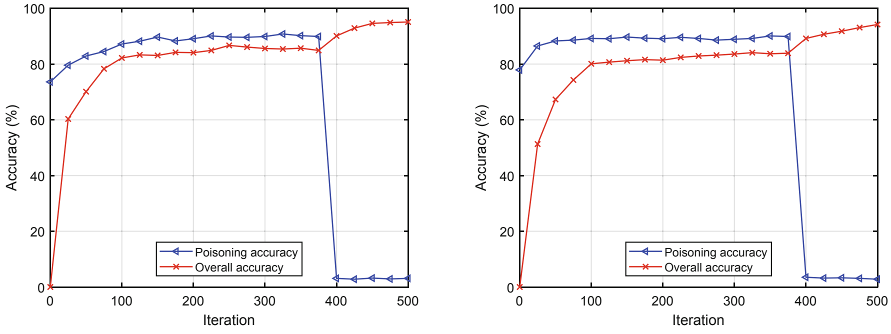


(a) 200 iteration (b) 400 iteration (c) 600 iteration (d) real data

Fig. 5. Fashion-MNIST reconstruction performance

5.4 Poisoning Defense

Figure 6 illustrates the poisoning accuracy and the overall accuracy of image classification on the MNIST dataset. According to the results, we can see that the poisoning task accuracy has achieved a high level immediately once the attacker starts to upload the poisoning updates. That is mainly because the attackers well train their local model for the poisoning target. Besides, the increasing rate of poisoning accuracy on three attackers scenarios is faster than the accuracy of the single attacker scenario, and the overall accuracy is almost similar in both scenarios. We set the *d_iter* to 400, which means that the PD-GAN begin to detect attacks at the 400th iteration. From Fig. 6(a), poisoning accuracy immediately drops to 3.1% at the 400th iteration, and the overall accuracy increases to 90.1%. The reason behind this result is that the PDGAN removes the malicious updates of the attacker, and only benign participants can contribute to the global model. From Fig. 6(b), poisoning accuracy drops to 3.5% at the 400th iteration, and the overall accuracy increases to 89.2%.

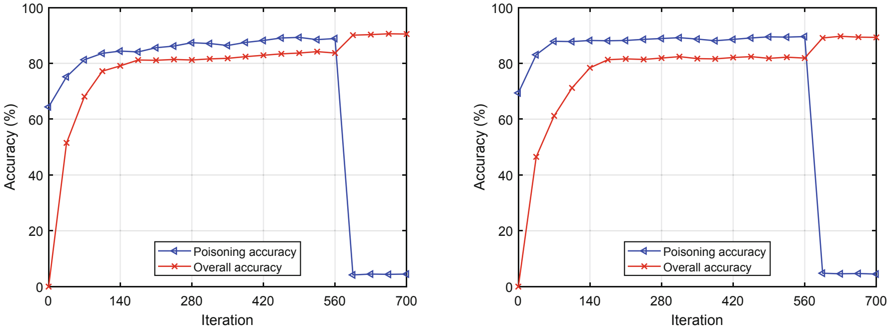


(a) Single attacker

(b) Three attackers

Fig. 6. Detection mechanism on MNIST dataset

As shown in Fig. 7, the experimental results on Fashion-MNIST are similar to the results of MNIST. The proposed defense method detect attacks at the 600-th iteration. For the single attacker scenario, the poisoning accuracy drops immediately to 4.5%, and the overall accuracy increases to 89.7%. For the three attackers scenario, the poisoning accuracy drops immediately to 5.1%, and the overall accuracy increases to 88.6%. The experimental results demonstrate that the proposed method can indeed detect the attackers and eliminate the effect of poisoning attacks.



(a) Single attacker

(b) Three attackers

Fig. 7. Detection mechanism on Fashion-MNIST dataset

6 Conclusion and Future Work

In this paper, we propose a novel poisoning defense method PDGAN in federated learning. The proposed method is based on the generative adversarial network, which is implemented on the server side and can reconstruct participants training data. By using the generated data, the proposed method audits the accuracy of each participant's model and then identify attackers. Experiment results demonstrate that the PDGAN can effectively reconstruct the training data and successfully defend the poisoning attack by auditing the accuracy of the participant model. In future work, we plan to explore the poisoning defense for federated learning with device, class, or user-level differential privacy.

References

1. Ribeiro, M., Grolinger, K., Capretz, M.A.M.: MLaaS: machine learning as a service. In: Proceedings 14th International Conference on Machine Learning and Applications (ICMLA 2015), pp. 896–902 (2015)
2. Hesamifard, E., Takabi, H., Ghasemi, M., Wright, R.N.: Privacy-preserving machine learning as a service. In: Proceedings 19th Privacy Enhancing Technologies Symposium (PETS 2018), pp. 123–142 (2018)
3. Yu, S.: Big privacy: challenges and opportunities of privacy study in the age of big data. IEEE Access **4**, 2751–2763 (2016)
4. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017), pp. 1–10 (2017)
5. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 1–19 (2019)
6. Jagielski, M., Oprea, A., Biggio, B., Liu, C., N-Rotaru, C., Li, B.: Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: Proceedings 39th IEEE Symposium on Security and Privacy (SP 2018), pp. 19–35 (2018)

7. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. In: Proceedings 29th International Conference on Machine Learning (ICML 2012), pp. 1807–1814 (2012)
8. Fang, M., Yang, G., Gong, N.Z., Liu, J.: Poisoning attacks to graph-based recommender systems. In: Proceedings 34th Annual Computer Security Applications Conference (ACSAC 2018), pp. 381–392 (2018)
9. Fung, C., Yoon, C.J.M., Beschastnikh, I.: Mitigating sybils in federated learning poisoning (2018). https://arxiv.org/abs/1808.04866
10. Bonawitz, K., et al.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings 24th ACM Conference on Computer and Communications Security (CCS 2017), pp. 1175–1191 (2017)
11. Mohassel, P., Zhang, Y.: SecureML: a system for scalable privacy-preserving machine learning. In: Proceedings 38th IEEE Symposium on Security and Privacy (SP 2017), pp. 19–38 (2017)
12. Phong, L.T., Aono, Y., Hayashi, T., Moriai, S.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. **13**(5), 1333–1645 (2018)
13. Blanchard, P., Mhanmdi, E.M.E., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. In: Proceedings 32th Annual Conference on Neural Information Processing Systems (NIPS 2017), pp. 119–129 (2017)
14. Yin, D., Chen, Y., Ramchandran, K., Bartlett, P.: Byzantine-robust distributed learning: towards optimal statistical rates. In: Proceedings 35th International Conference on Machine Learning (ICML 2018) (2018)
15. Shen, S., Tople, S., Saxena, P.: AUROR: defending against poisoning attacks in collaborative deep learning systems. In: Proceedings 32nd Annual Computer Security Applications Conference (ACSAC 2016), pp. 508–519 (2016)
16. Baracaldo, N., Chen, B., Ludwig, H., Safavi, J.A.: Mitigating poisoning attacks on machine learning models: a data provenance based approach. In: Proceedings 10th ACM Workshop on Artificial Intelligence and Security (AISec 2017), pp. 103–110 (2017)
17. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings 22nd ACM Conference on Computer and Communications Security (CCS 2015), pp. 1310–1321 (2015)
18. Zhang, X., Felix, X.Y., Kumar, S., Chang, S.-F.: Learning spread-out local feature descriptors. In: Proceedings IEEE International Conference on Computer Vision (ICCV 2017), pp. 4595–4603 (2017)
19. Goodfellow, I., et al.: Generative adversarial nets. In: Proceedings 29th Annual Conference on Neural Information Processing Systems (NIPS 2014), pp. 2672–2680 (2014)
20. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. In: Proceedings 27th International Joint Conference on Artificial Intelligence (IJCAI 2018) (2018)
21. Bhagoji, A.N., Chakraborty, S., Mittal, P., Prateek, M., Calo, S.: Analyzing federated learning through an adversarial lens (2018). https://arxiv.org/abs/1811.12470
22. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: user-level privacy leakage from federated learning. In: Proceedings 38th Annual IEEE International Conference on Computer Communications (INFOCOM 2019) (2018)

23. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the GAN: information leakage from collaborative deep learning. In: Proceedings 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS 2017), pp. 603–618 (2017)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp. 2278–2324 (1998)
25. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017). https://arxiv.org/abs/1708.07747