Received: 2 December 2019 | Revised: 8 May 2020 | Accepted: 18 May 2020

DOI: 10.1002/cpe.5906

WILEY

SPECIAL ISSUE PAPER

Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks

Ying Zhao[1] | Junjun Chen[1] | Jiale Zhang[2] | Di Wu[3,4] | Michael Blumenstein[4] | Shui Yu[3]

[1]College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China

[2]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, China

[3]School of Computer Science, University of Technology Sydney, Ultimo, New South Wales, Australia

[4]School of Information Technology, University of Technology Sydney, Ultimo, New South Wales, Australia

Correspondence
Junjun Chen, College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China.
Email: chenjj@mail.buct.edu.cn

Summary

In the age of the Internet of Things (IoT), large numbers of sensors and edge devices are deployed in various application scenarios; Therefore, collaborative learning is widely used in IoT to implement crowd intelligence by inviting multiple participants to complete a training task. As a collaborative learning framework, federated learning is designed to preserve user data privacy, where participants jointly train a global model without uploading their private training data to a third party server. Nevertheless, federated learning is under the threat of poisoning attacks, where adversaries can upload malicious model updates to contaminate the global model. To detect and mitigate poisoning attacks in federated learning, we propose a poisoning defense mechanism, which uses generative adversarial networks to generate auditing data in the training procedure and removes adversaries by auditing their model accuracy. Experiments conducted on two well-known datasets, MNIST and Fashion-MNIST, suggest that federated learning is vulnerable to the poisoning attack, and the proposed defense method can detect and mitigate the poisoning attack.

KEYWORDS

federated learning, generative adversarial networks, model security, poisoning attacks

1 | INTRODUCTION

Recently, with the proliferation of Internet of Things (IoT) and mobile edge computing techniques, billions of mobile and IoT devices are connected to the Internet, generating massive sensed data at the edge of the network. For enabling intelligence networking applications, conventional machine learning architecture provides centralized big data analysis services to make automated decisions, which known as the machine-learning-as-a-service (MLaaS).[1,2] Driven by this trend, many machine learning applications (eg, health monitoring, smart city, recommendation systems, smart grid) are provided in a centralized manner. However, such a learning scenario causes a notable data privacy leakage problem due to the training data is outsourced to the unknown third parties, where participants' private raw data could be disclosed to the untrusted entities.[3,4] For solving this privacy issue and meanwhile benefiting from the MLaaS framework, collaborative learning has been explored recently, which is the decentralized learning framework for mobile devices. Shokri et al[5] present the first collaborative learning model, which can preserve the participants' data privacy in the model training procedure. This global model is predefined by the curator and is trained by each participant locally to generate the local model parameters. By sampling these parameters among all the participants, the global model can be updated within a series of iterations.

Different from conventional collaborative learning frameworks, federated learning[6] (ie, deep learning framework in edge computing or fog computing)[7] can protect participants' data privacy in the training procedure. The most significant improvement of federated learning is that the global model at the server side is updated by the federated averaging algorithm,[6] which can efficiently handle the non-independent and identically distributed (non-IID) training data.[8] However, federated learning still faces significant security challenges because what the participants do at the

local side is invisible to the central server. In other words, the local model updates are determined by the participants, while the central server cannot control them. By leveraging this property, the federated learning model could be attacked by any malicious participant through the poisoned local model updates. We recall this malicious behavior as the poisoning attack,[9,10] which is wildly explored in the traditional centralized learning system. In this article, we aim at the label-flipping attack,[11] where the adversaries are able to inject the crafted poisoned samples into the training dataset by flipping specific target samples' labels, thus changing the model prediction boundary in the inference phase.

The reasons why federated learning is vulnerable to poisoning attacks can be summarized as follows. (i) There are no corresponding participant's identity authentication mechanisms in federated learning where adversaries can pretend they are benign participants. (ii) The participant trains model locally, and the training procedure is transparent to other participants and the server. The server cannot determine whether the model parameters uploaded by the participants are malicious. (iii) Because the participants' training data is non-IID, it is hard to use the discrepancy-based method to detect the uploaded model parameters. Furthermore, recent defense methods mainly focus on encryption and secure multiparty computation techniques,[12-14] Byzantine-tolerant learning,[15,16] anomaly detection,[17] and clustering.[18] However, the methods mentioned above do not take into account the context of federated learning. Therefore, these methods can not be used to defend poisoning attacks in federated learning.

Motivated by the vulnerability of federated learning and shortcomings of existing defense proposals, we focus on detecting and mitigating the label-flipping poisoning attack in this work. The main idea of our defense mechanism is that we utilize the generative adversarial networks (GAN)[19] on the server side to generate an auditing dataset, which is used to check the participant model accuracy. If the accuracy of one participant model is less than a threshold, the proposed defense method identifies this participant as an adversary and removes the model parameters uploaded by this participant in this training iteration. The contributions of this article can be summarized as follows.

- We devise a poisoning defense mechanism for federated learning. The defense approach based on accuracy auditing can detect and mitigate poisoning attacks in federated learning.
- We deploy a GAN in the server of federated learning to generate an auditing dataset that can be used to identify adversaries by checking the participant model accuracy. Compared with the model inversion method, the GAN can generate high-quality auditing data.
- We simulate the poisoning attack and defend it in federated learning. Experiments conducted on two well-known datasets, MNIST and Fashion-MNIST, suggest that federated learning is vulnerable to the poisoning attack, and the proposed defense mechanism can detect and mitigate the poisoning attack.

This article is structured as follows. Section 2 provides a review on federated learning and poisoning attacks. In Section 3, we introduce the poisoning attack model and the defense intuition. Section 4 describes the proposed poisoning defense mechanism. Experimental results are given in Section 5. Finally, Section 6 concludes this article.

2 | BACKGROUND

2.1 | Federated learning

In the conventional collaborative deep learning training framework, a powerful centralized cloud server is required to ask participants to upload their training data, bringing intelligent services to the mobile and IoT devices. After receiving training data from participants, the server trains a deep neural network (DNN) model on these samples until its optimal parameters are reached. In the end, participants can use this DNN model by downloading it or executing the prediction on the cloud server. The workflow of a centralized deep learning model is shown in Figure 1A. Obviously, such a centralized training method is possible to leak participants' data privacy.

Different from the conventional collaborative learning mechanism, federated learning can provide a unique distributed training architecture, which can prevent user data privacy leakage.[20] In federated learning, the model training phases are executed at the local side, and the shares between the central server and participants are only model parameters (gradients). To speed up the convergence of the global model, each participant trains the local model for multiple times and uploads the generated local model parameters to the server. After that, the server receives multiple versions of local model parameters, which will be aggregated to improve the global model over communication rounds. Ultimately, the server and participants can get the optimal model parameters. Figure 1B illustrates the overall federated learning architecture. Note that, in the federated learning, all participants agree with the training protocol, including learning objectives, model structures, and other information. The federated training procedure is iteratively executed until the global model accuracy reaches a predefined threshold.

2.2 | Poisoning attacks and defenses

To illustrate the poisoning attack against federated learning, we describe the attack process in this section. As shown in Figure 2, two benign participants and one adversary jointly train an image classification model in federated learning. The purpose of benign participants is to obtain a global

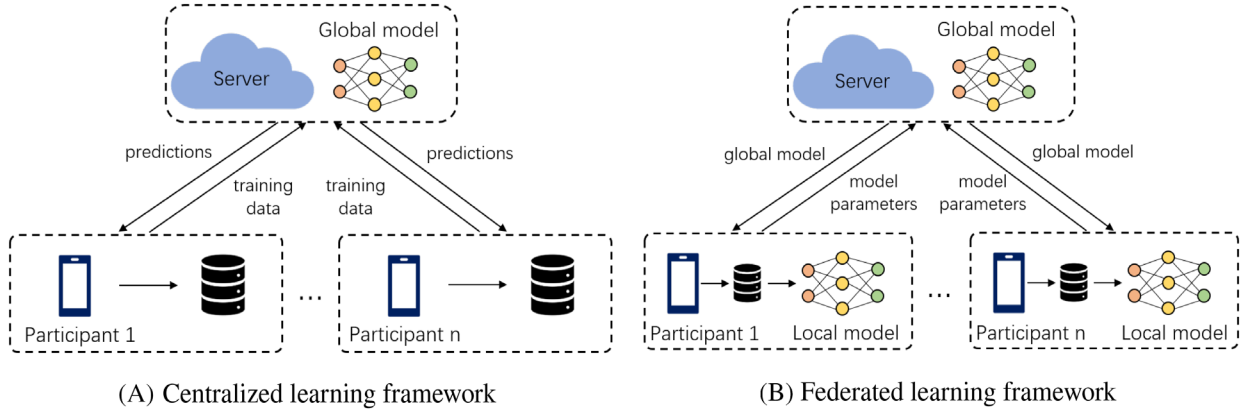


FIGURE 1 Comparison of, A, centralized learning and, B, federated learning frameworks

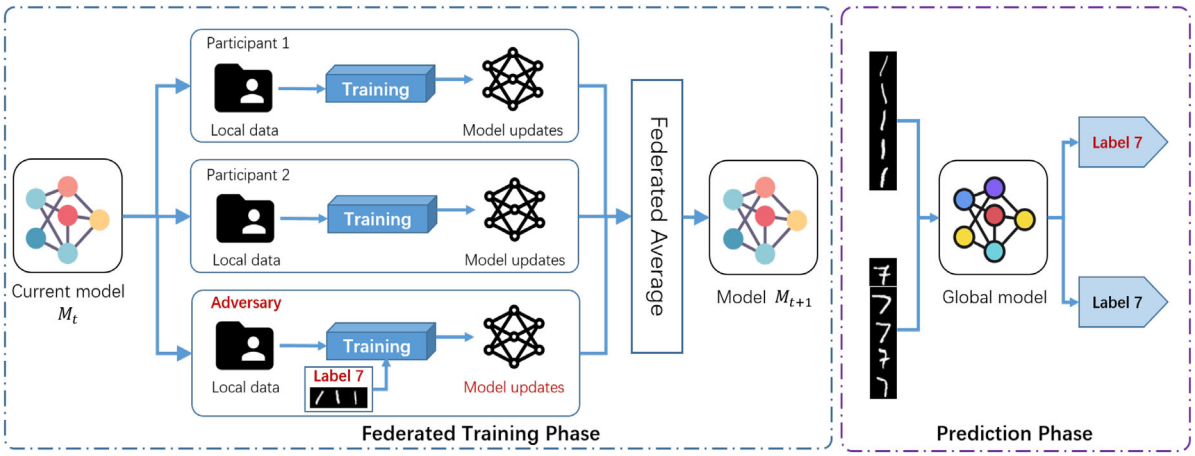


FIGURE 2 Poisoning attack in federated learning

model with excellent classification performance, and the purpose of the adversary is to make the global model misclassify target samples. To initiate poisoning attacks, the adversary first modifies labels of target samples. Then, the local model is trained on these modified datasets and generates the malicious model parameters. In the end, these model parameters, which are scaled up to improve their weights, are sent to the server to renew the classification model. After the global model update, the uploaded malicious parameters can change the original prediction boundary of the classification model and cause the model to make incorrect classification decisions on target samples.

The poisoning attacks and poisoning defense mechanisms have been well investigated for conventional deep learning in recent years. Shen et al[18] proposed a countermeasure to poisoning attacks for collaborative deep learning, which identified suspicious users by analyzing users' uploaded features distribution. Baracaldo et al[17] devised a poisoning defense based on data provenance to identify and filter the poisonous data. Although the above methods achieved excellent detection performance under their experimental settings, they did not consider that participants' training data is non-IID.

3 | OVERVIEW OF THE POISONING ATTACK MODEL AND DEFENSE METHOD

In this section, we first describe the poisoning attack model in the context of federated learning and then introduce our defense assumptions, goals, as well as intuition.

3.1 | Attack model

Following the descriptions of federated learning in Section 2, our attack model considers the poisoning attacks in federated learning, which performs an image classification task. Assuming that the central server is fully trusted, there are one or more adversaries among the participants, who aim to contaminate the global model by uploading poisonous model updates.

Adversary's goal. In the training stage of federated learning, the adversary pretends to be a benign participant and uploads malicious model parameters to induce the global model to misclassify specific target samples. To accomplish this goal, the adversary mainly focuses on the following evaluation metrics.

- Poisoning accuracy: the global model should achieve a great poisoning accuracy on specific target samples, which means the global model is already contaminated with malicious model parameters.
- Overall accuracy: the poisonous model parameters should only impact the classification results of specific target samples. In other words, the global model should achieve high overall accuracy.

Adversary's capability. In the context of federated learning, the adversary usually has the following capabilities to launch poisoning attacks.

- Knowledgeable: since all the participants in federated learning agree with the learning protocol, participants know about the model structure, loss function, learning objective, and others; therefore, the adversary can use this information to enhance their attacks.
- Active: participants train model locally and only upload model parameters to the server, so the adversary can control the whole poisoning procedure, where he can craft the model parameters to contaminate the global model.

3.2 | Defense assumptions and goals

We make assumptions about our poisoning defense mechanism as follows. First, we assume that the server is fully trusted, which plays a role as the defender to detect poisoning attacks and mitigate such phenomena. Although our defense mechanism relies on reconstructing participants' training samples, this approach does not destroy the user-level privacy guarantees since the generated samples are the overall training data from all participants, not a specific participant. Second, the defender has sufficient computing resources, such as GPUs or cloud-based units, to handle the computation consumption of poisoning detection and poisoning mitigation. Furthermore, our defense mechanism detects and mitigates poisoning attacks after several training iterations rather than at the beginning because the GAN needs to be well trained to generate high-quality samples.

The proposed defensive mechanism includes the following two goals:

- Detecting poisoning attacks: we want to make a precise determination of whether a certain federated learning model has been poisoned by the adversaries.
- Mitigating poisoning attacks: we want to eliminate the influence of poisoning attacks by removing poisonous updates of adversaries. Moreover, the above operations do not affect the collaborative training of benign participants in federated learning.

3.3 | Defense intuition

Key intuition. We aim to devise an effective poisoning defense mechanism by auditing the participant model accuracy to identify the adversary in federated learning. We derive our defense intuition from two basic facts of the federated training method. One is that the adversary's model will misclassify specific inputting samples since the purpose of the adversary is to contaminate the global model by its malicious model parameters. Another fact is that the poisonous model parameters are scaled up by the adversary to improve their weights, which means that the accuracy of the adversary's model will become lower. For example, Bagdasaryan et al[21] proposed to scale up the adversary's local gradients ($\Delta\hat{M}_t = \lambda\Delta M_t$) to increase the influence of malicious updates after the model average in federated learning. According to these two facts, we present our poisoning defense approach to identify malicious adversaries and mitigate the poisoning attacks in federated learning. Apparently, the key point for constructing our defense approach is to generate an auditing dataset and use it to check the participant model accuracy in the training procedure of federated learning.

In the setting of federated learning,[22,23] there is usually an auxiliary dataset X_{aux} in the central server. Moreover, federated learning is designed to protect data privacy, where participants do not share their training data to a third party server. Therefore, the X_{aux} may only contain a small number of samples collected by the server. Once the auxiliary dataset does not contain data on the target category of the poisoning attack, it can not be used to audit the participant model accuracy to identify poisoning attacks. To overcome the above problem, we use GAN to generate auditing data for the proposed poisoning defense method.

As shown in Figure 3, with the federated learning protocol goes on, the server can observe all the participants' model updates u_t^k at a certain training iteration t. However, the server is impossible to directly access the participants' local training datasets. In this situation, for constructing an auditing dataset, we use a GAN deployed in the central server to generate an auditing dataset. Once the GAN can generate high-quality samples, our

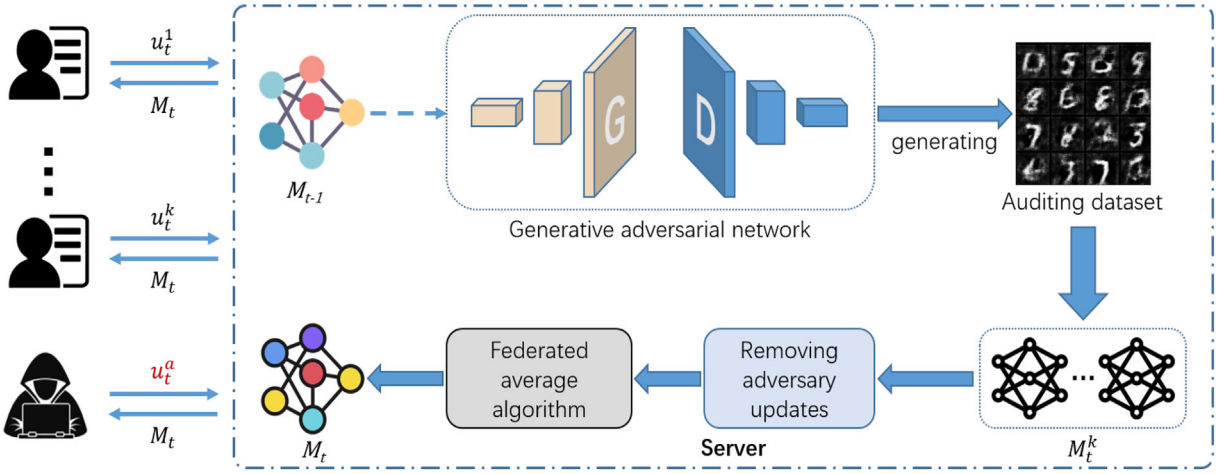


FIGURE 3 Proposed defense mechanism using GAN

detection mechanism starts to audit the participant model accuracy using generated samples. In this way, the adversary can be identified through a predefined accuracy threshold θ. Note that the local model M_t^k for the participants k is calculated by adding participant's model updates u_t^k to the global model M_{t-1}^k in previous training iteration.

Detecting poisoning attacks. Our key intuition of detecting poisoning is that if a federated learning model is poisoned by adversaries, the participant's local model will present different prediction results on the same inputting samples. Since the adversaries usually use the scaling up method to increase the impact of poisonous local updates, this classification difference could be easily noticed by a trusted central server. Therefore, our defense mechanism iteratively reconstructs participants' training data as the auditing dataset, verifies the accuracy of each classification model M_t^k, and determines whether there exist poisoning attacks.

Mitigating poisoning attacks. After the poisoning detection, we can identify adversaries from participants. To mitigate poisoning attacks, the proposed defense approach removes adversaries' poisonous updates from the federated averaging algorithm in the current training iteration.

4 | DETAILED DEFENSE METHODOLOGY

In this section, we describe details of the proposed defense approach to detect and mitigate the poisoning attack in federated learning. We first describe the auditing data generation procedure. Then, we introduce the defense algorithm and its neural network structures.

4.1 | Auditing dataset generation using GAN

GAN has achieved great success in computer vision, data privacy protection,[24] and other research areas since it was introduced in 2014. Owing to its adversarial training architecture, GAN can learn original data distribution of samples and generate high-quality fake samples in many domains, such as text generation and image generation. In this article, we use the GAN to reconstruct user training data as the auditing dataset for the poisoning defense method. According to the defense intuition in Section 3.3, we can audit the participant model accuracy to identify and remove the adversary in the training process; therefore, a high-quality auditing dataset is a key to the proposed defense mechanism.

In the server side of federated learning, a GAN is implemented to generate the auditing dataset, which includes two parts, discriminator ($\mathcal{D}$) and generator ($\mathcal{G}$). Because participants collaboratively train the global model M with their own training data, the M includes the information of user data; therefore, we can use M to initialize $\mathcal{D}$ at the beginning of each training iteration to reconstruct participants' training data. In the training process of the GAN, $\mathcal{G}$ and $\mathcal{D}$ play an adversarial game, where $\mathcal{G}$ tries to fool $\mathcal{D}$ with generating data similar to the real data, and $\mathcal{D}$ tries to differentiate the generated data between real data. In the setting of federated learning, there is usually an auxiliary dataset X_{aux}, which may not contain all classes data, for the server side to evaluate the learning performance of the global model, so we can feed X_{aux} to $\mathcal{D}$ to update its neural network in the iterative training process of the GAN. The optimization objectives of $\mathcal{G}$ and $\mathcal{D}$ are expressed in Equations (1) and (2), respectively.

$$V_{\mathcal{D}} = E_{x\sim p_{aux}}[\log \mathcal{D}(x)] + E_{z\sim p_z}[\log(1 - \mathcal{D}(\mathcal{G}(z)))], \tag{1}$$

$$V_{\mathcal{G}} = E_{z\sim p_z}[\log(1 - \mathcal{D}(\mathcal{G}(z)))], \tag{2}$$

where $x \sim p_{aux}$ represents sample x from the auxiliary dataset, and $z \sim p_z(z)$ indicates the random vector z from a random distribution. In the training produce, $\mathcal{D}$ maximizes $V_{\mathcal{D}}$ to discriminate the real data and generated data, and $\mathcal{G}$ minimizes $V_{\mathcal{G}}$ to make the generated data as similar as possible to the real data. Until the $\mathcal{D}$ cannot differentiate the generated data between the real data, the training achieves the Nash equilibrium.

4.2 | Detecting and mitigating poisoning attacks

To initiate poisoning attacks, adversaries upload malicious model parameters to contaminate the global model, which leads to a misclassification of specific target samples. To defend against such attacks, some defense methods, such as updates distance detection, have been proposed in recent years. Fung et al[11] calculated the distances between model updates uploaded by different participants to detect the harmful updates. However, this detection method can be evaded by an elaborate attack. For example, adversaries[22] can minimize the distance of malicious updates between benign updates by an optimization algorithm to accomplish stealthy poisoning attacks. In this article, we use an accuracy auditing methods to identify adversaries by checking the participant model accuracy with the generated auditing dataset.

Algorithm 1: Federated learning with poisoning defense mechanism

Before Training: All participants confirm learning objective, model network structure, and other information.

Training Phase: $\mathcal{L}$ is the loss function; η is the learning rate; The participants are indexed by k; The training iteration is indexed by t; u_t^k is parameters updates; M_t is the global model; X_{aux} is the auxiliary dataset; L is the label list; θ is the accuracy threshold; S is used to store benign updates; CV is the counting variable for benign participants.

for *Iteration t* **do**
  /* Participant side: */
  Train local model with private training data
  Update model parameters with $u_t^k = u_t^k - \eta \frac{\partial}{\partial X^k} \mathcal{L}$
  Send u_t^k to the server
  /* Server side: */
  Receive model updates $\{u_t^1, u_t^2, ..., u_t^k\}$
  Update discriminator $\mathcal{D} = M_{t-1} + \frac{1}{N} \sum_{k=1}^{N} u_t^k$
  Train $\mathcal{G}$ and $\mathcal{D}$ with X_{aux}
  Generate auditing dataset $X_{auditing}$ by $\mathcal{G}$
  if $t \geq d\_iter$ **then**
    for *k=1 to N* **do**
      Construct classification model, $M_t^k = M_{t-1} + u_t^k$, for each participant
      foreach *x in* $X_{auditing}$ **do**
        $L[k][x] = M_t^k(x)$
      end
    end
    Specify labels for $X_{auditing}$ and use $X_{auditing}$ to calculate model accuracy a^k of M_t^k
    Initialize $S = 0$ and $CV = 0$
    for *k=1 to N* **do**
      if $a^k \geq \theta$ **then**
        $S = S + u_t^k$
        $CV = CV + 1$
      end
    end
    $M_t = M_{t-1} + \frac{S}{CV}$
    Sent M_t to all participants
  end
  else
    $M_t = M_{t-1} + \frac{1}{N} \sum_{k=1}^{N} u_t^k$
    Sent M_t to all participants
  end
end

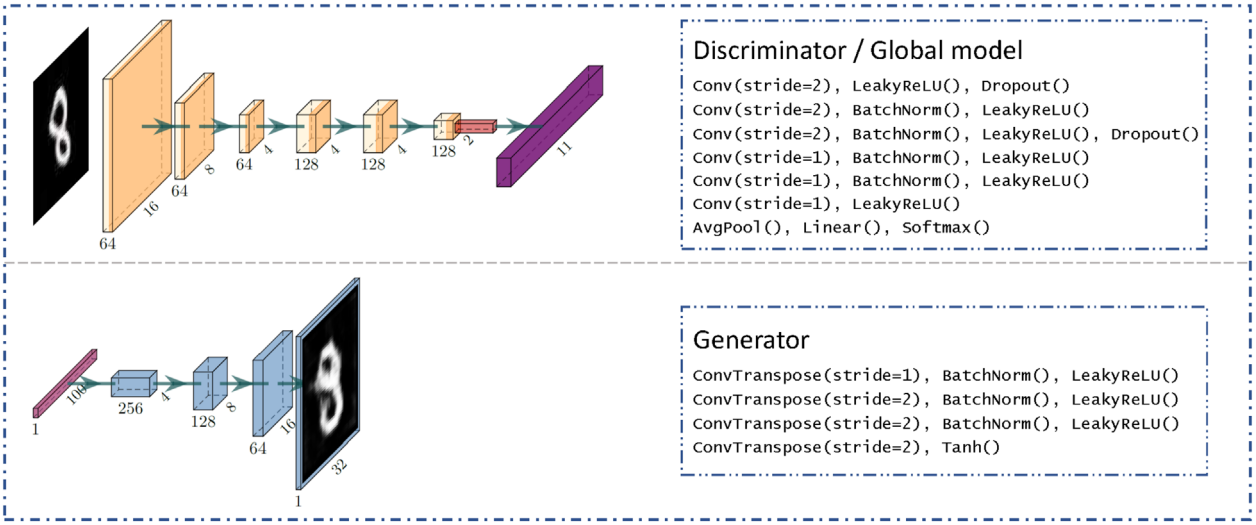


FIGURE 4 Network structures used in poisoning defense mechanism

Algorithm 1 describes the federated learning algorithm with the proposed poisoning defense mechanism, where total k participants, including benign participants and adversaries, collaboratively engage in training a global model.

Detecting poisoning attacks. To obtain a high-quality auditing dataset, the GAN needs to be well trained after some iterations; therefore, the proposed defense method does not detect poisoning attacks at the beginning of the federated learning. Before a predefined iteration *d_iter*, on the participant side, participants upload their model updates to the server and continue to train the global model received from the server with their own training data. Moreover, on the server side, the server uses these uploaded updates to renew the discriminator and the global model, and trains GAN with Equations (1) and (2). After the iteration *d_iter*, the defense mechanism is activated, as shown in *Line 11 to Line 28* in Algorithm 1. The server uses model updates u_t^k to construct model M_t^k and then identifies the adversary by auditing the accuracy of M_t^k.

In the generation procedure of the auditing dataset, we use the GAN to generate data, but we cannot obtain the data label information. To solve this problem, we let all participants' classification models vote on the label for each generated data and specify the label with the most support as its label. The above practice is based on the assumption that in the real scenario, the number of adversaries is smaller or much smaller than that of benign participants. After acquiring the auditing dataset and its labels, we can calculate the participant model accuracy. According to the poisoning defense intuition in Section 3.3, we can use a predefined accuracy threshold θ to identify adversaries in federated learning.

Mitigating poisoning attacks. If one participant is identified as the adversary, the defense mechanism will discard its updates in the current iteration to mitigate the adverse influence, as shown in *Line 19 to Line 27*in Algorithm 1. After that process, the new global model M_t without adversary's updates will be sent to participants to carry out next iteration training.

4.3 | Network structures used in poisoning defense

The proposed poisoning defense architecture comprises three components, global model, discriminator, and generator, whose neural network structures are shown in Figure 4. The global model and the discriminator share the same convolutional neural network structures, and the generator is composed of multiple deconvolution layers.

For the global model and discriminator, the output of the last layer is 11, which is different from the conventional 10-dimension output classifier for MNIST.[25] The reason for this design is that the discriminator needs one extra dimension to classify the fake samples from the generator. There are six convolutional layers with different kernel size (4×4 or 3×3) in the global model and the discriminator. Moreover, *Dropout* and *BatchNorm* layers are used to improve neural network performance. For the generator, there are four deconvolution layers with 4×4 kernel size. The input of the generator is a random variable with 100-dimension, and the output is a 32×32 image.

5 | EXPERIMENTS

5.1 | Datasets and experimental setup

To evaluate the proposed poisoning defense mechanism, two well-know public datasets, MNIST and Fashion-MNIST, are used in experiments.

- *MNIST dataset*. The MNIST dataset[25] is a classical hand-written digits recognition dataset, which is generally used for the performance evaluation of image classification algorithms in the computer vision field. There are 10 number classes from digit 0 to digit 9 in this dataset. This MNIST includes 70 000 gray-scale images with the resolution of 28×28, which is split into two separate parts, 60 000 training images and 10 000 testing images.
- *Fashion-MNIST dataset*. The Fashion-MNIST dataset[26] is a typical dataset used for evaluating deep learning algorithms. There are totally 10 categories of fashion product images in this dataset. The Fashion-MNIST dataset is also composed of gray-scale images, which includes 60 000 training images and 10 000 testing images, with the resolution of 28×28.

We resized the resolution of training and testing samples to 32×32 to feed the neural networks shown in Section 4.3. Moreover, we coded with the PyTorch framework and conducted all experiments on an RHEL7.5 server with Nvidia GeForce 1080 Ti GPU.

Two experimental scenarios are designed to evaluate the proposed poisoning defense mechanism. For each scenario, 10 participants are selected from 100 participants in every training iteration of federated learning.

1. *Single adversary*: For each iteration of federated learning, one adversary, who uploads its malicious model updates to contaminate the global model, and the other nine benign participants jointly train a model.
2. *Multiple adversaries*: In this setting, we assess the impact of multiple adversaries in federated learning. In the real learning situation, there are more benign participants than adversaries. Thus, the number of adversaries is fixed at three in each training iteration in our experiments.

In the experiments, the goal of federated learning is to train an image classification model for all participants. We randomly assign training data to benign participants. For the MNIST dataset, adversaries try to induce the global model to classify number 1 as number 7. For the Fashion-MNIST dataset, adversaries try to induce the global model to classify the T-shirt to pullover. Two metrics, poisoning accuracy and overall accuracy, are used to evaluate the proposed poisoning defense mechanism in this paper. The poisoning accuracy indicates the ratio of the number of misclassified target images to the total number of target images. The overall accuracy presents the ratio of the number of correctly classified images to the total number of images.

5.2 | Auditing dataset generation

The quality of the generated image ultimately determines whether the proposed method can detect and mitigate poisoning attacks. To generate an auditing dataset, a GAN is deployed in the server. The generation results of MNIST and Fashion-MNIST datasets are shown in Figures 5 and 6, respectively. For the MNIST, the auxiliary dataset X_{aux} consists of two categories images, that is, number 0 and number 4, which are used to train the discriminator. For Fashion-MNIST, the X_{aux} is composed of three categories images, that is, dress, coat, and sandal, in the experiments. What can be seen in Figures 5 and 6 is that the generated images become more and more clear as the number of training iteration increases. The GAN can generate high-quality images at the 400th training iteration for the MNIST dataset. Since the images are complicated, the GAN needs to cost more training time to generate clear images for the Fashion-MNIST dataset.

Next, we compare the generation performance of GAN against that of the model inversion attack in federated learning. The model inversion attack[27,28] is another data reconstruction method, which can be launched by only accessing the application programming interface (API) of the machine learning model with a series of iterative queries. For example, to exploit user training data from an image classification model, the adversary starts with a random image and then iteratively optimizes the image to improve its confidence under the target model. In federated learning, the server can access every participant's model, so we try to implement the model inversion to generate the auditing dataset for poisoning detection. However, the model inversion fails to produce convincing images, as shown in Figure 7. The reason for poor attack performance is that the neural network model is too complicated for model inversion attack, and the gradient descent methods easily stuck in local minima.

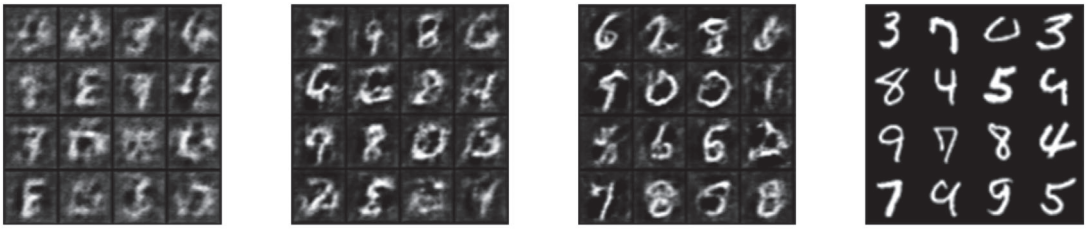


FIGURE 5 MNIST generation performance

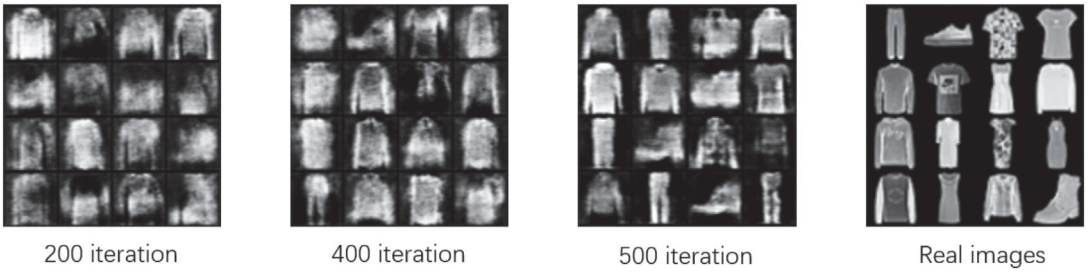
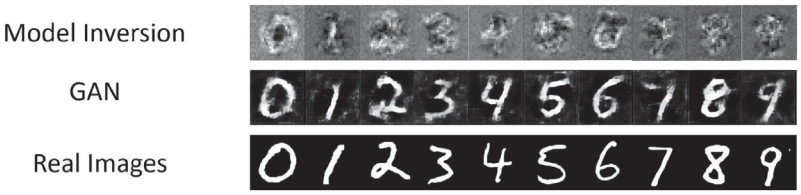


FIGURE 6 Fashion-MNIST generation performance

FIGURE 7 Generation performance comparison



5.3 | Poisoning attack and defense in federated learning

In this section, we first prove that federated learning is vulnerable to poisoning attacks. Similar to the backdoor attack[21] in federated learning, if adversaries intend to contaminate the global model, they should increase the number of malicious participants or scale their model updates to eliminate the impact of model updates averaging in the server. Figure 8 illustrates that the mean poisoning accuracy and mean overall accuracy of the global model on the different number of adversaries. For the MNIST dataset, the mean poisoning accuracy is stable at around 88%, and the mean overall accuracy is stable at around 81%. For the Fashion-MNIST dataset, the mean poisoning accuracy is stable at around 85%, and the mean overall accuracy is stable at around 78%. As shown through the black-dash, with the increase of the number of malicious participants, adversaries can decrease the scaling factor to evade the detection based on distance.

We then evaluate the effectiveness of the proposed poisoning defense mechanism. Figures 9 and 10 show the overall accuracy and the poisoning accuracy on the MNIST dataset and Fashion-MNIST dataset, respectively. From experimental results, the poisoning accuracy increases after adversaries initiate poisoning attacks. The reason behind this result is as follows: adversaries train their models locally to achieve high poisoning accuracy and then upload scaled model updates to the server, so the global model is influenced by these malicious updates. Compared with the multiple adversaries scenario, the poisoning accuracy of the single adversary scenario converges more slowly. The above phenomenon shows that the more attackers in federated learning, the easier it is to achieve poisoning attacks.

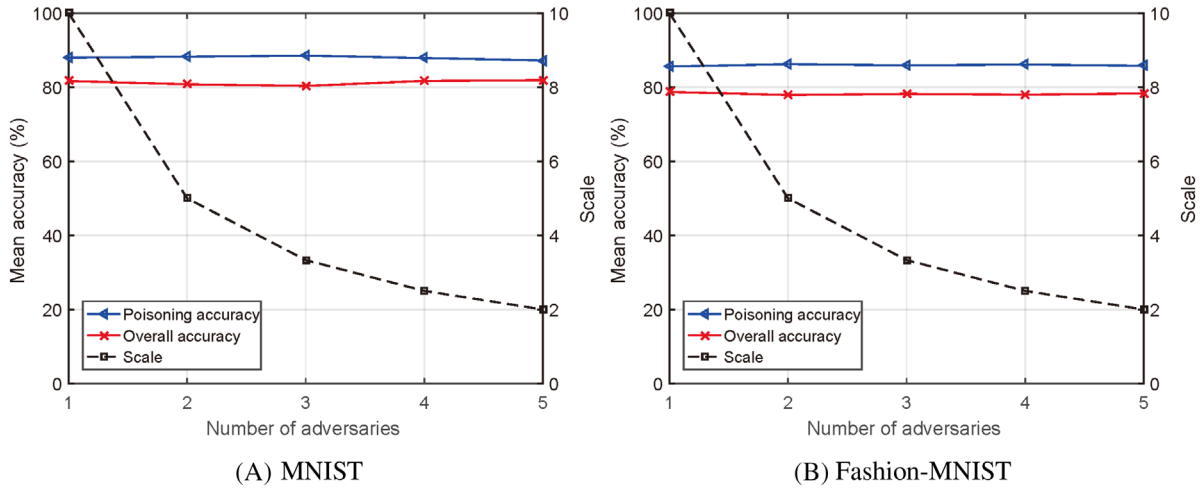


FIGURE 8 Impacts of different number of adversaries. A, MNIST. B, Fashion-MNIST

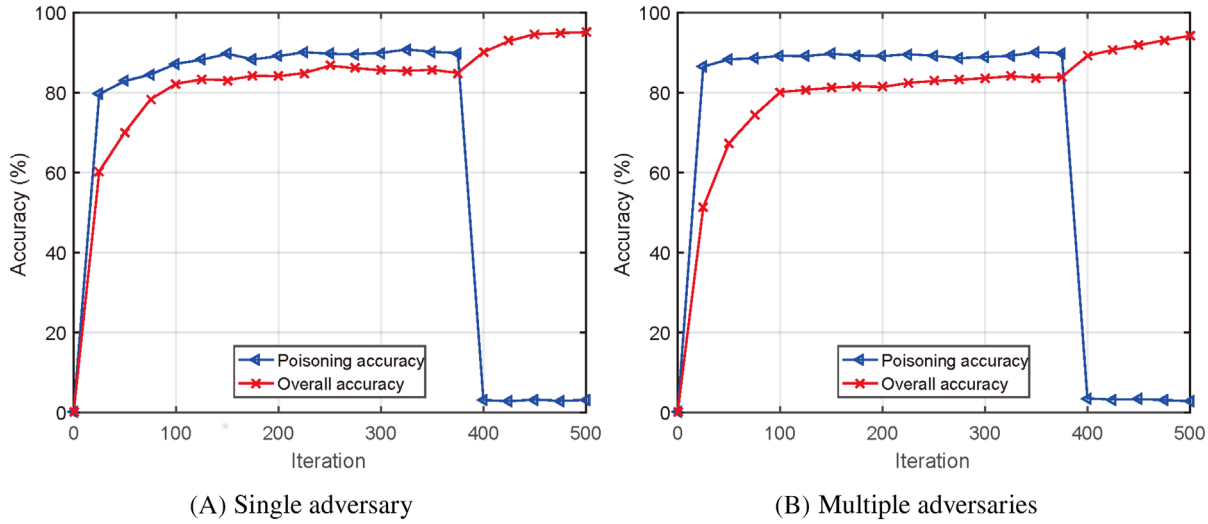


FIGURE 9 Poisoning defense on MNIST dataset. A, Single adversary. B, Multiple adversaries

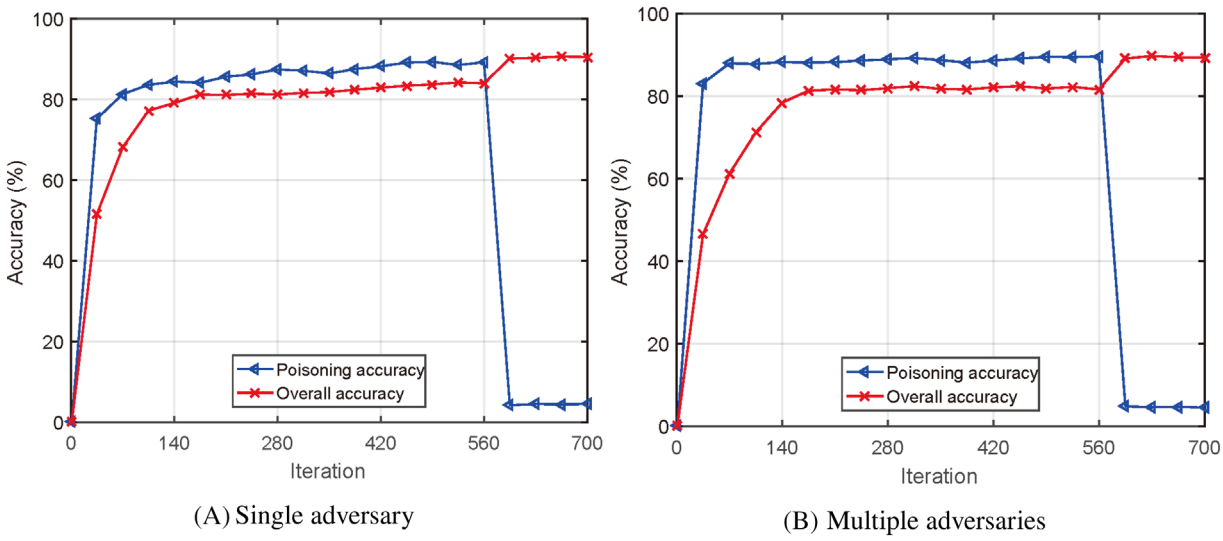


FIGURE 10 Poisoning defense on Fashion-MNIST dataset. A, Single adversary. B, Multiple adversaries

For the MNIST dataset, the *d_iter* is set to 400. As illustrated in Figure 9A, the proposed defense mechanism starts to detect and mitigate the poisoning attack at the 400th training iteration. There is a sharp increase in overall accuracy. At the same time, the poisoning accuracy decreases from 89.9% to 3.1%, which means the global model is only updated by benign model updates. From Figure 9B, we can see that the proposed approach also achieves an excellent defense performance under the multiple adversaries scenario.

For the Fashion-MNIST dataset, the proposed defense mechanism is activated at the 600th training iteration. As can be seen from Figure 10, poisoning accuracy rapidly decreases while overall accuracy sharply increases, which is similar to the trend in the MNIST dataset. The above experimental results indicate that the proposed defense mechanism can detect and mitigate poisoning attacks in federated learning.

Moreover, we use the GAN and model inversion to generate auditing data and compare their poisoning defense performance in the multiple adversaries scenario. The mean overall accuracy and mean poisoning accuracy are shown in Table 1. For the MNIST dataset, before eliminating the poisoning attack, the overall accuracy and poisoning accuracy are 80.33% and 88.54%, respectively. As the above experimental settings, the poisoning defense mechanism is activated at 400th iteration. In the GAN situation, the overall accuracy increases to 91.85%, and the poisoning accuracy decreases to 3.18%. In the model inversion situation, the overall accuracy decreases to 73.52%, and the poisoning accuracy decreases to 78.37%. The above experiments further verify that the data generation effect of GAN is better than that of model inversion. Furthermore, we can draw the same conclusion from the experimental results on the Fashion-MNIST dataset.

TABLE 1 Poisoning defense performance with different method

	MNIST		Fashion-MNIST	
Dataset	Overall accuracy	Poisoning accuracy	Overall accuracy	Poisoning accuracy
Before detection	80.33%	88.54%	78.21%	85.92%
GAN	**91.85%**	**3.18%**	**89.37%**	**4.65%**
Model inversion	73.52%	78.37%	74.87%	80.39%

6 | SUMMARY

As a collaborative learning framework, federated learning is widely used in the IoT. Meanwhile, federated learning is under the threat of poisoning attacks. To detect and mitigate poisoning attacks, we propose a novel defense method in this article. The proposed defense method deploys a GAN model in the server to generate the auditing dataset by reconstructing participants' private training data. With the generated auditing dataset, the proposed method checks the participant model accuracy to identify adversaries in federated learning. Experiments conducted on two well-known datasets suggest that federated learning is vulnerable to the poisoning attack, and the proposed defense method can detect and mitigate the poisoning attack.

ORCID

Junjun Chen https://orcid.org/0000-0001-8902-2553
Jiale Zhang https://orcid.org/0000-0002-2143-5666

REFERENCES

1. Ribeiro M, Grolinger K, Capretz MA. Mlaas: machine learning as a service. Paper presented at: Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA); 2015; Miami. https://doi.org/10.1109/ICMLA.2015.152.
2. Hesamifard E, Takabi H, Ghasemi M, Wright RN. Privacy-preserving machine learning as a service. *Proc Priv Enhanc Technol*. 2018;2018(3):123-142. https://doi.org/10.1515/popets-2018-0024.
3. Yu S. Big privacy: challenges and opportunities of privacy study in the age of big data. *IEEE Access*. 2016;4:2751-2763. https://doi.org/10.1109/ACCESS.2016.2577036.
4. Qu Y, Yu S, Zhou W, Peng S, Wang G, Xiao K. Privacy of things: emerging challenges and opportunities in wireless Internet of Things. *IEEE Wirel Commun*. 2018;25(6):91-97. https://doi.org/10.1109/MWC.2017.1800112.
5. Shokri R, Shmatikov V. Privacy-preserving deep learning. Paper presented at: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS); 2015; Denver. https://doi.org/10.1145/2810103.2813687.
6. McMahan HB, Moore E, Ramage D, Hampson S. Communication-efficient learning of deep networks from decentralized data. Paper presented at: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS); 2017; Fort Lauderdale. https://arxiv.org/abs/1602.05629.
7. Qu Y, Gao L, Luan T.H, Xiang Y, Yu S, Li B. Decentralized privacy using blockchain-enabled federated learning in fog computing. IEEE IoT J. 2020. https://doi.org/10.1109/JIOT.2020.2977383.
8. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. *ACM Trans Intell Syst Technol*. 2019;10(2):12:1-12:19. https://doi.org/10.1145/3298981.
9. Jagielski M, Oprea A, Biggio B, Liu C, Nita-Rotaru C, Li B. Manipulating machine learning: poisoning attacks and countermeasures for regression learning. Paper presented at: Proceedings of the IEEE Symposium on Security and Privacy (SP); 2018; San Francisco. https://doi.org/10.1109/SP.2018.00057.
10. Biggio B, Nelson B, Laskov P. Poisoning attacks against support vector machines. Paper presented at: Proceedings of the International Conference on Machine Learning (ICML); 2012; Edinburgh, UK. http://icml.cc/2012/papers/880.pdf.
11. Fung C, Yoon CJ, Beschastnikh I. Mitigating sybils in federated learning poisoning; 2018. arXiv preprint arXiv:1808.04866. https://arxiv.org/pdf/1808.04866.pdf.
12. Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning. Paper presented at: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS); 2017; Dallas. https://doi.org/10.1145/3133956.3133982.
13. Mohassel P, Zhang Y. Secureml: a system for scalable privacy-preserving machine learning. Paper presented at: Proceedings of the IEEE Symposium on Security and Privacy (SP); 2017; San Jose. https://doi.org/10.1109/SP.2017.12.
14. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans Inf Foren Sec*. 2018;13(5):1333-1345. https://doi.org/10.1109/TIFS.2017.2787987.
15. Blanchard P, Mhamdi E.M.E, Guerraoui R, Stainer J. Machine learning with adversaries: byzantine tolerant gradient descent. Paper presented at: Proceedings of the Advances in Neural Information Processing Systems (NIPS); 2017; Long Beach https://doi.org/10.5555/3294771.3294783.
16. Yin D, Chen Y, Ramchandran K, Bartlett P. Byzantine-robust distributed learning: towards optimal statistical rates. Paper presented at: Proceedings of the International Conference on Machine Learning (ICML); 2018; Stockholm, Sweden. http://proceedings.mlr.press/v80/yin18a/yin18a.pdf.
17. Baracaldo N, Chen B, Ludwig H, Safavi, JA. Mitigating poisoning attacks on machine learning models: a data provenance based approach. Paper presented at: Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec); 2017; Dallas. https://doi.org/10.1145/3128572.3140450.
18. Shen S, Tople S, Saxena P. Auror: defending against poisoning attacks in collaborative deep learning systems. Paper presented at: Proceedings of the Annual Computer Security Applications Conference (ACSAC); 2016; Los Angeles. https://doi.org/10.1145/2991079.2991125.

19. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Paper presented at: Proceedings of the Advances in Neural Information Processing Systems (NIPS); 2014; Montreal, Canada. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.
20. Zhao Y, Chen J, Wu D, Teng J, Yu S. Multi-task network anomaly detection using federated learning. Paper presented at: Proceedings of the Symposium on Information and Communication Technology (SoICT); 2019; Ha Noi - Ha Long Bay, Viet Nam. https://doi.org/10.1145/3368926.3369705.
21. Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning; 2018. arXiv preprint arXiv:1807.00459. https://arxiv.org/pdf/1807.00459.pdf.
22. Bhagoji AN, Chakraborty S, Mittal P, Calo S. Analyzing federated learning through an adversarial lens. Paper presented at: Proceedings of the International Conference on Machine Learning (ICML); 2019; Long Beach, US. http://proceedings.mlr.press/v97/bhagoji19a/bhagoji19a.pdf.
23. Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H. Beyond inferring class representatives: user-level privacy leakage from federated learning. Paper presented at: Proceedings of the IEEE Conference on Computer Communications (INFOCOM); 2019; Paris, France. https://doi.org/10.1109/MWC.2017.1800112.
24. Qu Y, Yu S, Zhang J, Binh H.T.T, Gao L, Zhou W. GAN-DP: generative adversarial net driven differentially privacy-preserving big data publishing. Paper presented at: Proceedings of the International Conference on Communications (ICC); 2019; Shanghai, China. https://doi.org/10.1109/ICC.2019.8761070.
25. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324. https://doi.org/10.1109/5.726791.
26. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms; 2017. arXiv preprint arXiv:1708.07747. https://arxiv.org/pdf/1708.07747.pdf.
27. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. Paper presented at: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS); 2015; Denver. https://doi.org/10.1145/2810103.2813677.
28. Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. Paper presented at: Proceedings of the 23rd USENIX Security Symposium; 2014:17-32. https://www.usenix.org/system/files/conference/usenixsecurity14/sec14-paper-fredrikson-privacy.pdf.

How to cite this article: Zhao Y, Chen J, Zhang J, Wu D, Blumenstein M, Yu S. Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks. *Concurrency Computat Pract Exper*. 2020;e5906. https://doi.org/10.1002/cpe.5906