Computers & Security 132 (2023) 103366

Contents lists available at ScienceDirect

Computers & Security

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

ADFL: Defending backdoor attacks in federated learning via adversarial distillation

Chengcheng Zhu[a], Jiale Zhang[a,*], Xiaobing Sun[a], Bing Chen[b], Weizhi Meng[c]

[a] School of Information Engineering, Yangzhou University, Yangzhou, 225009, China
[b] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China
[c] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Copenhagen, Denmark

ARTICLE INFO

Article history:
Received 5 December 2022
Revised 12 June 2023
Accepted 22 June 2023
Available online 26 June 2023

Keywords:
Federated learning
Backdoor attack
Generative adversarial network
Knowledge distillation
Backdoor defense

ABSTRACT

Federated learning enables multi-participant joint modeling with distributed and localized training, thus effectively overcoming the problems of data island and privacy protection. However, existing federated learning frameworks have proven to be vulnerable to backdoor attacks, where attackers embed backdoor triggers into local models during the training phase. These triggers will be activated by crafted inputs during the prediction phase, leading to misclassification targeted by attackers. To address these issues, existing defense methods focus on both backdoor detection and backdoor erasing. However, passive backdoor detection methods cannot eliminate the effect of embedded backdoor patterns, while backdoor erasing may degenerate the model performance and cause extra computation overhead. This paper proposes ADFL, a novel adversarial distillation-based backdoor defense scheme for federated learning. ADFL generates fake samples containing backdoor features by deploying a generative adversarial network (GAN) on the server side and relabeling the fake samples to obtain the distillation dataset. Then, taking the labeled samples as inputs, knowledge distillation which employs the clean model as a teacher and the global model as a student is implemented to revise the global model and eliminate the influence of backdoored Neurons in it, thereby effectively defending against backdoor attacks while maintaining the model performance. Experimental results show that ADFL can lower the attack success rates by 95% while maintaining the main task accuracy above 90%.

1. Introduction

Recently, with the explosive growth of the number of user terminals, the centralized machine learning mode under the traditional cloud computing architecture is facing serious challenges due to its defects such as high delay, high concurrency and weak privacy protection Bonawitz et al. (2017b); McMahan et al. (2017); Yang et al. (2019). As a new AI paradigm, federated learning enables multiple participants to collaboratively train a global model in a distributed manner, which is an emerging technology to solve data islands and privacy protection in machine learning. Worldwide leading enterprises have also launched a variety of federated learning application frameworks, such as TensorFlow Federated (TFF) Bonawitz et al. (2017a) developed by Google and FATE Liu et al. (2021) of Webank. Overall, federated learning has shown great practical development potential, which has attracted wide attention of scholars all over the world Aledhari et al. (2020); Lim et al. (2020); Lyu et al. (2020); Yin et al. (2021).

However, the participants in federated learning may be any relevant network devices in the application scenarios with a large number of users and massive data where these devices are not necessarily trusted entities. In addition, due to the localization and distributed training nature of federated learning, the model training process is invisible to the central server, and it is difficult to achieve effective supervision of local training. This severely restricted the effectiveness of anomaly detection on the server side. Therefore, the standard federated learning framework is vulnerable to adversarial attacks launched by untrusted parties Bhagoji et al. (2019); Song et al. (2020). Specifically, the attack methods usually affect the accuracy of the global model by abnormal local updates (malicious gradients), such as backdoor attacks Bagdasaryan et al. (2020), poisoning attacks Fang et al. (2020); Zhang et al. (2020), and adversarial example attacks Luo et al. (2018).

* Corresponding author.
E-mail addresses: mx120220554@stu.yzu.edu.cn (C. Zhu), jialezhang@yzu.edu.cn (J. Zhang), xbsun@yzu.edu.cn (X. Sun), cb_china@nuaa.edu.cn (B. Chen), weme@dtu.dk (W. Meng).

https://doi.org/10.1016/j.cose.2023.103366
0167-4048/

Backdoor attack is a typical adversarial attack type against deep neural networks (DNN). Gu et al. Gu et al. (2017) first explored the backdoor attack method against DNN, called Badnets. In the training phase, attackers try to embed hidden triggers in DNN. The backdoored DNN performs well on benign samples. However, the model will output the label desired by the attacker when taking crafted samples that can activate the hidden backdoor triggers as inputs. Currently, the commonly used backdoor trigger types include pixels, sinusoidal stripes, natural attributes, and imperceptible noise Li et al. (2021). The selection of these backdoor triggers makes backdoor attacks more hidden and difficult to detect even if involving human intervention. Moreover, backdoor attacks can also strengthen the connection between backdoor triggers and DNN inter neurons through pre-training Yao et al. (2019) or knowledge distillation Liu et al. (2018). These neurons have a strong activation effect on the backdoor triggers in the prediction phase, thus further enhancing the concealment and harmfulness of backdoor attacks.

Backdoor attacks in federated learning differ from those in centralized learning. Specifically, in centralized learning scenarios, backdoor attacks are usually implemented by data poisoning Lin et al. (2020). For example, in the car and aircraft classification task of the CIFAR-10 dataset, an attacker can tamper all “green cars” in the training data as “aircraft” and train the model on these crafted datasets to make the model classify “green cars” as “aircraft” in the prediction phase. However, in federated learning scenarios, the attackers cannot access the training data since the data is distributed locally among various participants. Therefore, backdoor attacks in federated learning are usually carried out by model poisoning Sun et al. (2019); Wang et al. (2020); Xie et al. (2020); Zhang et al. (2019), i.e., the attacker injects backdoor into the local model in the local training phase to generate local updates containing malicious backdoor patterns. When the backdoored updates aggregate with local updates of other participants, the newly generated aggregate model for the next federated communication round will possess the inherent characteristics of the backdoored model.

Recently, many researchers began to conduct defense research from different angles. For now, the defense intuitions of backdoor attacks mainly come in two types: backdoor detection and backdoor erasing. Firstly, the backdoor detection aims to identify whether there are backdoor in the target model Andreina et al. (2021); Wang et al. (2019), or directly filter suspicious samples from the training data for retraining Chen et al. (2019). These defense methods based on passive detection can only judge whether the neural network model suffers from backdoor attacks, but is unable to eliminate the negative impact of backdoor attacks on the target model. Therefore, researchers began to explore how to purify the backdoor model by eliminating the backdoor trigger meanwhile maintain the high performance of the model on benign samples Zhang et al. (2021). At present, the defense methods for erasing triggers are mainly to fine-tune the model on a part of clean datasets, which reduces the possible overfitting phenomenon during the fine-tuning process by using model pruning and other methods Liu (2018). In addition, some methods such as data augmentation, regularization, and model distillation Truong et al. (2020) have also been proposed in succession to reduce the effect of backdoor attacks. However, these backdoor erasing methods will reduce the classification accuracy of the main task and cause extra computation and communication overhead Ozdayi et al. (2021); Xie et al. (2021).

Therefore, given the above problems, this paper proposes a novel backdoor defense scheme in federated learning, called ADFL, based on adversarial distillation. Firstly, we deploy the GAN model on the server side to generate fake samples, which contain backdoor features. Then, the clean model trained on the clean datasets hold by the server labels the generated samples to obtain the distillation data. Finally, we perform knowledge distillation that takes the distillation data as inputs and employs the clean model to guide the global model thereby eliminating the influence of backdoor parameters in the global model. The contributions of this paper can be summarized as follows:

- We explore a distillation data generation method based on generation adversarial networks. By deploying the GAN model on the server, we generate fake samples and relabel them through a clean model to obtain the distillation datasets.
- To defend against backdoor attacks in federated learning, we proposed a novel backdoor defense method based on adversarial distillation, which overcomes the dependence on accounts of clean datasets and can eliminate the backdoor influence rather than be limited to backdoor detection.
- We conduct extensive experiments on four benchmark datasets. Firstly, we verify the effectiveness of the backdoor attacks based on pixels, watermarks, and attributes in federated learning. Then, we evaluate the performance of ADFL as compared with similar approaches. Experimental results illustrate that ADFL can effectively lower the attack success rates by 95% while maintaining the main task accuracy above 90%.

This paper will be organized as follows. Section 2 introduces the related works of backdoor attacks and defenses. The preliminaries are introduced in Section 3. Methods and algorithms of our proposed ADFL are discussed in Section 5. Section 6 details on our experimental evaluation of ADFL. Section 7 concludes this paper.

2. Related works

2.1. Backdoor attacks in federated learning

Gu et al. Gu et al. (2017) first explored a backdoor attack method called Badnets. Followed by Badnets, many new backdoor attacks were proposed and the attack scenarios were converted from traditional deep learning into federated learning. For example, Wang et al. Wang et al. (2020) established that robustness to backdoors implies model robustness to adversarial examples and proposed edge-case backdoors which force a model to misclassify on seemingly easy inputs not belong to part of the training or test data. Zhang et al. Zhang et al. (2020) introduced GAN into backdoor attacks. An attacker first acts as a benign participant and stealthily trains a GAN to mimic prototypical samples of the other participants training set. After that these generated samples will be fully controlled by the attacker to train the backdoored models. Different from centralized backdoor attacks above where each party embeds the same global trigger during training, the distributed backdoor attack (DBA) was proposed by Xie et al. Xie et al. (2020), which fully exploits the distributed nature of federated learning. DBA decomposes a global trigger pattern into separate local patterns which will be embedded into the training set of different adversarial parties respectively.

2.2. Defense solutions

For defending the aforementioned backdoor attacks in federated learning, several countermeasures were proposed, which can be roughly classified into backdoor detection Awan et al. (2021) and backdoor elimination Ozdayi et al. (2021); Xie et al. (2021). Awan et al. Awan et al. (2021) introduced CONTRA, a cosine-similarity-based measurement method to evaluate the reliability of local model parameters in every iteration. Then, a reputation scheme is utilized to adjust the standing of individual clients dynamically, depending on their historical and per-round contributions

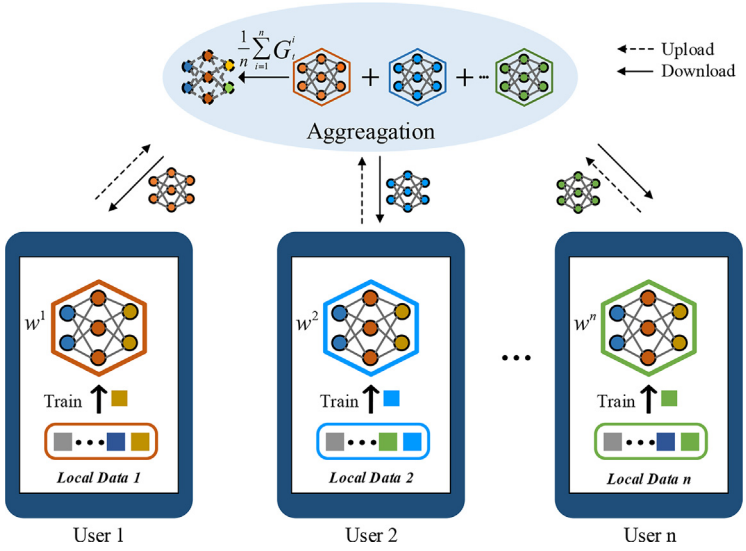


Fig. 1. The architecture of federated learning.

to the global model. Andreina et al. Andreina et al. (2021) presented Baffle, a round-based feedback loop engaging clients in validating the global model, which exploits the availability of diverse datasets at the various clients by incorporating a feedback loop into the federated learning process, to integrate the views of those clients when deciding whether a given model update is genuine or not. CRFL was proposed by Xie et al. Xie et al. (2021), which exploits clipping and smoothing on parameters to keep the smoothness of the global model, yielding a sample-wise robustness certification on backdoors. Different from them, Ozdayi et al. Ozdayi et al. (2021) defend against backdoor attacks by adjusting the aggregation server's learning rate, per dimension and per round based on the sign information of the agent's updates, which is considered a lightweight defense since it just requires little change to federated learning protocol.

3. Preliminary

3.1. Federated learning

Federated learning was first proposed by Google in 2017 which is a collaborative machine learning framework that protects the privacy of users. Its design goal is to train a joint machine learning model on the training data of different participants while ensuring the privacy of the training data of all participants. To achieve this goal, federated learning trains a global model in a decentralized manner, allowing each participant to download the global model, and use the local data to train the model and further update the parameters. Finally, these updated parameters will be sent to the server for aggregation and averaging. Fig. 1 shows the standard framework of federated learning. Specifically, before the federated learning training starts, the server first initializes a global model G_0 according to the service requirements of the participants and sends the model to the n participants. Once receiving the initialization model, each participant trains the model on its local dataset within its local epochs. Then the updated local model parameters G_t^k are uploaded to the server. The aggregation rule of the Fedavg algorithm are as follows:

$$G_{t+1} = \sum_{k \in S_t} \frac{D_k}{D_{S_t}} G_t^k. \tag{1}$$

where t represents the training round of federated learning, $k \in S_t$ represents the k-th participant, D_{S_t} is the total sample number of the selected clients in communication round t. The above process is performed iteratively until the global model initialized by the server converges.

3.2. Generative adversarial nets

Generative adversarial network Goodfellow et al. (2014) was proposed by Goodfellow et al in 2014. It is structurally inspired by the zero-sum game in game theory, that is, under strict competition, the benefits of one party will necessarily mean the losses of the other party, and the sum of the benefits and losses of all parties in the game will always be "zero". Since GAN has the ability to generate the same distribution samples through existing data samples, it has been widely used in data enhancement and other fields in recent years. Specifically, the generator G generates fake samples with the same distribution as the real data by inputting random noise Z_{noise}, and the discriminator D judges $G(z)$ to calculate the probability that they are real samples $D(G(z))$. In order to make the generated samples closer to the real samples, the value of the probability is required to be as large as possible. Therefore, the objective function of the generator is:

$$\mathcal{L}_G(\theta_g) = \mathbb{E}_{z \sim p(z_{\text{noise}})}[\log(D(G(z)))]. \tag{2}$$

At the same time, the discriminator inputs the real sample x and the generated sample $G(z)$ and gives the judgment result $G(x)$ and $D(G(z))$. The purpose is to realize the two classification judgment of the source of the input sample, that is, true (from the real sample) or false (from the generator sample). The objective function of the discriminator is:

$$\begin{aligned}\mathcal{L}_D(\theta_d, \theta_g) = {} & \mathbb{E}_{z \sim p(X_{\text{real}})}[\log(D(x))] \\ & + \mathbb{E}_{z \sim p(z_{\text{noise}})}[\log(1 - D(G(z)))].\end{aligned} \tag{3}$$

The performance of the generator and the discriminator is constantly improved through the confrontation game and iterative optimization, further reaching the Nash equilibrium. Finally, the discriminator cannot correctly judge whether the data is true or not, and the generator is considered to be able to generate data with the same distribution as the real sample.

3.3. Knowledge distillation

Generally speaking, large-scale deep learning models can often obtain higher classification accuracy. However, these complex models with a large number of parameters are difficult to deploy on

low-resource devices with limited computation capacity and storage capacity. To address this problem, many scholars have carried out plenty of research. Hinton et al. first proposed the concept of knowledge distillation Hinton et al. (2015). The core idea of compressing a large-scale neural network model with deeper layers and more nodes into a model with smaller parameters is to guide a relatively weak student model through a teacher model. Normally, the teacher model is complex and powerful, where the student model can learn the output logits and ground truth of the teacher model at the same time to realize knowledge transfer. Traditional neural networks usually use the "softmax" output layer to generate class probabilities, and the purpose of knowledge distillation is to make the softmax output of the student model and the teacher model sufficiently close. To achieve this, knowledge distillation introduces the softmax function with temperature parameters, which is specifically defined as:

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}, \tag{4}$$

where Z_i is the logits of the model and T is the temperature factor. When $T = 1$, q_i is the standard softmax function. When $T \leftarrow 0$, the maximum value of each category will be close to 1, and the other values will be close to 0, which is similar to one-hot coding. In this situation, the results output by the softmax layer will be more distributed and more information between and within classes will be retained with the increase of the temperature factor. According to the above properties, when knowledge distillation is implemented, the logits output by the teacher model and the student model will be processed with a higher temperature factor to obtain a soft target. Let p_j^T and q_j^T denote the output soft target of the teacher model and the student model after being "softened" under the temperature T, and let L denote a standard cross entropy loss which is used to measure the direct distribution difference between p_j^T and q_j^T. The loss function of the soft target is as follows:

$$L_{soft} = -\sum_j^N p_j^T \log\left(q_j^T\right). \tag{5}$$

Since the teacher model also has a certain error rate, the use of ground truth can effectively reduce the possibility of errors being transmitted to the student model. c_j is defined as the value of ground truth in the jth class. The positive label takes 1 and the negative label takes 0. The loss function of the hard target is as follows:

$$L_{hard} = -\sum_j^N c_j \log\left(q_j^1\right). \tag{6}$$

Combining the loss of soft and hard target, the total object function of knowledge distillation:

$$L = L_{\text{hard}} + \alpha L_{\text{soft}}, \tag{7}$$

where α is a hyperparameter balancing the two terms.

4. Threat model and defense goal

In this section, we introduce the threat model of backdoor attacks in federated learning, as well as the defense goal of the proposed ADFL scheme.

4.1. Threat model

Given clean local samples, the attacker aims to pollute a portion of them by adding specific triggers and tricks the model to output desired target label when taking the polluted samples as input. Meanwhile, the backdoor model can perform normally on clean samples. In FL setting, followed by recent backdoor attack methods Bagdasaryan et al. (2020); Wang et al. (2020), we define the capabilities of attackers as follows:

- Attackers can manipulate the process of the training phase where they can plant the trigger into the benign samples and assign them with the desired label.
- Attackers can upload the crafted parameters and replace the federated model with a backdoored model.

However, they must keep the structure of the global model and comply with established agreements. Besides, the attacker has no access to interference with the server.

4.2. Defense goal

For the defender who is acted by the server, similar to recent backdoor defense solutions and other FL settings Fang and Ye (2022); Park et al. (2021); Zhang et al. (2022); Zhao et al. (2022), we assume the defender's ability as follows:

- The defender can only obtain local model updates and knows nothing about the backdoor knowledge.
- The defender can only obtain a limited portion of clean data rather than the whole training set.

We claim that the assumption of clean data is not newly used in our method, which has been widely adopted in FL-related research fields, such as poisoning attack defense and heterogeneous data process, et al. In fact, having a public dataset is indispensable to the design of neural network architecture in FL. For example, training an FL-based face recognition system can use public datasets like AT&T, VGG Face, et al. to pretrain the joint model, making the global model converge faster. Besides, Cao et al. (2021) assume the root datasets sampled from the union of the clients clean local training data at random, which essentially is the clean dataset. Moreover, Zhou et al. (2023) assume that the server can obtain a small set of benign samples covering all categories, e.g., from public sources, as an auxiliary dataset to execute inference attacks. However, in this paper, we focus on defending backdoor attacks from the client side. Therefore, we assume the server is honest and non-curious about the privacy inference attack, the main aim of the server is to obtain a global model whose utility/integrity is not breached by malicious users. Besides, the server is definitely with sufficient computing resources as well as a fast connection to the Internet. Therefore, such a setting is in accord with real-world scenarios. Finally, we define the defense goal as the proposed can significantly reduce the backdoor attack success rate (ASR) on backdoored samples while maintaining high model accuracy on normal samples (ACC).

5. Method

To achieve an effective defense against backdoor attacks in federated learning without degenerating the performance of the global model, we obtain distillation datasets by relabeling the fake samples generated by GAN. Then, we revise the global model in combination with knowledge distillation to erase the negative impact of backdoor parameters in the global model. In this section, we first describe the backdoor attack model in federated learning. Subsequently, we detail a GAN-based distillation dataset generation algorithm. Finally, the defense method of backdoor attack in federated learning based on adversarial distillation (ADFL) is explained.

5.1. Backdoor attack model in federated learning

The workflow of the backdoor attack in federated learning is shown in Fig. 2. It can be roughly divided into two phases. In the training phase, an attacker trains a backdoor model with specific attributes, which are defined as backdoor triggers, on crafted backdoor data. Once the model is embedded in these backdoor triggers,

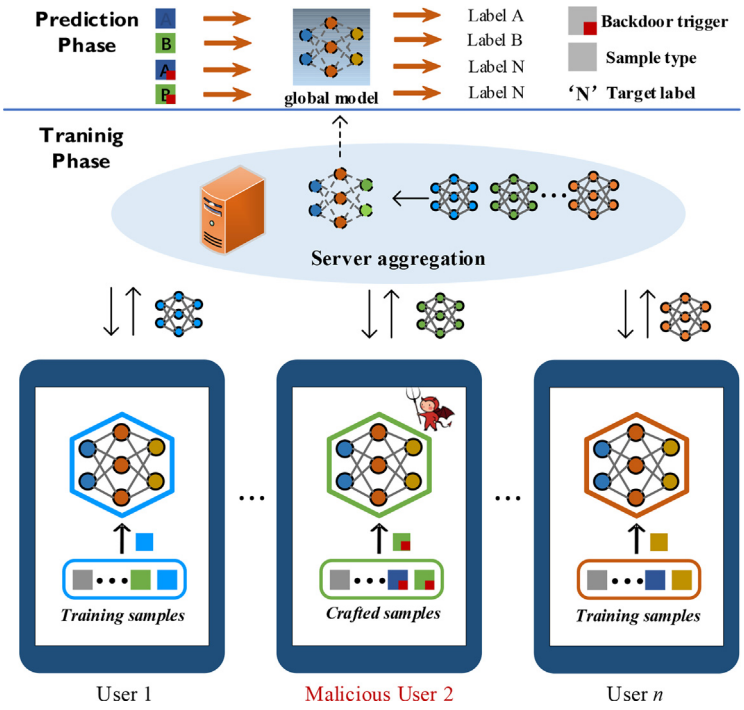


Fig. 2. Workflow of backdoor attack in federated learning.

it will be misled by crafted input in the prediction phase. In more detail, the attacker embeds backdoor triggers (such as the red pixel block in the lower right corner of Fig. 2) in the samples assigned with the desired label "*N*" and mixes them with normal samples. After training on the tampered dataset, the local model performs normally in predicting the normal samples while making the prediction specified by the attacker for the input samples with backdoor triggers. Subsequently, the malicious user uploads the backdoored model to the server. To maintain the effect of backdoor update parameters in the global model for a long time in following federated communication, the attacker can enlarge the backdoor update parameters by several times through the scaling mechanism to enhance the contribution of backdoor update in the global model Bagdasaryan et al. (2020). After the federated averaging algorithm, the new global model will also misclassify the samples containing backdoor triggers. As shown in Fig. 2, in the prediction phase, the samples embedded with backdoor triggers are classified as "*N*', while those without backdoor triggers are still classified as their original labels.

5.2. GAN-Based distillation dataset generation

Most of the existing backdoor defense methods need to detect the training dataset or supervise the training process on the server side. However, in federated learning, due to privacy and security protection, the server cannot directly access the local data or interfere with the local training, which inevitably brings great difficulties to the defense against backdoor attacks. Therefore, we proposed a GAN-based distillation dataset generation algorithm that GAN model is employed to generate fake samples (including backdoor) identically distributed with the real samples, and uses a clean model to relabel the generated samples to obtain distillation datasets, which will be used for backdoored model purify.

The GAN includes two models: the generation model and the discrimination model. Generally speaking, the discriminator in the GAN requires real samples as inputs to update the discriminator. Fortunately, this constraint is broken in the scenario of federated learning since the global model is aggregated by the local models which are trained by each participant using their local datasets. Naturally, the server can directly use the obtained global model to update the discriminator. This method can also promote the generator to generate fake samples that are the same distribution as the local real training samples of all participants. It is worth noting that if there are malicious users using datasets embedded in backdoor triggers for training among the local users, the samples generated by the generator will also contain backdoor triggers.

Fig. 3 describes the structure of the GAN model in the federated learning scenario. The global model is optimized with the continuous iteration of communication rounds for federated learning and finally converges. Notably, the global model and the discriminator model will adopt a network structure with different outputs but the same number of layers to ensure the discriminator model will be updated synchronously through the global model parameters. Among them, the output of the global model is the classification task result of the federated learning, while the output of the discriminator model is the authenticity of the samples generated by the generator. In this process, the generator continuously optimizes through the game with the discriminator until the discriminator cannot judge the difference between the generated samples and the real samples.

Subsequently, we relabel the generated datasets before implementing knowledge distillation. Assuming that the server has a small proportion of clean data, a light, and simple model was trained, which is just required to have the ability to roughly classify the unlabeled samples. Note that the generated datasets consist of both clean samples and backdoor samples containing triggers whose proportion is related to the extent of backdoor attacks. However, since the clean model has never touched the backdoor samples meaning that it is insensitive to the embedded backdoor triggers, it is not vulnerable to the impact of backdoor attacks and causes misclassification.

Algorithm 1 formally describes the process for generating distillation datasets based on GAN. Specifically, in each round of communication of federated learning, the server obtains the global

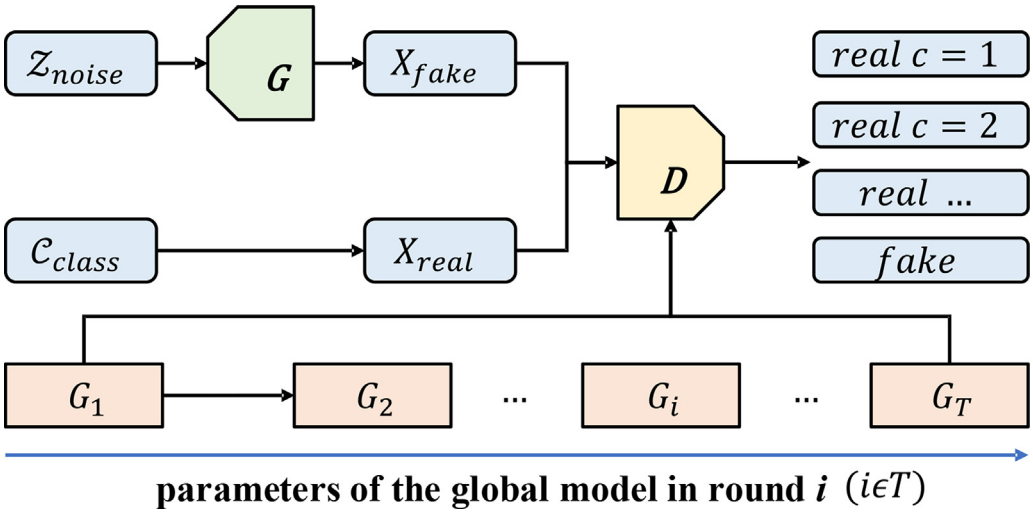


Fig. 3. GAN model of federated learning.

Algorithm 1: GAN-based Distillation Data Generation Algorithm.

Input: Global model $\tilde{G}_t$, random noise z, clean data D_{clean}.
Output: Generated dataset for distillation $D_{distillation}$
// **Adversarial Sample Generation**
Initialize generator $G(x)$ and discriminator $D(x)$
Update $G(x)$ through global model: $D(x) \leftarrow \tilde{G}_t$
$\mathcal{B}_{gan} \leftarrow$(split D_{clean} into batches of size B_{kd})
for *GAN epoch* $k \in [1, E_{gan}]$ **do**
  for *GAN batch* $b \in \mathcal{B}_{gan}$ **do**
    Update $G(x)$ and $D(x)$ based on Eq. 2 and Eq. 3;
  end
end
Return $G(x)$.
// **Relabel**
Initialize clean model G_{clean}
$\mathcal{B}_{clean} \leftarrow$(split D_{clean} into batches of size B_{clean})
for *local epoch* $k \in [1, E_{clean}]$ **do**
  for *local batch* $b \in \mathcal{B}_{clean}$ **do**
    Clean Model Training: $G_{clean} = G_{clean} - \eta \nabla \ell(G_{clean}, b)$;
  end
end
Use $G(x)$ to get generated data:$x_{fake} = G(z)$
Use clean model to label generated data:$y = f(G_{clean}, x_{fake})$
Add (x_{fake}, y) into $D_{distillation}$: $D_{distillation} \leftarrow (x_{fake}, y)$**Return** $D_{distillation}$.

model G_t by aggregating the local models of each participant. Then the parameters of the discrimination model and generator model are installed. The generator generates samples X_{fake} by inputting random noise z and transmits them to the discriminator. Meanwhile, the discriminator judges whether they are real samples. In the process of the adversarial game, the parameters are updated based on Eq. 2 and Eq. 3 until both of the models converge.Subsequently, in the data distillation phase, a small number of clean data D_{clean} are used to train the clean model while the generated model is used to generate a set of fake samples containing backdoor samples X_{fake}, which are input to the clean model. The clean model takes X_{fake} as inputs and outputs the prediction y employed as the labels of the generated datasets. Finally, the (x_{fake}, y) is added to the distillation datasets $D_{distillation}$.

5.3. Backdoor defense via adversarial distillation

Based on the above distillation dataset generation algorithm, we further propose a novel backdoor defense method ADFL. Fig. 4 describes the framework of ADFL. By deploying the GAN on the server side, we generate fake samples containing backdoor triggers and then use the clean model to relabel generated samples. Finally, taking these distillation datasets as inputs, the clean model guides the backdoored model (global model) through knowledge distillation, weakening the effect of backdoor parameters of the global model and finally realizing the defense against backdoor attacks. The specific steps of ADFL are as follows:

Step 1: each participant downloads the global model from the server and trains the model on the local dataset. The malicious participants configure the backdoor samples and train the backdoored model on the crafted dataset. Finally, all participants upload the model updates to the server.

Step 2: the server receives the updates of each participant, and performs federated averaging to obtain the global model for next communication round.

Step 3: the server uses the parameters of the global model to update the discriminator in the generative adversarial network according to Algorithm 1, and obtains the generator through the adversarial game. Subsequently, the server uses the generator to generate datasets containing backdoor triggers and then relabels the generated datasets through the clean model to obtain the distillation datasets.

Step 4: according to Algorithm 2, taking the distillation dataset obtained by Algorithm 1 as the input, the server employs the clean model trained from a small amount of clean data as the teacher model and the backdoored model as the student model, executing the knowledge distillation algorithm, so as to purify the backdoored model. Finally, the purified model is employed as a new global model and sent to all participants for the next communication round.

Note that the operation of GAN on the server is parallel to the local training of the local client. More specifically, we will deploy GAN (i.e. execute step 3) after the accuracy of the global model has reached an expected value or the round of federated learning has reached a certain specified round T_{GAN} so that GAN can achieve a stable training effect. To slow down the delay, we first aggregate the global model (i.e. execute step 2) and immediately distribute the aggregated model. Once receiving the global model, the clients can train the local model for the next round. Meanwhile, the server

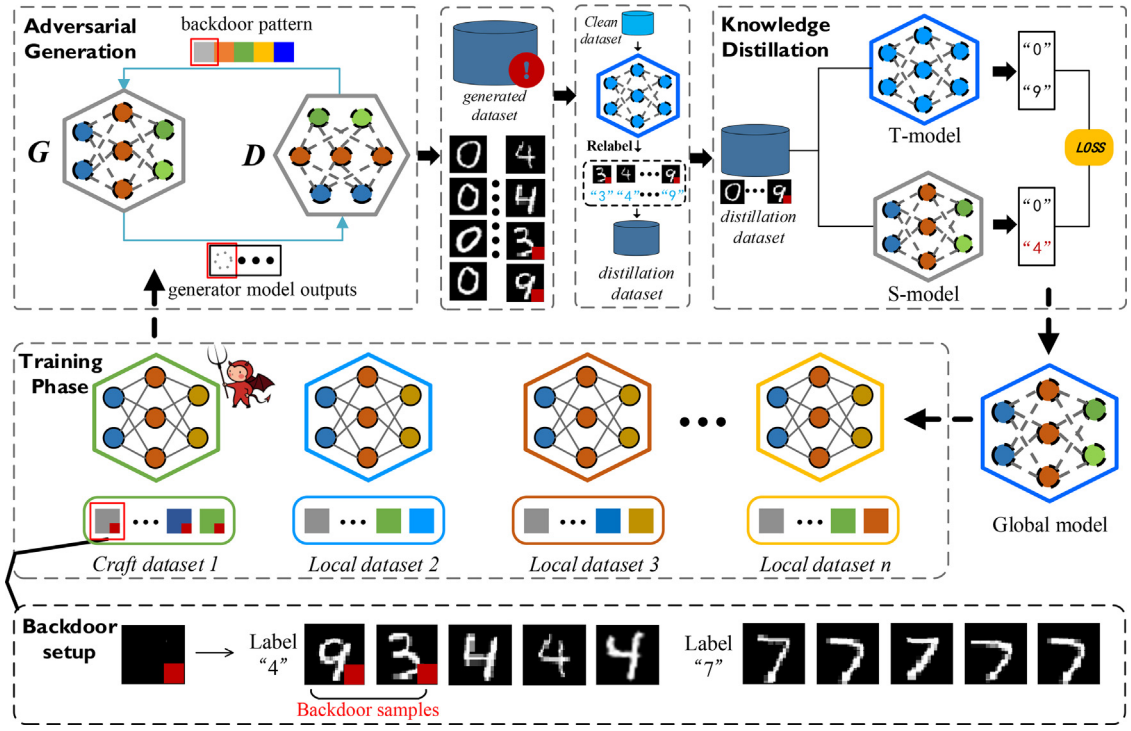


Fig. 4. The architecture of ADFL.

train GAN networks and update the discriminator with the global model of the following rounds to ensure that the backdoor samples can be generated. Such settings mean that clients do not have to wait for the server to complete GAN training, thereby reducing communication latency. Moreover, we will not execute knowledge distillation (i.e. execute step 4) until the effect of GAN reaches the expectation. In fact, we only need to perform knowledge distillation in the last few or even one round of federated learning since the accuracy of the global model is limitedly affected by knowledge distillation.

6. Experimental evaluation

In this section, we evaluate the proposed ADFL scheme from three aspects: the effectiveness of backdoor attacks in federated learning, the effectiveness of distillation datasets generation, and the defense performance of ADFL. We implement pixel backdoor attack, watermark backdoor attack, and attribute backdoor attack on MINST and CIFAR-10 datasets and verify the effectiveness of the ADFL under these attacks.

6.1. Datasets and experimental setup

6.1.1. Datasets

To validate the effectiveness of backdoor attacks and the proposed ADFL scheme in federated learning, different classification tasks are constructed on the following five datasets:

MNIST dataset: this dataset contains 70,000 handwritten digital images of 10 classes (0–9), including 60,000 training samples and 10,000 test samples. In all experiments, samples MNIST dataset were normalized to 28 $\times$ 28 pixels. For the pixel block backdoor attack, the attacker embeds 5 $\times$ 5 pixel block in the samples and assigns them with the target label "1" desired by attackers. In the watermarking backdoor attack experiment, the attacker added the watermarking "1" with a different watermarking factor to some real samples Zheng et al. (2022), and set its label as "1".

Fashion-MNIST dataset: Fashion-MNIST has the same data size and data scale as MNIST. In contrast, Fashion-MNIST is more complex which contains 10 categories of fashion goods.

EMNIST dataset: EMNIST is the extension of MNIST, a more challenging benchmark in classification tasks. In our experiments, MNIST and EMNIST datasets are used on non-independent and identically distributed (Non-IID) scenarios.

CIFAR-10 dataset: the CIFAR-10 dataset contains 60,000 color images in 10 categories (such as "aircraft", "car", "bird", etc.), including 50,000 training samples and 10,000 test samples. The images in the CIFAR-10 dataset are normalized to a 32 $\times$ 32 three-channel input during data preprocessing. For the attribute backdoor attack experiment, "car" in the CIFAR-10 dataset, "cars with stripes", "cars next to striped walls" and "green cars" were selected as the attribute backdoor which is the backdoor triggers Bagdasaryan et al. (2020).

CelebFaces Attributes Dataset: CelebA is a large-scale face attributes dataset, which contains 10,177 identities with 202,599 face images. Followed by prior work Nguyen and Tran (2021), we select 3 out of 40 attributes, namely Heavy Makeup, Mouth Slightly Open, and Smiling, and concatenate them into 8 classes to create a multiple label classification task. The input images were all resized into 64 $\times$ 64 pixels.

6.1.2. Experimental settings

We designed a federated learning prototype system based on the TensorFlow 2.6 framework. The hardware environment is lntel i5-10600kf (4.10Ghz) CPU, RTX2080 (8GB), and 64GB memory. The software environment is the window10 operating system, python 3.6, and TensorFlow 2.6 + Cudn10.1. In this experiment, we set up a total of 100 federated learning participants (including malicious participants) and divided the original dataset into 100 pieces, which are distributed to each participant. Notably, the number of samples in each category is the same which means the data held by all participants are independent identically distribution (IID). To verify the applicability of ADFL, we further experiment on the MNIST and EMNIST datasets under the Non-IID setting. Next, we

Algorithm 2: ADFL Algorithm.

Input: Number of clients n, communication round t, random noise z, clean data D_{clean}, local datasets D_i, local epoch E_{local}, local learning rate η and start epoch of GAN T_{GAN}.
Output: Global model $\tilde{G}_t$ for round t.
Assign model $\tilde{G}_t$ to clients
for *each communication round* $t \in [1, 2, \cdots, T]$ **do**
  // **Client Executes**
  for *each client* $i \in S_t$ **do**
    EVENT: Received global model $\tilde{G}_t$
    Replace the local model: $G_t^i \leftarrow \tilde{G}_t$
    $\mathcal{B}_{local} \leftarrow$(split D_i into batches of size B_{local})
    for *local epoch* $k \in [1, E_{local}]$ **do**
      for *local batch* $b \in \mathcal{B}_{local}$ **do**
        LocalTraining: $G_{t+1}^i = G_t^i - \eta \nabla \ell(G_t^i, b)$;
      end
    end
    Upload G_{t+1}^i to the server;
  end
  // **Server Executes**
  Compute average gradient $\tilde{G}_{t+1}$ based on Eq. 1
  if $t \geq T_{GAN}$ **then**
    Execute algorithm 1 to obtain $D_{distillation}$
    $\mathcal{B}_{kd} \leftarrow$(split $D_{distillation}$ into batches of size B_{kd})
    for *distillation epoch* $k \in [1, E_{kd}]$ **do**
      for *distillation batch* $b \in \mathcal{B}_{kd}$ **do**
        Update joint model $\tilde{G}_{t+1}$ based on Eq. 7;
      end
    end
  end
  Send $\tilde{G}_{t+1}$ to the clients;
  $S_{t+1} \leftarrow$(random set of n clients);
end
Return $\tilde{G}_{t+1}$.

will give a detailed introduction from three aspects: model setting, training configuration, and evaluation metrics.

Model Settings. In this experiment, convolutional neural networks (CNN) are used to construct classifiers in federated learning and the discriminator in GAN, while the generators in GAN are mainly constructed by convolution transposition. Table 1 shows the detailed structure of neural networks used in MNIST and CIFAR-10 datasets. Among them, the classifier in federated learning and discriminators in GAN adopt the same network structure, where only the dimension of the last output is different (the output of the classifier in federated learning is 10 dimensions, i.e. 10 classification tasks are performed, and the output of the discriminator is 1 dimension, i.e. whether the generated data is true or false). In addition, to facilitate the construction of the generator, network structures on the MNIST task and CIFAR-10 task are the same in this experiment. Note that the above netw ork structures are employed on the MNIST task and CIFAR-10 task since MNIST and CIFAR-10 are relatively simple. For the CeleBA dataset, we use ResNet18 architecture as the classification network also being the discriminator, and employ the regularized Resnet generator architecture suggested by Roth et al. (2017) which is proven to be more stable. It is worth noting that the structure of the generator can be designed according to the difficulty of the task. Moreover, to comply with the setting of attack and defense methods and ensure fairness, we used Resnet18 for classification tasks on all the datasets in comparison evaluations.

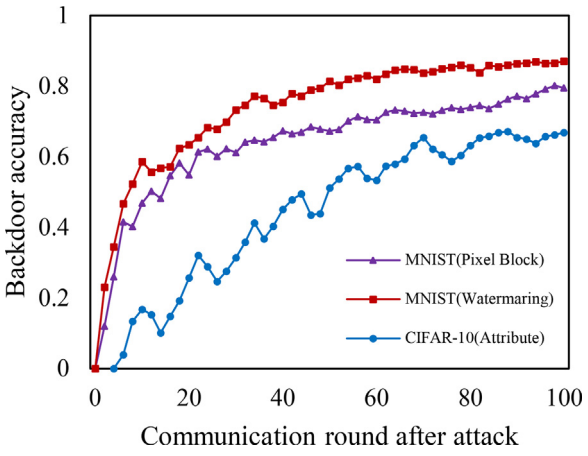


Fig. 5. Effectiveness of backdoor attack in federated learning.

Training Configurations. Regarding the federated learning algorithm in the experiment, we adopted the standard federated learning framework McMahan et al. (2017) proposed by Google in 2017. For each round, we randomly select 10 participants to perform the training algorithm where 3 malicious participants are selected. That means the malicious participants will participate in federated learning in each round, which makes the resistance of federated learning to backdoor attacks due to the randomness of selection disappear. For pixel block backdoor attacks and watermarking backdoor attacks on MNIST datasets, each participant has 6000 (60000 / 10) samples, and the attacker selects 200 samples (2000 in total) from 10 categories to embed pixel block (watermarking) and assigns them with the target labels specified by attackers. During the local training, the participants train for 20 epochs with a learning rate of 0.05. The communication rounds of federated learning are 300. First, we deploy the generation network at the initial stage of federated learning. After 100 rounds ($T_{GAN} = 100$), we fully deploy the ADFL scheme, in which the generator generates 10,000 pieces of samples without labels to be distilled and extracts 1%, 3%, 5%, and 10% data from testing datasets as clean samples to train the clean model. Moreover, the training rounds of the clean model and knowledge distillation is 20 rounds.

Evaluation Metrics. We evaluate the performance of ADFL with three metrics: 1) Attack success rate (ASR), which indicates the ratio of backdoor samples classified as the labels desired by the attackers; 2) Main task accuracy (ACC), which represents the performance of the model on clean samples; 3) Relabeling accuracy represents the ratio of the number of generated samples correctly classified by the clean model to the total amount of generated samples.

6.2. Effectiveness of backdoor attack in federated learning

We first verified the effectiveness of Federated learning backdoor attacks on MNIST and CIFAR-10 datasets. Among them, the MNIST dataset is used in pixel block backdoor attacks while the CIFAR-10 dataset is employed in attribute backdoor attacks. The experimental results are shown in Fig. 5. From the trend of the lines in the figure, we can see that the accuracy of backdoor attacks increases rapidly once the attackers start uploading the backdoored local model. In more detail, the pixel block backdoor and watermarking backdoor on MNIST datasets can reach around 80% and 85% respectively, while the accuracy rate of attribute backdoor attack on CIFAR-10 is about 65%. According to our analysis, it is mainly because a growing number of updates updated by benign participants are aggregated with the global model with the continuous iteration of federated learning communication rounds, resist-

Table 1
Structure of Neural Networks .

Discriminator	$28^1 \times 1 \xrightarrow{Conv,\ LReLU,\ DP} 14^2 \times 64 \xrightarrow{Conv,\ BN,\ LReLU} 7^2 \times 128 \xrightarrow{Conv,\ BN,\ LReLU} 4^2 \times 256 \xrightarrow{Global\_Average\_Pooling} 256 \xrightarrow{Dense} 10 \times 1$
Generator	$100 \times 1 \xrightarrow{Dense,\ BN,\ LReLU,\ Reshape} 7^2 \times 256 \xrightarrow{ConvT,\ BN,\ LReLU} 7^2 \times 128 \xrightarrow{ConvT,\ BN,\ LReLU} 14^2 \times 64 \xrightarrow{ConvT,\ Tanh} 28^2 \times 1$

LReLU → LeakyReLU, *DP* → Dropout, *BN* → BatchNorm

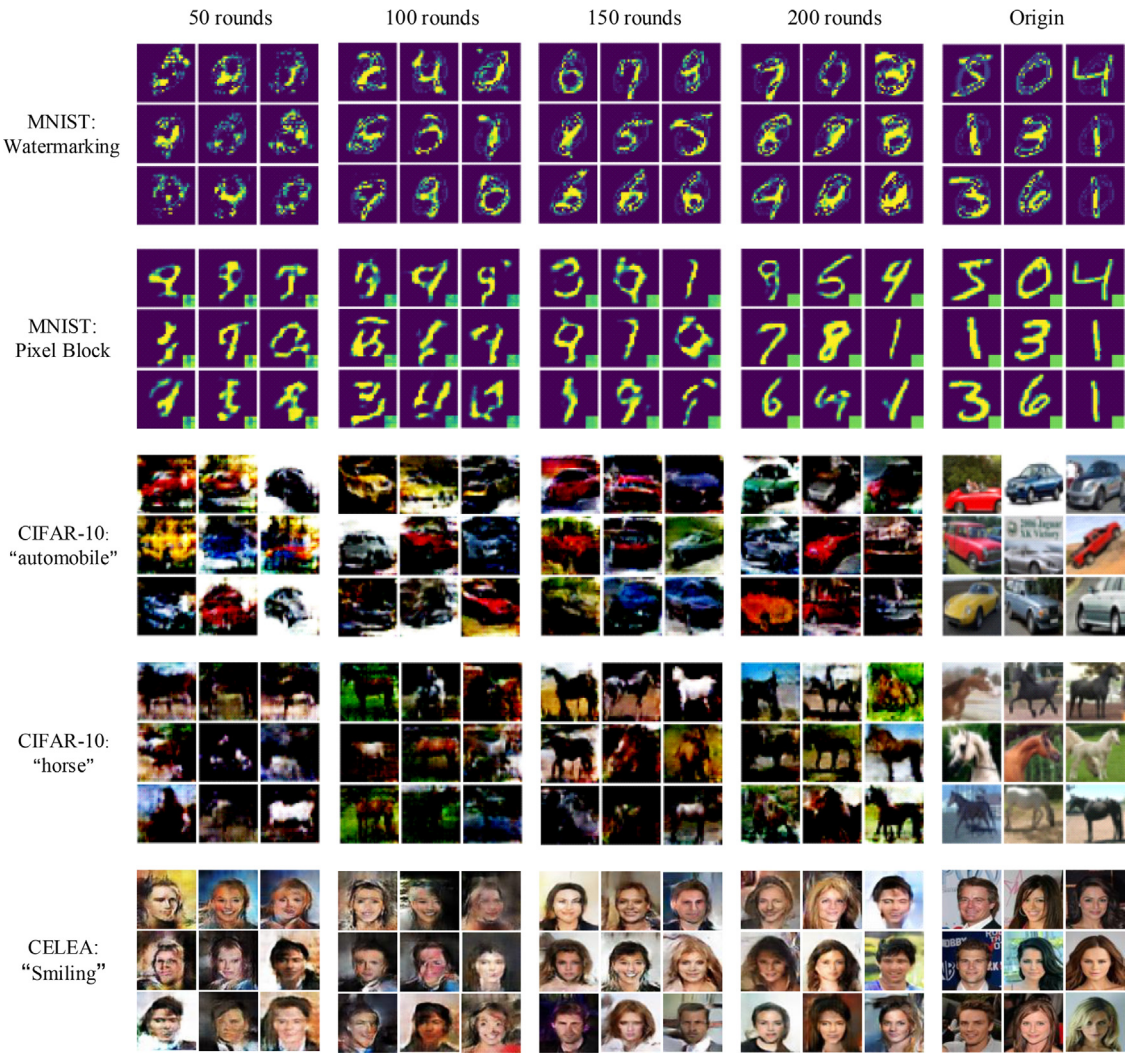


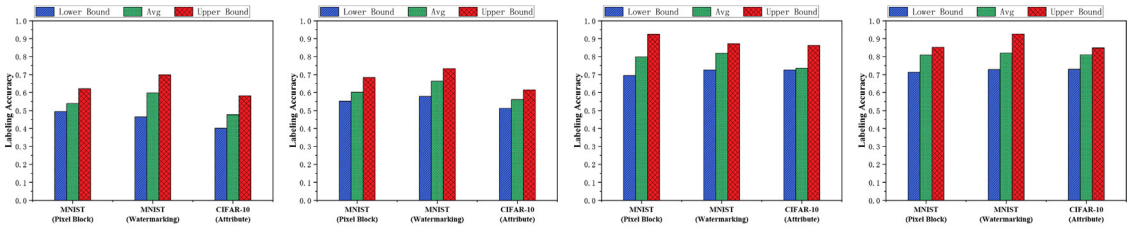
Fig. 6. Reconstruction results on MNIST and CIFAR-10 datasets based on GAN.

ing the poisoning updates uploaded by attackers. To prevent the above phenomenon, we design a simple adaptive mechanism that when the accuracy rate of backdoor attacks is less than 60%, the local learning rate of the attacker is canceled, thus ensuring the dominant position of backdoor update in the global model.

6.3. Validity of distillation data generation algorithm

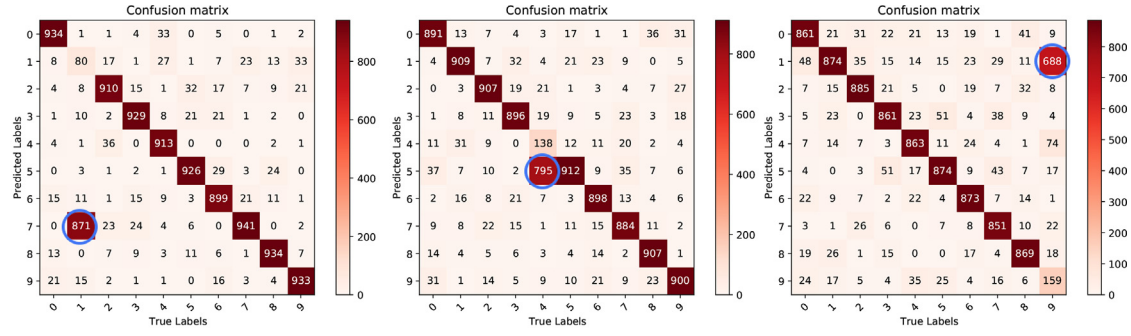
The purpose of distillation dataset generation based on GAN is to gain a number of trigger samples which are later assigned with correct labels to revise the global model. Therefore, the quality of the generated samples is the key to affecting the defense effect. There are two key factors that affect their quality of them: the generation effect of GAN and the ability of clean models to relabel.

Specifically, due to the existence of malicious participants, there are backdoor parameters in the global model, leading that the generated samples generated by the generator also have backdoor attributes. Fig. 6 presents the fake samples with a backdoor trigger generated with the iteration of the federated learning communication rounds under different attacks and different datasets. The first row shows the generation effect for the watermarking backdoor where the attacker embeds a watermarking "0" with a watermarking factor of 0.3 in the original dataset. It can be observed that the category of handwritten digits and the embedded watermarking can be basically recognized after about 100 iterations. Furthermore, the performance of the generator is constantly optimized and the generated image is clearer with the continuous iteration of federated learning. The second row demonstrates generated samples under the pixel block backdoor attack, embedding 5 × 5 pixel blocks in normal samples. Compared with watermarking triggers, pixel blocks have a very clear square outline in a very early round. However, there is still some noise point inside the square that will disappear after certain rounds. The third and fourth rows show the sample generation effect of backdoor attacks on the CIFAR-10 dataset. Since the CIFAR-10 dataset is 32 × 32 three-channel and the complexity of the neural network used in

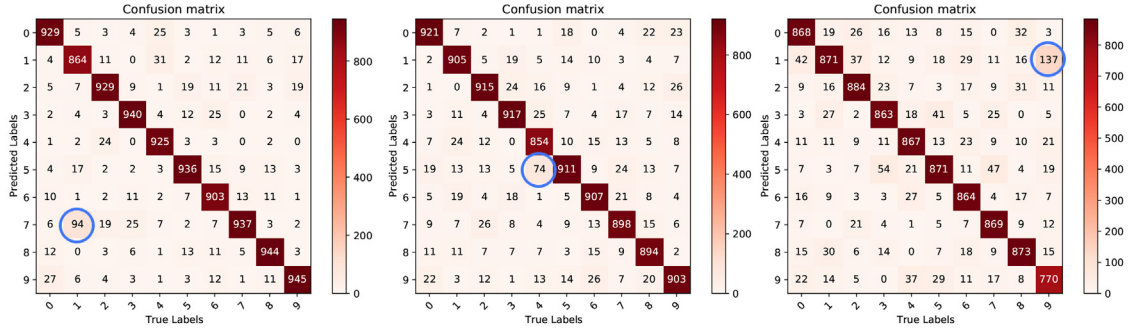


(a) Clean Data Rates: 1% (b) Clean Data Rates: 3% (c) Clean Data Rates: 5% (d) Clean Data Rates: 10%

Fig. 7. Relabeling accuracy of the clean model with different clean data holding rates on two datasets.



(a) MNIST with backdoor attack (Watermarking) (b) MNIST with backdoor attack (Pixel Block) (c) CIFAR-10 with backdoor attack (Attribute)



(d) MNIST with ADFL (Watermarking) (e) MNIST with ADFL (Pixel Block) (f) CIFAR-10 with ADFL (Attribute)

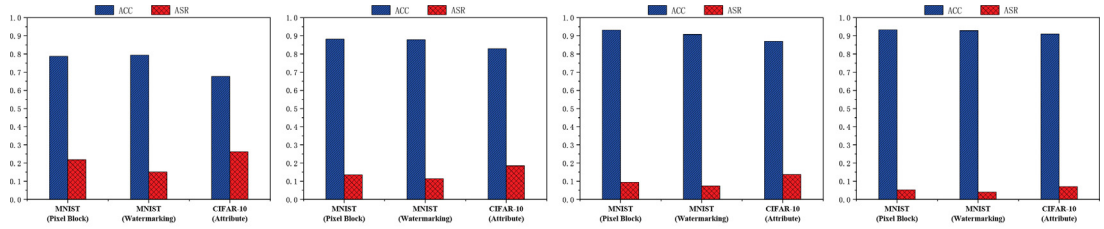
Fig. 8. Confusion matrix of the proposed ADFL and baseline backdoor model on MNIST and CIFAR-10 datasets.

this experiment is limited (the same network structure as that on the MNIST dataset), the convergence speed of it is much slower than that of the MNIST dataset that clear fake samples is not generated until 150–200 rounds.

From the above analysis, it can be seen that the generated datasets still contain many samples with backdoor triggers, which are the root cause of the misclassification of the model. Therefore, we verified the results of relabeling under different proportions (1%, 3%, 5% and 10%) of clean data. Fig. 7 shows the accuracy of the clean model in relabeling the generated samples of the MNIST dataset (pixel block and watermarking) and CIFAR-10 dataset (attribute backdoor). Note that the same relabeling task is repeated 10 times. Intuitively, Fig. 7 presents the lower limit (lowest value), the upper limit (highest value), and the average value of the relabeling accuracy. It can be seen that about 50% of the generated samples containing backdoor triggers can be labeled correctly using only 1% of the clean data. Within expectation, the clean model performs better with the higher holding of the clean data. Especially, the accuracy has reached about 85% when the proportion of the clean data reaches 5%, at which the clean model for the generated samples has converged. Even if the proportion of the clean data enlarges double times (10%), it can only achieve a negligible improvement. It is worth noting that the CIFAR-10 dataset is relatively complex so the model on it converges slowly. But in general, with the increase in the proportion of clean data, its classification accuracy is still steadily improving.

6.4. ADFL Defense performance analysis

Finally, we verified the defense performance of ADFL against backdoor attacks in federated learning. In this experiment, we obtained 10,000 labeled distillation samples through the distillation datasets generation algorithm based on GAN. According to the ADFL scheme, the backdoored model is purified by using these distillation data to perform the knowledge distillation process. To better evaluate the robustness of ADFL, we visualize the confusion matrix without and with ADFL on MNIST and CIFAR-10 under Watermarking and Pixel block backdoor attacks respectively (Fig. 8). In the MNIST task, the test samples in which the original labels are “1” were embedded in a special watermarking and assigned with a target label “7” under watermarking attack, while that of it are “4” were embedded in a 4×4 pixel block and assigned with the label “5” under pixel block attack. For CIFAR-10 task, we assign the 9-th class (truck) with the label of the 1-th class (automobile) under the same setting as MNIST under Pixel Block attack. Intuitively, results indicate that most of the samples “1” with triggers are misclassi-



(a) Clean Data Rates: 1% (b) Clean Data Rates: 3% (c) Clean Data Rates: 5% (d) Clean Data Rates: 10%

Fig. 9. Performance evaluation of the proposed ADFL with different clean data rates on three attacks.

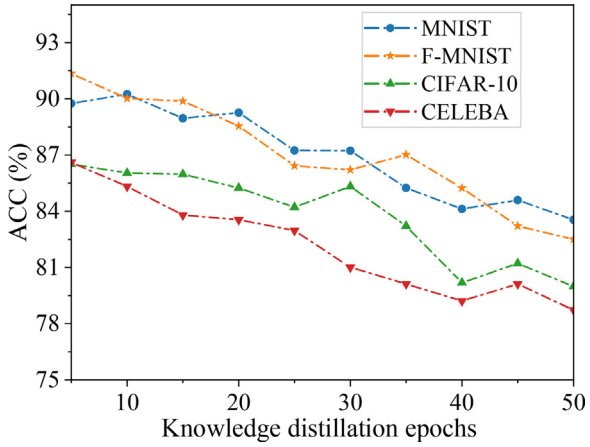


Fig. 10. ACC on four datasets with different Knowledge distillation epochs.

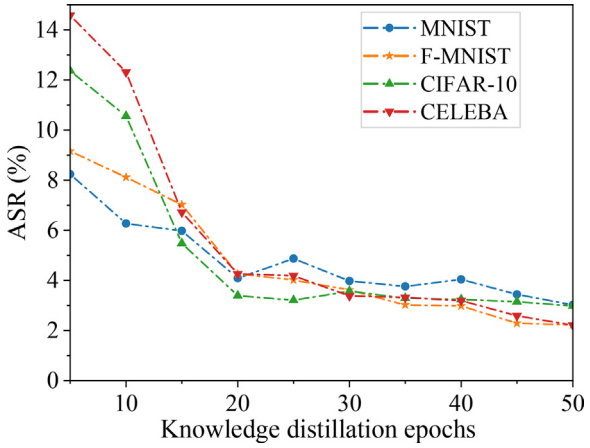


Fig. 11. ASR on four datasets with different Knowledge distillation epochs.

fied into "7". The same situation occurs in CIFAR-10. Fortunately, the vast majority of crafted inputs are classified into their original labels with the help of ADFL.

Subsequently, we conducted a more comprehensive experiment. Fig. 9 summarized the results of ADFL scheme in the federated learning prototype system. We select different proportions of clean data (1%, 3%, and 5%), and execute 200 rounds of federated learning communication rounds to compare the average accuracy of the backdoor task and the main task. It can be seen from Fig. 8 that with 1% clean data, the accuracy of the main task of the global model has reached 70% - 80%, while that of the backdoor task is effectively controlled between 20% - 30%. Especially, the accuracy of the watermarking backdoor attack is reduced to 15%. Since the model used in the experiment is relatively simple, the defense effect on CIFAR-10 dataset is slightly lower than that on MNIST dataset, but it is still dropped at about 25%. Overall, both the accuracy of remarking and the performance of ADFL are constantly enhanced with a higher proportion of clean data. Furthermore, even with 5%, ADFL still achieves an effective defense (ACC reaching more than 90% while ASR maintained about 5%).

Moreover, we explore the influence of knowledge distillation epochs on ADFL. Note that knowledge distillation is executed in the last round of federated learning when the generator can generate high-quality samples with backdoor triggers. The knowledge distillation epochs refer to the number of epochs that we perform knowledge distillation in the last federated communication round. We vary the knowledge distillation epochs from 5 to 50 where the span is 5. Fig. 10 and Fig. 11 show the trend of ACC and ASR with knowledge distillation rounds, respectively. Expectedly, ACC and ASR both decreased with the increase of epochs, meaning that more distillation epochs contribute to the defense effect but may cause degradation to main task accuracy. Besides, more distillation epochs are meant for more calculation overhead. Therefore, it is significant to set reasonable epochs to balance the ACC and ASR. In our experiments, we set the distillation epoch to be 20 since the ASR maintains an ideal value and the drop in ACC is acceptable.

6.5. Comparison evaluations

6.5.1. Different types of triggers

We verified the performance of ADFL compared with three baseline backdoor defense methods Andreina et al. (2021); Awan et al. (2021); Fung et al. (2020), where ADFL is assumed to be access to 5% of unlabeled clean data. For backdoor attacks, we use three types of visible triggers and different trigger sizes, which represent the watermark coefficient, size of pixel block, and parameter of noise respectively.

The experimental results (Table 2) show that ADFL can lower the ASR of all three backdoor attacks to 3% on the two benchmark datasets, which surpasses the performance of the other three defense methods. Overall, all defense methods in the experiments are effective against the three backdoor attacks. Specifically, FoolsGold and Baffle have a similar level of defense against backdoor attacks. Compared with them, the effect of CONTRA, lowering the ASR to about 2%, is slightly better. However, all of the baseline defense methods cause degeneration in ACC while ADFL maintains the ACC of the global model above that of æbeforeg, which is a distinct superiority of ADFL. Two reasons can account for this. On the one hand, ADFL overcomes the limitation through learning the knowledge from the teacher network with the distillation data as inputs. The GAN enabled data augmentation method fully exploits the potential data distribution, and these generated data are given high-quality labels, which contribute to improving the model accuracy. On the other hand, the same structure of clean mode and joint model (S-model) make the knowledge better absorbed by the student (T-model).

Furthermore, we evaluated the performance of ADFL in two new types of backdoor attacks, namely distributed backdoor at-

Table 2
Comparison results of different federated backdoor defense methods on two datasets.

Backdoor attacks	Trigger size	Before		**FoolsGold** Fung et al. (2020)		**CONTRA** Awan et al. (2021)		**Baffle** Andreina et al. (2021)		***ADFL***	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
Watermarking (MNIST)	**0.1**	59.86	84.75	3.92	78.63	1.42	82.05	2.65	80.34	**0.83**	**85.62**
Zheng et al. (2022)	**0.3**	60.32	83.57	3.68	79.83	1.96	82.15	3.72	79.01	**0.53**	**86.85**
	0.5	58.49	84.85	3.76	79.05	1.83	80.45	4.51	77.41	**1.11**	**85.83**
Pixel block (MNIST)	**3x3**	56.73	86.49	3.24	80.92	1.08	78.59	4.84	79.56	**1.04**	**87.28**
Tran et al. (2018)	**5x5**	57.76	85.28	2.85	79.68	1.85	82.42	3.38	79.27	**0.41**	**85.92**
	7x7	56.47	86.52	2.33	79.22	1.51	81.17	2.88	78.22	**0.78**	**85.77**
Attribute (CelebA)	**20**	59.27	84.05	4.26	80.41	2.13	81.04	3.83	79.74	**1.24**	**85.02**
Bagdasaryan et al. (2020)	**30**	63.04	83.92	3.05	79.98	2.01	79.65	4.02	79.96	**0.97**	**84.14**
	40	64.71	83.14	3.78	79.72	2.52	80.71	3.19	80.22	**1.14**	**84.79**
Attribute (CIFAR-10)	**20**	56.89	83.57	2.98	79.65	1.03	79.83	3.71	78.75	**0.89**	**86.85**
Bagdasaryan et al. (2020)	**30**	62.54	83.73	3.79	79.75	1.37	78.09	3.43	80.74	**0.46**	**86.64**
	40	63.12	83.35	4.83	78.51	1.81	80.05	3.67	78.18	**0.83**	**85.76**
Average		59.13	84.68	3.49	79.41	1.54	80.53	3.64	79.05	**0.76**	**86.28**

Table 3
Comparison results of different federated backdoor defense methods against backdoor attacks based on distributed triggers and imperceptible triggers.

Backdoor attacks	Attack	Before		FoolsGold		CONTRA		Baffle		ADFL	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
	DBA	100	90.39	97.82	79.65	7.36	87.97	8.06	81.97	**3.25**	**88.56**
MNIST	**WANET**	93.76	90.59	18.63	81.32	13.7	84.35	15.62	80.06	**4.09**	**89.25**
	DBA	100	89.57	98.17	79.45	6.9	86.49	9.11	79.39	**2.97**	**88.72**
F-MNIST	**WANET**	94.19	90.23	17.71	80.94	13.21	83.17	14.06	81.83	**4.26**	**88.54**
	DBA	93.96	72.54	92.14	71.62	8.42	69.75	8.68	68.41	**2.31**	**71.69**
CIFAR-10	**WANET**	92.85	86.85	18.67	79.37	14.18	79.61	16.7	80.56	**3.39**	**85.24**
	DBA	92.37	71.42	91.19	69.45	8.91	68.94	9.11	69.39	**3.97**	**71.06**
CelebA	**WANET**	92.19	85.23	17.71	78.94	13.21	80.17	14.06	79.83	**4.26**	**83.54**

Table 4
Comparison results of different federated backdoor defense methods on MNIST and EMINIST (Non-IID).

Datasets	Setting	Before		FoolsGold		CONTRA		Baffle		ADFL	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
	α **= 0.05**	84.79	83.53	82.12	76.79	53.36	79.12	67.73	73.64	**6.59**	**80.97**
MNIST	α **= 0.1**	88.76	87.76	58.53	77.13	33.7	82.96	28.96	81.16	**4.17**	**86.72**
	α **= 10**	93.76	90.59	18.63	81.32	13.7	84.35	15.62	80.06	**4.09**	**89.25**
	α **= 0.05**	80.1	68.5	77.14	65.74	43.42	62.17	54.91	60.87	**7.32**	**66.37**
E-MNIST	α **= 0.1**	84.85	71.85	58.67	66.89	34.18	65.43	52.81	63.91	**5.86**	**69.68**
	α **= 10**	90.54	76.54	18.17	70.09	19.2	70.94	10.49	71.42	**2.99**	**74.96**

tacks Xie et al. (2020) (DBA) and invisible backdoor attacks Nguyen and Tran (2021) (WANET). Compared with the above three backdoor attacks, DBA and WANET achieve higher ASR and ACC, meaning that they are more disruptive in federated learning. Overall, all defenses degrade in performance against the two new attacks. Surprisingly, FoolsGold is almost completely ineffective against DBA. However, our method still outperforms the three baseline defenses, lowing the two backdoor attacks to 4.5% on all the datasets. It is worth mentioning that although the backdoor WANET uses is invisible, the GAN we deploy can still be effective since the GAN generated is a potential data distribution that has nothing to do with whether the trigger is visible visually.

6.5.2. Non-IID setting

Non-IID data distributions are significant and realistic settings in the FL scenarios where data is distributed in a Non-IID fashion among clients. Following prior art Zhu et al. (2021), we model non-iid data distributions using a Dirichlet distribution Dir(α), where a smaller α indicates higher data heterogeneity. Specifically, we vary α from 0.05, 0.1, and 10 on MNIST and EMNIST against WANET, where $\alpha = 10$ represents IID. Table 2 shows the performance of ADFL compared with three baseline backdoor defense methods. The results indicate that ADFL significantly outperforms all baselines on Non-IID settings and lowers the ASR of WANET by around 7% while causing negligible degeneration in ACC. Intuitively, both the ASR and the ACC are lower as the degree of data heterogeneity increases, where the ASR still is above 80% which is effective. However, the three baseline defenses are no longer effective. According to our analysis, it is because both of FoolsGold and CONTRA rely on detecting abnormal updates to defend against backdoor attacks while it is hard to distinguish abnormal or normal updates under Non-IID setting. For Battle, it dynamically adjusts the weight of each client based on the update, so the performance of Battle also degrades. For ADFL, it will not be affected by update differences caused by Non-IID. Thus, ADFL is more robust than the previous methods in the scene of Non-IID data distributions.

7. Conclusion

This paper proposed a novel backdoor defense framework (ADFL) that combined both adversarial generative networks and knowledge distillation. ADFL mainly implemented backdoor defense in two steps, namely distillation dataset generation, and knowledge distillation. In the first step, ADFL deploys adversarial generative networks on the server side to generate unlabeled samples. Then, a clean model trained on a small amount of clean data labels these generated samples to obtain distillation datasets. Subsequently, knowledge distillation that employs the clean model as

the teacher model and the global model as the student model was implemented by taking the distillation datasets as inputs, so that purify the backdoored global model while maintaining its performance. We verified the effectiveness of ADFL through experiments and a comparison with state-of-the-art approaches. In future work, we will further study how to reduce the demand for clean data, even in the absence of clean data, realizing more effective defense against backdoor attacks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Chengcheng Zhu: Conceptualization, Methodology, Software. **Jiale Zhang:** Investigation, Writing – original draft. **Xiaobing Sun:** Data curation, Software. **Bing Chen:** Validation, Supervision. **Weizhi Meng:** Writing – review & editing.

Data availability

The data that has been used is confidential.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62206238), the Natural Science Foundation of Jiangsu Province (Grant No. BK20220562), the Yangzhou city-Yangzhou University Science and Technology Cooperation Fund Project (YZ2021158), and the Natural Science Research Project of Universities in Jiangsu Province (No. 22KJB520010).

References

Aledhari, M., Razzak, R., Parizi, R.M., Saeed, F., 2020. Federated learning: a survey on enabling technologies, protocols, and applications. IEEE Access 8, 140699–140725.

Andreina, S., Marson, G.A., Möllering, H., Karame, G., 2021. Baffle: backdoor detection via feedback-based federated learning. In: IEEE International Conference on Distributed Computing Systems. IEEE, pp. 852–863.

Awan, S., Luo, B., Li, F., 2021. Contra: defending against poisoning attacks in federated learning. In: European Symposium on Research in Computer Security. Springer, pp. 455–475.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V., 2020. How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2938–2948.

Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S., 2019. Analyzing federated learning through an adversarial lens. In: International Conference on Machine Learning. PMLR, pp. 634–643.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T.V., Petrou, D., Ramage, D., Roselander, J., 2017a. Towards federated learning at scale: system design. In: Proceedings of Machine Learning and Systems, pp. 374–388.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K., 2017b. Practical secure aggregation for privacy-preserving machine learning. In: ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 1175–1191.

Cao, X., Fang, M., Liu, J., Gong, N.Z., 2021. FLTrust: byzantine-robust federated learning via trust bootstrapping. Network and Distributed System Security (NDSS) Symposium. ISOC.

Chen, H., Fu, C., Zhao, J., Koushanfar, F., 2019. Deepinspect: a black-box trojan detection and mitigation framework for deep neural networks. In: International Joint Conference on Artificial Intelligence. Morgan Kaufmann, pp. 4658–4664.

Fang, M., Cao, X., Jia, J., Gong, N., 2020. Local model poisoning attacks to byzantine-robust federated learning. In: 29th USENIX Security Symposium. USENIX Association, pp. 1605–1622.

Fang, X., Ye, M., 2022. Robust federated learning with noisy and heterogeneous clients. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, pp. 10072–10081.

Fung, C., Yoon, C.J.M., Beschastnikh, I., 2020. The limitations of federated learning in sybil settings. In: International Symposium on Research in Attacks, Intrusions and Defenses. Springer, pp. 301–316.

Goodfellow I.J., Shlens J., Szegedy C.. Explaining and harnessing adversarial examples. 2014. ArXiv preprint arXiv:1412.6572.

Gu T., Dolan-Gavitt B., Garg S.. Badnets: identifying vulnerabilities in the machine learning model supply chain. 2017. ArXiv preprint arXiv:1708.06733.

Hinton G., Vinyals O., Dean J.. Distilling the knowledge in a neural network. 2015. ArXiv preprint arXiv:1503.02531.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X., 2021. Neural attention distillation: erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations. PMLR.

Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., Miao, C., 2020. Federated learning in mobile edge networks: acomprehensive survey. IEEE Commun. Surv. Tutor. 22 (3), 2031–2063.

Lin, J., Xu, L., Liu, Y., Zhang, X., 2020. Composite backdoor attack for deep neural network by mixing existing benign features. In: ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 113–131.

Liu, K., 2018. Brendan Dolan-gavitt, and siddharth garg. fine-pruning: defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions and Defenses. Springer, pp. 273–294.

Liu, Y., Fan, T., Chen, T., Xu, Q., Yang, Q., 2021. FATE: an industrial grade platform for collaborative learning with data protection. J. Mach. Learn. Res. 22 (1), 10320–10325.

Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X., 2018. Trojaning attack on neural networks. Network and Distributed Systems Security (NDSS) Symposium. ISOC.

Luo, B., Liu, Y., Wei, L., Xu, Q., 2018. Towards imperceptible and robust adversarial example attacks against neural networks. In: AAAI Conference on Artificial Intelligence. AAAI, pp. 1652–1659.

Lyu L., Yu H., Yang Q.. Threats to federated learning: A survey. 2020. ArXiv preprint arXiv:2003.02133.

McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR, pp. 1273–1282.

Nguyen, A., Tran, A., 2021. Wanet–imperceptible warping-based backdoor attack. In: International Conference on Learning Representations. PMLR.

Ozdayi, M.S., Kantarcioglu, M., Gel, Y.R., 2021. Defending against backdoors in federated learning with robust learning rate. In: AAAI Conference on Artificial Intelligence. AAAI, pp. 9268–9276.

Park, J., Han, D.J., Choi, M., Moon, J., 2021. Sageflow: robust federated learning against both stragglers and adversaries. In: Advances in Neural Information Processing Systems. MIT Press, pp. 840–851.

Roth, K., Lucchi, A., Nowozin, S., Hofmann, T., 2017. Stabilizing training of generative adversarial networks through regularization. In: Advances in Neural Information Processing Systems. MIT Press, pp. 2018–2028.

Song, M., Wang, Z., Zhang, Z., Song, Y., Wang, Q., Ren, J., Qi, H., 2020. Analyzing user-level privacy attack against federated learning. IEEE J. Sel. Areas Commun. 38 (10), 2430–2444.

Sun Z., Kairouz P., Suresh A.T., McMahan H.B.. Can you really backdoor federated learning?2019. ArXiv preprint arXiv:1911.07963.

Tran, B., Li, J., Madry, A., 2018. Spectral signatures in backdoor attacks. In: Advances in Neural Information Processing Systems. MIT Press, pp. 8011–8021.

Truong, L., Jones, C., Hutchinson, B., August, A., Praggastis, B., Jasper, R., Nichols, N., Tuor, A., 2020. Systematic evaluation of backdoor data poisoning attacks on image classifiers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE, pp. 788–789.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y., 2019. Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: IEEE Symposium on Security and Privacy. IEEE, pp. 707–723.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.y., Lee, K., Papailiopoulos, D., 2020. Attack of the tails: yes, you really can backdoor federated learning. In: Advances in Neural Information Processing Systems. MIT Press, pp. 16070–16084.

Xie, C., Chen, M., Chen, P.Y., Li, B., 2021. Crfl: certifiably robust federated learning against backdoor attacks. In: International Conference on Machine Learning. PMLR, pp. 11372–11382.

Xie, C., Huang, K., Chen, P.Y., Li, B., 2020. Dba: distributed backdoor attacks against federated learning. In: International Conference on Learning Representations. PMLR.

Yang, Q., Liu, Y., Chen, T., Tong, Y., 2019. Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. 10 (2), 1–19.

Yao, Y., Li, H., Zheng, H., Zhao, B.Y., 2019. Latent backdoor attacks on deep neural networks. In: ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 2041–2055.

Yin, X., Zhu, Y., Hu, J., 2021. A comprehensive survey of privacy-preserving federated learning: a taxonomy, review, and future directions. ACM Comput. Surv. 54 (6), 1–36.

Zhang, J., Chen, B., Cheng, X., Binh, H.T.T., Yu, S., 2020. Poisongan: generative poisoning attacks against federated learning in edge computing systems. IEEE Internet Things J. 8 (5), 3310–3322.

Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Ding, S., Shen, C., Wu, C., 2022. Dense: data-free one-shot federated learning. In: Advances in Neural Information Processing Systems. MIT Press, pp. 21414–21428.

Zhang, J., Chen, J., Wu, D., Chen, B., Yu, S., 2019. Poisoning attack in federated learning using generative adversarial nets. In: EEE International Conference On Trust, Security And Privacy In Computing And Communications/IEEE Interna-

tional Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, pp. 374–380.
Zhang, J., Ge, C., Hu, F., Chen, B., 2021. RobustFL: robust federated learning against poisoning attacks in industrial iot systems. IEEE Trans. Ind. Inf. 18 (9), 6388–6397.
Zhao, B., Sun, P., Wang, T., Jiang, K., 2022. Fedinv: byzantine-robust federated learning by inversing local model updates. In: AAAI Conference on Artificial Intelligence. AAAI, pp. 9171–9179.
Zheng, X., Dong, Q., Fu, A., 2022. WMDefense: using watermark to defense byzantine attacks in federated learning. In: IEEE Conference on Computer Communications Workshops. IEEE, pp. 1–6.
Zhou, C., Gao, Y., Fu, A., Chen, K., Dai, Z., Zhang, Z., Xue, M., Zhang, Y., 2023. Trojaning attack on neural networks. Network and Distributed Systems Security (NDSS) Symposium. ISOC.
Zhu, Z., Hong, J., Zhou, J., 2021. Data-free knowledge distillation for heterogeneous federated learning. In: International Conference on Machine Learning. PMLR, pp. 12878–12889.

Chengcheng Zhu received the B.S. degree from Yangzhou University, Yangzhou, China, in 2022. He is currently working toward the M.S. degree in computer science and technology with Yangzhou University, Yangzhou, China. His research interests include backdoor attacks and adversarial learning.

Jiale Zhang received the Ph.D. degree in computer science and technology the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2021. He is currently a Lecture with the School of Information Engineering, Yangzhou University, Yangzhou, China. His research interests are mainly federated learning, AI security, blockchian security, and privacy-preserving.

Xiaobing Sun received the B.S. degree in computer science and technology from Jiangsu University of Science and Technology, Zhenjiang, China, in 2007, and the Ph.D. degree from the School of Computer Science and Engineering, Southeast University, Nanjing, China, in 2012. He is currently a Professor with the School of Information Engineering, Yangzhou University, Yangzhou, China. His research interests include software maintenance and evolution, software repository mining, and intelligence analysis. He has been authorized more than 20 patents. He has published more than 80 papers in refereed international journals.

Bing Chen received the B.S. and M.S. degrees in computer engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1992 and 1995, and the Ph.D. degree in computer technology from the College of Compute Science and Technology, NUAA, in 2008. He is currently a Full Professor with the College of Compute Science and Technology NUAA. His research interests include cloud/edge computing, security and privacy, federated learning, and wireless communications.

Weizhi Meng eceived the Ph.D. degree in computer science from the City University of Hong Kong (CityU), Hong Kong. He worked as a Research Scientist with the Department of Infocomm Security (ICS), Institute for Infocomm Research, A*STAR, Singapore, and a Senior Research Associate with the Department of Computer Science, CityU. He is currently an Associate Professor with the Department of Applied Mathematics and Computer Science, Cyber Security Section, Technical University of Denmark (DTU), Denmark. His primary research interests are cyber security and intelligent technology in security, including intrusion detection, smartphone security, biometric authentication, HCI security, trust computing, blockchain in security, and malware analysis. He served as a program committee member for more than 50 international conferences. He won the Outstanding Academic Performance Award during his Ph.D. study, and was a recipient of Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in 2014 and 2017. He was also a recipient of the Best Paper Award from ISPEC 2018, and the Best Student Paper Award from NSS 2016 and Inscrypt 2019.