Hindawi
Security and Communication Networks
Volume 2024, Article ID 9856054, 1 page
https://doi.org/10.1155/2024/9856054

Retraction

Retracted: Defending Privacy Inference Attacks to Federated Learning for Intelligent IoT with Parameter Compression

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Zhu, H. Cao, Y. Ren et al., “Defending Privacy Inference Attacks to Federated Learning for Intelligent IoT with Parameter Compression,” *Security and Communication Networks*, vol. 2023, Article ID 9597905, 12 pages, 2023.

Hindawi
Security and Communication Networks
Volume 2023, Article ID 9597905, 12 pages
https://doi.org/10.1155/2023/9597905

Research Article

Defending Privacy Inference Attacks to Federated Learning for Intelligent IoT with Parameter Compression

Yongsheng Zhu,[1,2] Hongbo Cao,[3] Yuange Ren,[3] Wanqi Wang,[2] Bin Wang ,[4] Mingqing Hu,[5] Baigen Cai,[1] and Wei Wang [3]

[1]*School of Electronic and Information Engineering, Beijing Jiaotong University, No. 3 Shangyuancun, Beijing 100044, China*
[2]*Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*
[3]*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, No. 3 Shangyuancun, Beijing 100044, China*
[4]*Zhejiang Key Laboratory of Multi-Dimensional Perception Technology, Application and Cybersecurity, Hangzhou 310053, China*
[5]*iFLYTEK Co., Ltd., Hefei, China*

Correspondence should be addressed to Wei Wang; wangwei1@bjtu.edu.cn

Received 14 October 2022; Revised 25 November 2022; Accepted 18 April 2023; Published 11 May 2023

Academic Editor: Xingsi Xue

Federated learning has been popularly studied with people's increasing awareness of privacy protection. It solves the problem of privacy leakage by its ability that allows many clients to train a collaborative model without uploading local data collected by Internet of Things (IoT) devices. However, there are still threats of privacy leakage in federated learning. The privacy inference attacks can reconstruct the privacy data of other clients based on GAN from the parameters in the process of iterations for global models. In this work, we are motivated to prevent GAN-based privacy inference attacks in federated learning. Inspired by the idea of gradient compression, we propose a defense method called Federated Learning Parameter Compression (FLPC) which can reduce the sharing of information for privacy protection. It prevents attackers from recovering the private information of victims while maintaining the accuracy of the global model. Extensive experimental results demonstrated that our method is effective in the prevention of GAN-based privacy inferring attacks. In addition, based on the experimental results, we propose a norm-based metric to assess the performance of privacy-preserving.

1. Introduction

Deep learning (DL) [1], as one of the most popular machine learning methods driven by big data, has been widely studied and employed in various fields and different scenarios such as face detection [2], social networks [3, 4], natural language process [5, 6], speech technology [7–9], detection of network anomalies [10, 11], and multimodal learning [12–14].

However, the machine learning methods that require to train the aggregated original data collected from different entities have many problems. First, the original data contain sensitive privacy information. Aggravation of original data for training the model may cause severe privacy leakage. Second, it requires great computational resources during the process of training with big data. Third, it is also very costly in the aggregation of original data in the process of data transmission.

With the increasingly growing awareness of privacy protection, it is crucial to design a new effective machine learning paradigm in a way that protects sensitive data from privacy leakage [15–17].

Federated learning (FL) [18, 19] is a novel machine learning paradigm that performs learning in a distributed way. In federated learning, the data owner trains the model locally to avoid data leakage of clients.

Federated learning was first proposed by Google [20]. It is a server-client architecture that consists of a central parameter aggregation server and a number of distributed clients. The client trains the model locally with their own data and exchanges the parameters in iteration with central server. In this process, the data of the clients will not be submitted to the outside. As a result, the users' privacy and data security are guaranteed and the problem of data fragmentation and isolation are solved.

Although FL is very effective in privacy-preserving and breaks data silos, it is still surprisingly susceptible to GAN-based data reconstruction attacks [21], which is a kind of privacy inference attack [22] in the training phase of FL. Existing related work shows that differential privacy (DP) [23] is regarded as one of the strongest defense methods against these attacks. The core idea of DP is to introduce random noise into the privacy information, but DP often adds sufficient noise that the accuracy of the global model is reduced notably.

To address this problem, we focus on the privacy inference attacks toward nonindependent and identical distribution (non-i.i.d.) federated learning. In addition, we conduct various experiments to evaluate the privacy leakage that the adversary can get from the parameters of the global model during the training phase and understand the relationship between the reconstruction sample and global model information leakage. The experimental results show that parameter compression is an effective defense method against GAN-based reconstruction attacks toward federated learning.

In this paper, we extend our preliminary work that has appeared in [24]. First, we elaborate how our defense method is effective. Second, we present more detailed background in federated learning scenarios. Third, we discuss and analyze the advantages as well as the disadvantages of our methods.

We make the following contributions:

(i) We propose an efficient defense method of vertical federated learning based on parameter compression to avoid privacy leakage against GAN-based inferring attacks.

(ii) We compare our method with the current defense methods that add noise to the parameter. The experimental results demonstrate the effectiveness of our method.

(iii) We propose a norm-based metric for the assessment on the efficiency of various defense methods.

2. Background

It has been well recognized that FL is a peculiar form of collaborative machine learning technique. FL allows the clients to train their model without exchanging data to a centralized server, which combats the problems of privacy concerned about central machine learning and communication costs.

A traditional FL system is built by a central server to aggregate and exchange parameters and gradients. The end-user devices train their local model and exchange their parameter or gradient periodically without uploading data to ensure that there is no privacy leakage concern.

Generally, the whole training process of FL can be expressed as follows (see Figure 1):

(1) Client initialization: the clients download the parameter from the central server to initialize their local global

(2) Local training: every client uses the private data to train the model and upload parameters to the central server at last

(3) Parameter aggregation: the central server gathers the uploaded parameter from every client and generates a new global model by robust aggregation and SGD

(4) Broadcast model: the central parameter server broadcasts the global model to all the clients

Based on the characteristics of the data distribution [22], federated learning can be classified into three general types.

Horizon federated learning (HFL), which is also called homogeneous federated learning, usually occurs in the situation where the training data of the clients have overlapping identical feature space but have disparate sample space. Most research that focuses on FL assumes that the model is trained in HFL.

Vertical federated learning (VFL), which is also called heterogeneous federated learning, is suitable for the situation where the clients have the non-i.i.d. datasets [25]. Meanwhile, sample space is shared between clients who have different label spaces or feature spaces.

Federated transfer learning (FTL) [26] is suitable for situations similar to that of traditional transfer learning, which aims to leverage knowledge from previously available source tasks to solve new target tasks.

GAN-based privacy inferring attack is a kind of privacy attacks which occur in the training phase towards federated learning. The target of the adversary can be the sample reconstruction. This is an inferring attack that aims to reconstruct the training sample and/or associated labels used by other FL clients.

In the field of deep learning, generative adversarial networks (GANs) have recently been proposed, and they are still in a highly developed and researched stage [27, 28]. Various GANs have been proposed. They can be used to generate deepfake face [29] and generate image by text [30]. The goal of the GAN is not to classify images into different categories but to generate samples that are similar to the samples in the training dataset and have the same distribution without accessing the original samples.

The privacy leakage of the sample reconstruction attacks may come from model gradients [31] or model parameters [21]. Furthermore, the sample reconstruction attacks are considered not only on the client-side but on the server-side [32]. Besides, Fu et al. [33] proposed a label inference attack which is in a special and interesting non-i.i.d. federated learning setting. Existing related work regards differential privacy as an efficient method to defend the privacy inference attack [34–37]. In the local differential privacy, the

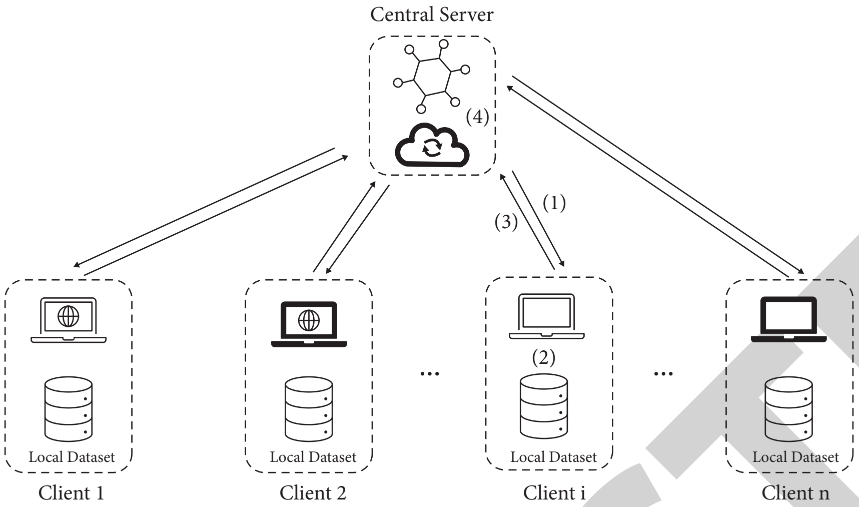

Figure 1: Overview of federated learning.

FL clients add Gaussian noise to the local gradients or parameters. Besides, there are also many other methods for defense inference attacks [38–40].

Another main attack toward federated learning is the robustness attack which aims to corrupt the model. Due to the characteristic of the inaccessibility of local training data in a typical FL system, poisoning attacks are easy to implement which causes FL to be even more vulnerable to poisoning attacks than classic centralized machine learning [41]. The goal of the adversary is to diminish the performance and the convergence of the global model. These misclassifications may cause serious security problems.

Otherwise, the backdoor attack [42] is known for its higher impact of its capabilities to set the trigger. It is an effective targeted method to attack FL system. Various methods are proposed to defend against poisoning attacks towards FL [43–46].

Besides, FL is vulnerable to adversarial attacks such as unauthorized data-stealing or debilitating global model.

There also exists related work on the prevention of android malware [47–51], on the adversarial attack toward machine learning [52–54], and on the detection of software vulnerabilities [55–58].

3. Methodology

We study the problem of the privacy leakage of GAN-based inferring attack toward federated learning. The goal of the attacker which is a malicious client in federated learning system is to reconstruct sample of another category in the classification task.

3.1. Problem Statement

3.1.1. Motivation. Hitaj et al. [21] first proposed a GAN-based reconstruction attack. The GAN-based inferring attack is definitely destroying the privacy-preserving system of federated learning. Although there are many defense methods against the inferring attack, some advanced attack still works. Besides, most recent defense method trades off the performance of FL with the security. In this context, the defense method with low impact to global model is indispensable.

3.1.2. Threat Model. In federated learning, all clients have their own data, and they train a global model with a common learning goal, which means that each client knows the data labels of the other clients. The central server is authoritative and trustworthy; it cannot be controlled by any attacker.

In this attack, malicious participants pretend to be the honest client in the FL system reconstruct the private data information of other honest participants. The attacker only needs to train a GAN locally to simulate the victim's training samples and then injects fake training samples into the system over and over again.

3.1.3. Analysis of Attack. In the horizon FL, every client holds overlapping identical feature space. In other words, every client can access the feature of every class, so the GAN-based inferring attack will not occurs. In this paper, we focus on the defense of GAN-based privacy inferring attacks toward federated learning which only occurs in non-i.i.d. settings.

Without anyone in the system noticing, the attacker can trick the victim into releasing more information about their training data and eventually recover the victim's sample data. Figure 2 shows the victim's data finally reconstructed by the attacker. It can be seen that the attacker recovers a very clear image.

Commonly, to train a GAN, some data of the target class and an appropriate network architecture of generator which learns to generate plausible data are needed.

The training of GAN can be expressed as a typical game confrontation process of finding the maximum and minimum values. The game between discriminator and generator is shown as follows:

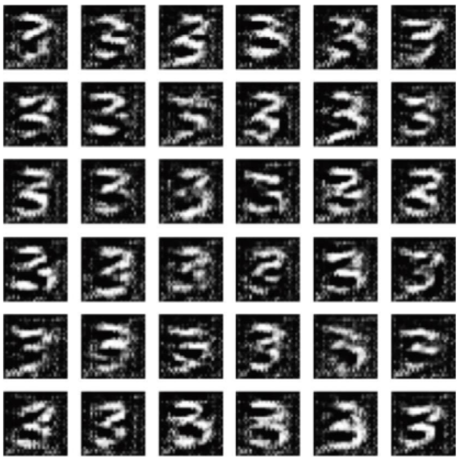

Figure 2: The image recovered by the attacker.

$$\min_{\theta_G}\max_{\theta_D}\sum_{i=1}^{n_+} F_1\left(\mathbf{x}_i;\theta_D\right)+\sum_{j=1}^{n_-}F_2\left(\mathbf{z}_j;\theta_G;\theta_D\right), \tag{1}$$

where

$$F_1\left(\mathbf{x}_i;\theta_D\right)=\log f\left(\mathbf{x}_i;\theta_D\right), \tag{2}$$

F_1 is the generator loss and

$$F_2\left(\mathbf{z}_j;\theta_G;\theta_D\right)=\log\left(1-f\left(g\left(\mathbf{z}_j;\theta_G\right);\theta_D\right)\right), \tag{3}$$

F_2 is the generator loss.

And, z is the random latent code, which will be input to the generator.

The attacker is an honest-but-curious and can access to the global model of every iteration in the federated learning system but tries to extract information about local data owned by other clients. The attacker builds a GAN model locally. At the same time, the attacker follows a protocol that is agreed upon by all clients. He uploads and downloads the correct number of gradients or parameters according to the agreement. The attacker influences the learning process without being noticed by other clients. He tricks the victim into revealing more information about his local data.

Adversary A participates in the collaborative deep learning protocol. All such clients agree in advance on a common learning objective, which means that they agree on the type of neural network architecture and labels on which the training would take place. Let V be another client (the victim) that declares labels [a, b]. The adversary A declares labels [b, c]. Thus, while b is in common, A has no information about class a. The goal of the adversary is to infer as much useful information as possible about class a.

The attack begins when the test accuracy of both the global model and the local model of the server is greater than a threshold. The attack process is as follows: first, V trains the local model and uploads the model parameters to the central server. Second, A downloads the parameters and updates his discriminator of GAN accordingly. A then generates samples of class a from GAN and marks it as class c. A trains his local model with these fake samples and uploads these parameters to the global model on the server side. Then, A tricks victim V to provide more information about class a. Finally, A can reconstruct images of class a that are very similar to V's own original images.

The key reason that the attack can get the changes of the global model in each round, which contains the feature of sensitive.

3.1.4. Compression Method. There is a gradient compression method in distributed learning, which reduces the communication overhead by compressing the gradient in each communication round [59–61]. Gradient sparsification is a kind of gradient compression. The sparsification algorithm decides to send a small part of the gradient to client in the parameter update, and most of the gradients with small changes are temporarily updated. The widely used gradient sparsification method is to select the gradient according to the compression rate $R\%$. In this method, the gradient with a maximum change of $1 - R\%$ was finally chosen. Usually, the compression ratios are 90%, 99%, and 99.9%.

Parameter compression (PC) method takes advantage of the idea of gradient compression. Since the parameters of the model contain the key information about the training data, compressing the parameters is equivalent to truncating some parameters, which reduces the data information leaked to the attacker and achieves the purpose of privacy protection. The framework of parameter compression towards FL is shown in Figure 3.

The algorithm of parameter compression of a single client model is presented in Algorithm 1. In the t^{th} round, for the j^{th} parameter component, it calculates the difference diff between round t and the previous round $t - 1$. Then, the k largest parameters are selected from the absolute value of diff. Finally, it can obtain the compression parameters of the j^{th} parameter component by adding these k parameters and the parameter of the round $t - 1$. When all the parameter components are compressed, the final compressed parameters of the model can be obtained. $R\%$ is defined as the compression ratio. If $R\%$ is 90%, it means that only the first 10% ($1 - R\%$) of the absolute value of the difference e will be updated.

The parameter compression scheme is applied to the GAN-based privacy inferring attacks. Before uploading the local model parameters, each client compresses the parameters and uploads them to the server. The server keeps its aggregation algorithm unchanged and still uses the federated average algorithm (FedAvg) to aggregate all parameters.

4. Results and Discussion

4.1. Dataset and Model Architecture. MNIST dataset: it consists of handwritten gray-scale images of digits ranging from 0 to 9. Each image is 28 × 28 pixels. The dataset consists of 60,000 training data samples and 10,000 testing data samples [62]. This experiment used a convolutional neural network (CNN) based architecture on the MNIST dataset. The layers of the networks are sequentially attached to one another based on the keras. Sequential () container so that

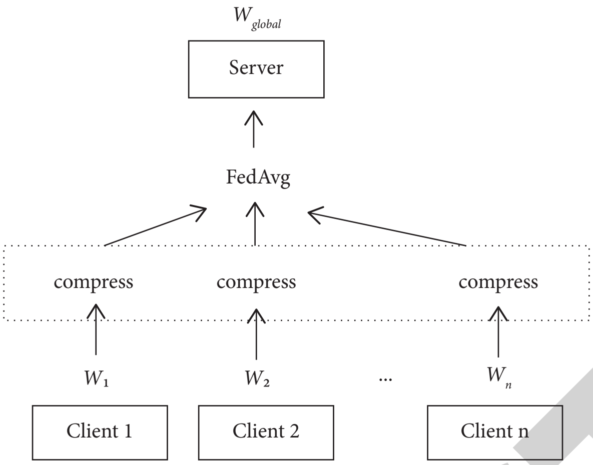

FIGURE 3: Compressing parameter to prevent GAN attack.

```
Require: parameters $w = \{w[0], w[1], \ldots, w[n]\}$
for $j = 0$ to $n$ do
    diff $\Longleftarrow w_t[j] - w_{t-1}[j]$
    count $\Longleftarrow |\text{diff}|$
    $k \Longleftarrow$ count $\cdot$ $(1 - R\%)$
    $w_{\text{compressed}}[j] \longleftarrow \text{top}_k(abs(\text{diff})) + w_{t-1}[j]$
end for
$C$ submit $w_{\text{compressed}}$ to server
```

ALGORITHM 1: Parameter compression on client C.

layers are in a feed-forward fully connected manner. The neural networks are trained by TensorFlow.

4.2. Results. The defense of GAN-based privacy inferring attacks takes the attack experiment of reconstructing the digital image of “3” as an example.

The results of the parameter compression scheme are as follows: when $R\% = 90\%$, the image finally recovered by the attacker is shown in Figure 4.

As can be seen, the image is much more blurred than the original image recovered by the attacker, but the number 3 in the image is still recognizable. Thus, this compression ratio is not high enough to prevent information leakage.

When $R\% = 99\%$, the attacker eventually recovers an image like Figure 5. The image is too fuzzy for the number to be recognized, but there are some outlines, which means some valid information is still leaked.

When $R\% = 99.9\%$, the image recovered by the attacker is shown in Figure 6. It can be seen that no valid data information can be seen at all. Therefore, when compression rate is 99.9%, the privacy leakage can be completely prevented.

4.3. Global Model Accuracy. In order to test whether the accuracy of the global model is influenced after the parameters of the client are compressed, the accuracy of the global model on the test dataset is calculated during each round of federated learning. Figure 7 shows the accuracy change of the global model on the test dataset: the final accuracy of the global model with different compression rates is above 94%. Compared with the baseline of the original attack without compression, it has no significant effect on the accuracy of the global model.

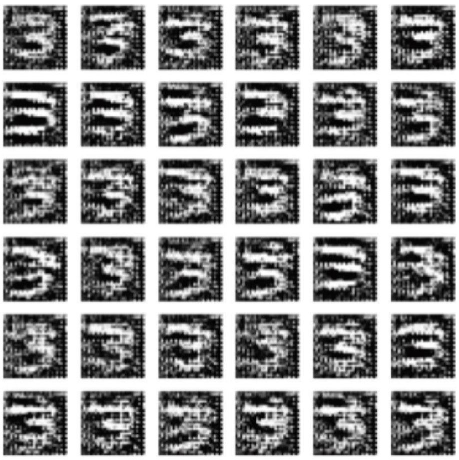

FIGURE 4: Defense result when $R\% = 90\%$.

4.4. Comparing with Gaussian Noise. Local differential privacy is often used to defend against this attack, but it may negatively impact the model performance if the strength of the noise is not appropriate (see Figure 8).

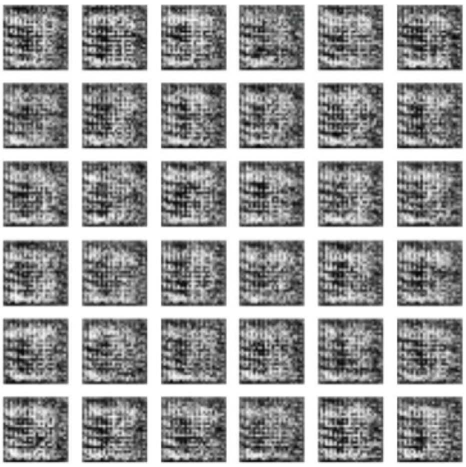

Figure 5: Defense result when $R\% = 99\%$.

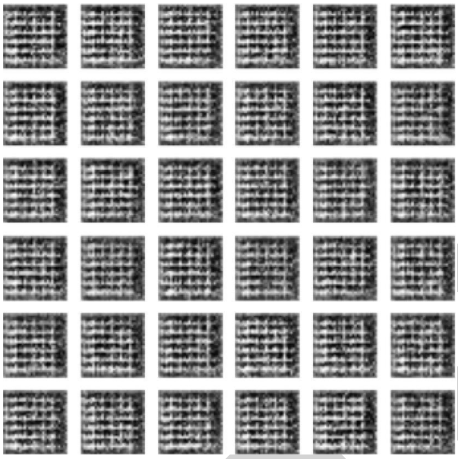

Figure 6: Defense result when $R\% = 99.9\%$.

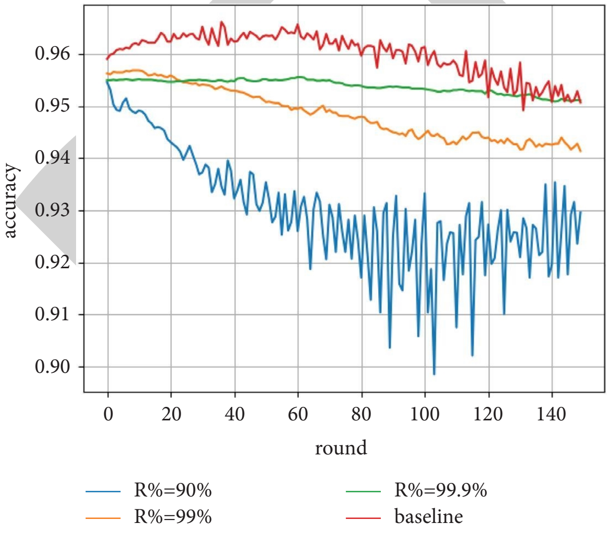

Figure 7: Test accuracy of the global model.

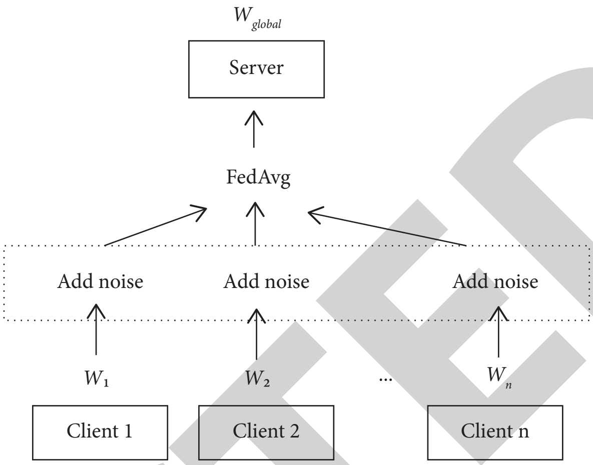

Figure 8: Adding noise to the parameter to prevent GAN attack.

Adding noise is a common way to disturb the information. When all clients upload updated parameters, they first add Gaussian noise to the updated parameters to protect their data information from leaking. In the experiments, the mean of Gaussian noise is set to 0, and the standard deviation of different noise is marked as $\text{noise}_{\text{scale}}$. And, $\text{noise}_{\text{scale}}$'s value is set as 10^{-4}, 10^{-3}, and 10^{-2}.

When $\text{noise}_{\text{scale}} = 10^{-4}$, the noise added is the smallest. It can be seen that it cannot prevent the leakage of data information, as shown in Figure 9. When $\text{noise}_{\text{scale}} = 10^{-3}$, the final image recovered by the attacker is shown in Figure 10. Although the image is more noisy than when $\text{noise}_{\text{scale}} = 10^{-4}$, there are very few outlines of the number three. When $\text{noise}_{\text{scale}} = 10^{-2}$, the image finally recovered by the attacker is shown in Figure 11. At this time, the content of the image is completely invisible. The attacker cannot obtain any valuable information about the digital image 3, which indicates that the attack failed.

From the previous experiments, it can be seen that only in the situation where the Gaussian noise standard deviation is greater than or equal to 10^{-2}, data leakage can be completely prevented. However, the accuracy of the global model is greatly affected. Figure 12 is the accuracy change curve of the global model on the test dataset. When $\text{noise}_{\text{scale}} = 10^{-3}$ and $\text{noise}_{\text{scale}} = 10^{-4}$, the final accuracy of the global model is similar to that of the baseline, both around 95%. But when $\text{noise}_{\text{scale}} = 10^{-2}$, as the blue curve shown in the figure, the final accuracy of the global model is 80.35%, which is a very large drop. It directly destroys the training and learning process of the global model.

4.5. Analysis of Privacy Protection. The experiment results are shown in Table 1. It can be seen that although noise, which is small, can be added to the parameters, it is not enough to cover up the information of the real samples. When the noise is large, it directly decreases the accuracy of

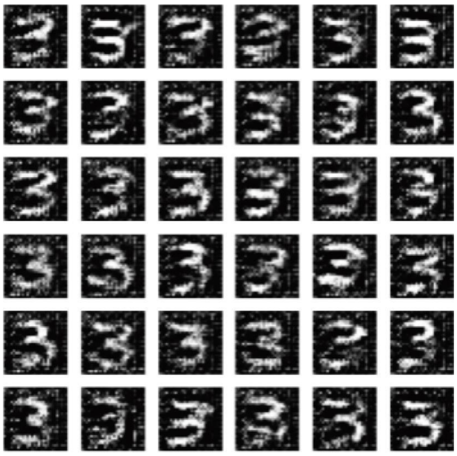

Figure 9: Defense result when $\text{noise}_{\text{scale}}$ is 10^{-4}.

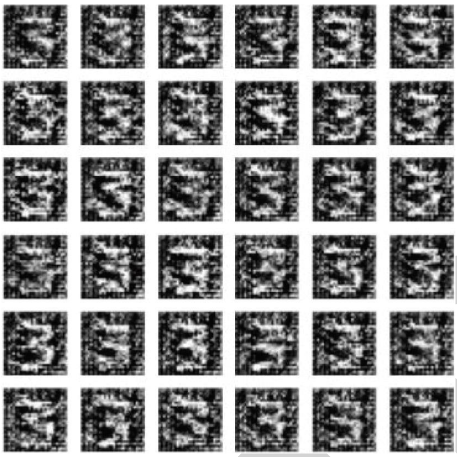

Figure 10: Defense result when $\text{noise}_{\text{scale}}$ is 10^{-3}.

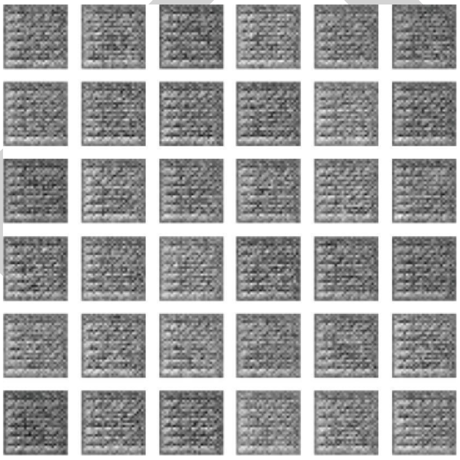

Figure 11: Defense result when $\text{noise}_{\text{scale}}$ is 10^{-2}.

the global model. Therefore, adding noise to the parameters is not a desirable defense method. In the parameter compression defense method, not only the private information is protected from leaking, but no great influence on the accuracy of the global model is exerted when the compression rate is 99.9%. Therefore, parameter compression is a desirable and efficient defense method.

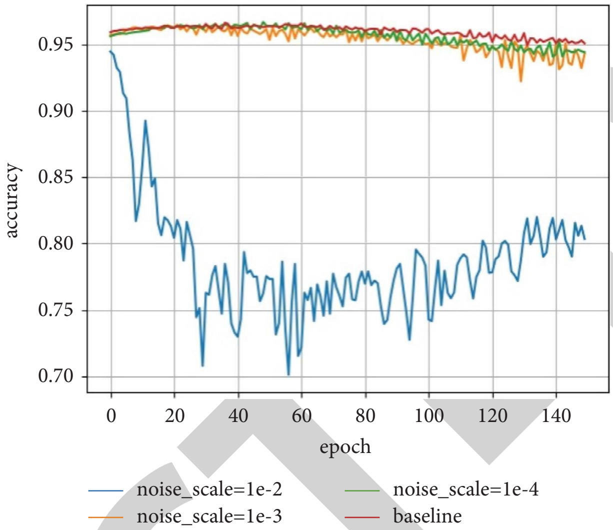

Figure 12: Test accuracy of the global model.

In GAN-based privacy inferring attacks, the premise on which the attacker's GAN network takes effect is that the model at the server and both local models have reached an accuracy that is higher than a certain threshold [21]. When the parameters are compressed, the accuracy of the model has reached a relatively high level and the accuracy of the model cannot be greatly affected. In the Gaussian noise defense method, adding larger noise is equivalent to directly making larger changes to the model parameters, which has a great impact on the accuracy of the global model. Therefore, parameter compression is an efficient defense method that prevents GAN-based privacy inferring attacks.

5. Security Assessment

5.1. Norm and Performance. Since both the parameter compression scheme and the adding noise scheme make changes to the parameters, in order to explain the degree of privacy protection of the two, the norm is introduced as a measurement standard.

The norm and model accuracy of the model in each round of training in the two schemes of parameter compression and parameter noise are calculated to verify the relationship between the global model norm change and the model performance. And, then the norms and model accuracy are plotted. The relationship can be found by analyzing the changes between the model norm and the model performance.

In the parameter compression defense scheme, each client compresses the parameter before uploading the parameters to the central server. The model parameters which are obtained after the global model performs the federated averaging algorithm are the same as the final parameters of each round of the global model. By calculating the calculation of norm between the parameters and the initial parameters, all the norms of the global model during the federated learning process can be obtained. The

Table 1: Comparison between Gaussian noise and parameter compression.

Defense methods		1^{th} round	150^{th} round	Decrease (%)
Baseline		0.9591	0.9507	0.84
Gaussian noise	$\text{Noise}_{\text{scale}} = 10^{-4}$	0.9560	0.9440	1.20
	$\text{Noise}_{\text{scale}} = 10^{-3}$	0.9570	0.9428	1.42
	$\text{Noise}_{\text{scale}} = 10^{-2}$	0.9334	0.8035	14.1
Parameter compression	Comlevel = 0.1	0.9585	0.9406	1.79
	Comlevel = 0.01	0.9593	0.9456	1.37
	Comlevel = 0.001	0.9565	0.9557	0.08

data set and model selected in the experiment are consistent with the previous experiment, which is the MNIST data set and the simplest three-layer neural network model. A parameter compression scheme is applied during federated learning and the norm and accuracy of the global model are recorded.

Each round of global model test accuracy is taken as the horizontal axis and the norm as the vertical axis. The curves are shown in Figure 13 which demonstrates that the scatter is trending upward and the points on each line go from sparse to dense. As the accuracy of the model increases, the value of the parameter norm also increases, and when the accuracy of the model increases more and more slowly, the norm increases more and more slowly, indicating that the model parameters change more and more less, and the less compressed curve is closer to the baseline curve. As can be seen from the figure, the change of each curve has a certain trend, and there is no large up and down fluctuation or deviation point. Therefore, the accuracy around a certain norm value is not very different, within a range.

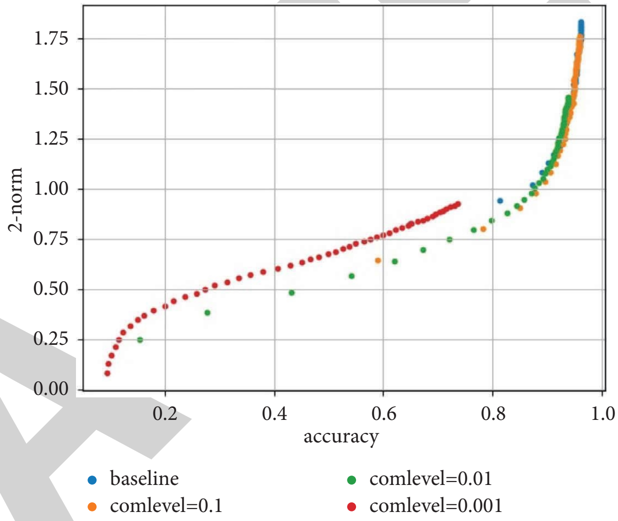

Figure 13: Relationship diagram between norm and accuracy.

Similarly, adding different scales of noise to the original parameters can obtain parameters with different noise scales. The accuracy of the model which loads the parameter with different noise can be calculated on the test set. A three-layer neural network model is trained with the MNIST dataset, and noises of different sizes are added to the current initial model parameters when the model is 81.89% accurate on the test set. The scatter plot of norm and accuracy with the parameters under different noise is shown in Figure 14, and the corresponding standard deviations are from 0.001 to 0.02 and the scatter points under 20 noises are from small to large. The horizontal axis is the norm of the noise parameter and the initial parameter, and the vertical axis is the accuracy of the model on the test set. It can be seen that as the noise continues to increase, the norm value continues to increase, and the model accuracy continues to decrease. The distribution of the overall points in the figure shows a downward trend, and there are no particularly abrupt abnormal points, indicating that there is a certain relationship between the performance of the model and the norm.

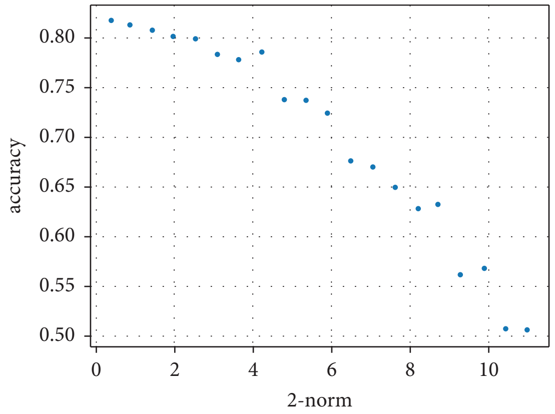

Figure 14: Relationship diagram between norm and accuracy.

From the previous experiments, it can be seen that no matter whether parameter compression or adding noise, there is no obvious deviation in the relationship between the norm and the model performance. In other words, the model performances are similar. Therefore, the learning situation of the model can be compared by comparing the norm of the model, that is, the change of the parameters of the model.

5.2. Norm and Strength of Protection. Without taking any defensive measures, the attacker can steal the victim's private data in the GAN attack scenario. Then, it can be seen from the previous experimental analysis that if the norm of the scheme with defensive measures is similar to the norm of no defensive measures; it can be considered that the model learning results are similar, the information leaked to the attacker is also similar, and the private data will still be leaked to protection. If the norm is far apart, the model learning results are far apart, and the leaked information is relatively less, which is more likely to protect privacy. Therefore, the method to measure the degree of privacy protection of parameter noise and parameter compression is to calculate the norm of each round of the global model under different defense schemes and draw its norm change curve, the closer it is to the original no defense measures. The farther the curve is, the less it can prevent the leakage of private data, and the farther the curve is, the more likely it is to protect the private data from being leaked. Specifically, the initial parameters of the global model are first recorded as W_0.

During the learning process, W^t_{gaussian} is the parameter of the t th round in the scheme of Gaussian noise defense method, W^t_{compress} is the parameter of the t th round in the scheme of parameter compression, and W^t_{original} is the parameter of the t th round without any protect. The norm of W_0 is calculated with W^t_{gaussian}, W^t_{compress}, and W^t_{original} after each round of global model aggregation, respectively. And, the norm of W_0 with W^t_{original} is used as the baseline.

For example, with the parameter compression scheme, the calculation process is shown in Figure 15. $W_1, W_2, \ldots, W_n$ represent the parameter of global model in the t th round, which will be compressed and upload to the central server. W_{compress} is the parameter that aggregated with FedAvg; then, the norm of the W_0 with W_{compress} can be calculated. After the norm of all rounds is calculated, its norm change curve will be drawn.

In the group of experiments with adding noise, the norm of the calculated global model is shown in Figure 16. The red curve is the norm without any protection method. When noise$_{\text{scale}}$ is 10^{-3} and 10^{-4}, the curves are almost overlap to the baseline curve. But when noise$_{\text{scale}}$ is 10^{-2}, the curves is far away from the baseline curve.

It can be seen from the previous parameter noise defense experiment that the defense results conform to the assumptions. When noise$_{\text{scale}}$ is 10^{-3} or 10^{-4}, the parameter is similar to the baseline's. Therefore, the local data cannot be protected from privacy leakage. And, there is a great difference between the parameter of the situation when noise$_{\text{scale}}$ is 10^{-2} and the baseline. Only in that case, the sensitive data are protected.

In the group of experiments with parameter compression, the norm of the calculated global model is shown in Figure 17. The red curve is the baseline method. The distance between the curves grows as the compression ratio decreases.

It can be seen from the previous parameter noise defense experiment that only in the situation that the comlevel is 0.001, the sensitive data are protected completely and the attacker cannot get any private information.

By comparing the model parameter norms, it shows that both the noise addition and compression schemes protect privacy data by making the parameters deviate from the original parameters. That is, the more the deviation, the better the privacy protection. However, if the parameters deviate too much, it will affect the accuracy of the model.

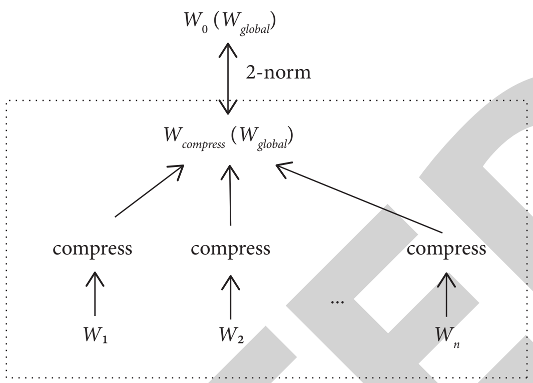

Figure 15: Norm of parameter compression diagram.

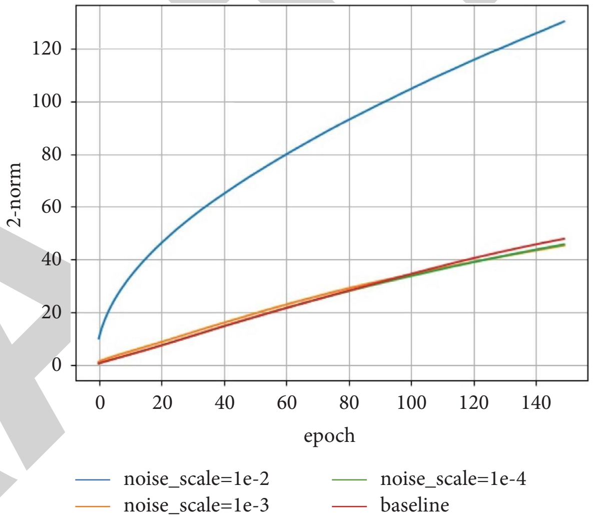

Figure 16: Norm of global model by adding noise.

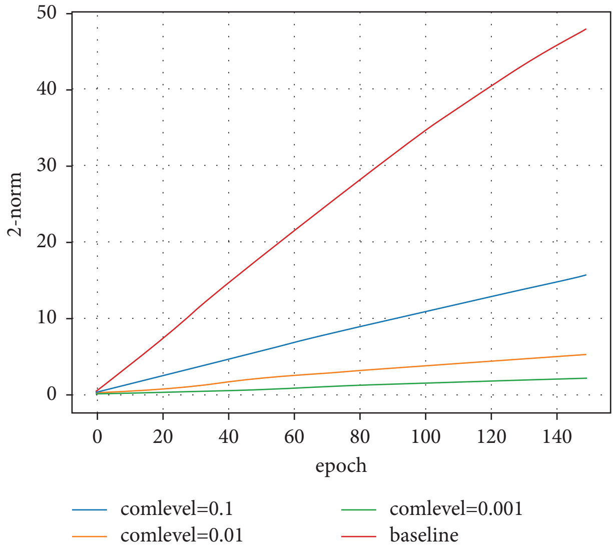

Figure 17: Norm of global model by parameter compressions.

Table 1 shows the specific accuracy rates of the parameter compression and parameter noise addition schemes of the two defense schemes. The Gaussian noise scheme can protect the victim's private data from being obtained when $noise_{scale}$ is 10^{-2}, but the model test accuracy is reduced by 14.1%. Although the accuracy of the other two noise addition schemes decreases less, they cannot prevent the leakage of private information. Therefore, although noise confusion can be added to the parameters when the noise is small, it is not enough to cover up the real sample information. When the noise is large, it directly affects the accuracy of the entire model.

And, in the case of comlevel is 0.001, the parameter compression scheme cannot only protect private information from leakage but also the model test accuracy rate is only reduced by 0.08%, which does not have a very large impact on the global model accuracy rate. Therefore, parameter compression is a desirable defense.

6. Conclusions

For the GAN-based privacy inferring attacks, experimental results demonstrate that our proposed parameter compression method, which uploads part of the parameters with the largest changes in each round, is effective in protecting data privacy.

In this way, the sharing of information is reduced to prevent private information leakage. By adopting Gaussian noise defense method, although privacy can be protected when the noise is large enough, the accuracy of the global model is reduced. Therefore, parameter compression is a better defense method, as it guarantees the accuracy of the model to a great extent by sharing only the important parameter updates. Finally, we compare the common schemes of confusing information. Several comparative experiments with different noise sizes are implemented to compare the defense effects. And, a norm hypothesis is proposed by calculating parameter changes to explain the protection of private information by the two defense methods and compared the final impact of the two on the accuracy of the global model.

The core idea of the parameter compression defense method proposed in this paper is gradient compression which was originally proposed to reduce communication costs by reducing the gradient amount to compress the gradient. The parameter compression method also reduces the exposure of data information by reducing the shared parameters so as to achieve the role of defending against GAN privacy inference attack. Besides, the performance in big model or discrete data is questionable. Therefore, studying whether the idea of gradient compression can prevent other privacy leakage problems in federated learning and how to optimize this compression algorithm to protect information can be our future work.

Data Availability

The data MNIST that supports the findings of this study are available in the public domain: https://yann.lecun.com/exdb/mnist/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by Science and Technology Research & Development Plan of China Railway Group Co., Ltd., under Grant no. N2021W005, and in part by the Open Project of the National Engineering Laboratory for Comprehensive Transportation Big Data Application Technology, under Grant no. 2022SK01-A.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Z. Wang, X. Dong, H. Xue et al., "Fairness-aware adversarial perturbation towards bias mitigation for deployed deep models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10379–10388, New Orleans, LA, USA, June, 2022.

[3] W. Wang, X. Suo, X. Wei et al., "Hgate: heterogeneous graph attention auto-encoders," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3938–3951, 2023.

[4] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and L. Qi, "Shine: signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 592–600, Marina Del Rey, CA, USA, February, 2018.

[5] A. Chan, M. Sanjabi, L. Mathias et al., "Unirex: a unified learning framework for language model rationale extraction," in *Proceedings of the International Conference on Machine Learning*, pp. 2867–2889, PMLR, Pittsburgh PA USA, July, 2022.

[6] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity," 2021, https://arxiv.org/abs/2104.08786.

[7] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. T. Sok, "The faults in our asrs: an overview of attacks against automatic speech recognition and speaker identification systems," in *Proceedings of the 2021 IEEE symposium on security and privacy (SP)*, pp. 730–747, IEEE, San Francisco, CA, USA, May, 2021.

[8] X. Li, G. Li, L. Liu, M. Meng, and S. Shi, "On the word alignment from neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1293–1303, Dublin, Ireland, May, 2019.

[9] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," 2019, https://arxiv.org/abs/1904.05734.

[10] B. Viswanath, M. Ahmad Bashir, M. Crovella et al., "Towards detecting anomalous user behavior in online social networks," in *Proceedings of the 23rd Usenix Security Symposium (usenix security 14)*, pp. 223–238, San Diego, CA, USA, August, 2014.

[11] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Information Sciences*, vol. 511, pp. 284–296, 2020.

[12] G. Li, Y. Wei, Y. Tian, C. Xu, J. R. Wen, and D. Hu, "Learning to answer questions in dynamic audio-visual scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19108–19118, New Orleans, LA, USA, June, 2022.

[13] A. Liu, S. Y. Jin, C.-I. Lai, A. Rouditchenko, A. Oliva, and J. Glass, "Cross-modal discrete representation learning," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 3013–3035, Dublin, Ireland, May, 2022.

[14] Y. Li, J. Fan, Y. Pan, T. Yao, W. Lin, and T. Mei, "Uni-eden: universal encoder-decoder network by multi-granular vision-language pre-training," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 18, no. 2, pp. 1–16, 2022.

[15] Y. Li, Y. Li, Q. Yan, and R. H. Deng, "Privacy leakage analysis in online social networks," *Computers and Security*, vol. 49, pp. 239–254, 2015.

[16] L. Li, J. Liu, L. Cheng et al., "Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.

[17] S. Terzi, C. S. Kouzinopoulos, K. Votis, D. Tzovaras, and I. Stamelos, "Securing emission data of smart vehicles with blockchain and self-sovereign identities," in *Proceedings of the IEEE International Conference on Blockchain, Blockchain 2020*, pp. 462–469, IEEE, Rhodes, Greece, November, 2020.

[18] J. Lee, H. L. Youn, N. Stevens, J. Poon, and S. C. Han, "Fednlp: an interpretable nlp system to decode federal reserve communications," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2560–2564, Canada, July, 2021.

[19] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, p. 4, 2022.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and Y A. Blaise Aguera, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, Ft. Lauderdale, FL, USA, April, 2017.

[21] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618, Dallas, TX, USA, November, 2017.

[22] L. Lyu, Y. Han, X. Ma et al., "Privacy and robustness in federated learning: attacks and defenses," 2020, https://arxiv.org/abs/2012.06337.

[23] M. Naseri, J. Hayes, and E. De Cristofaro, "Toward robustness and privacy in federated learning: experimenting with local and central differential privacy," 2020, https://arxiv.org/abs/2009.03561.

[24] H. Cao, Y. Zhu, Y. Ren et al., "Prevention of gan-based privacy inferring attacks towards federated learning," in *Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Hangzhou, China, October, 2022.

[25] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: a survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[26] S. Saha and T. Ahmad, "Federated transfer learning: concept and applications," *Intelligenza Artificiale*, vol. 15, no. 1, pp. 35–44, 2021.

[27] R. Tzaban, M. Ron, R. Gal, A. H. Bermano, and D. Cohen-Or, "Stitch it in time: Gan-based facial editing of real videos," 2022, https://arxiv.org/abs/2201.08361.

[28] L. Yang, Z. Zhang, Y. Song et al., "Diffusion models: a comprehensive survey of methods and applications," 2022, https://arxiv.org/abs/2209.00796.

[29] Y. Shi, X. Yang, Y. Wan, and X. Shen, "Semanticstylegan: learning compositional generative priors for controllable image synthesis and editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11254–11264, New Orleans, LA, USA, June, 2022.

[30] M. Ding, Z. Yang, W. Hong et al., "Cogview: mastering text-to-image generation via transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19822–19835, 2021.

[31] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10132–10142, New Orleans, LA, USA, June, 2022.

[32] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: user-level privacy leakage from federated learning," in *Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, IEEE, Paris, France, April, 2019.

[33] C. Fu, X. Zhang, S. Ji et al., "Label inference attacks against vertical federated learning," in *Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, USA, March, 2022.

[34] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 2587–2596, IEEE, Los Angeles, CA, USA, December, 2019.

[35] W. Wei, L. Liu, Y. Wut, S. Gong, and A. Iyengar, "Gradient-leakage resilient federated learning," in *Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pp. 797–807, IEEE, Washington DC, USA, July, 2021.

[36] K. Wei, J. Li, M. Ding et al., "Federated learning with differential privacy: algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[37] Q. Zheng, S. Chen, L. Qi, and W. Su, "Federated f-differential privacy," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 2251–2259, PMLR, Valencia, Spain, April, 2021.

[38] J. Liu, Y. Tian, Y. Zhou, Y. Xiao, and N. Ansari, "Privacy preserving distributed data mining based on secure multi-party computation," *Computer Communications*, vol. 153, pp. 208–216, 2020.

[39] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: provable defense against privacy leakage in federated learning from representation perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9311–9319, Nashville, TN, USA, June, 2021.

[40] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[41] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning," in *Proceedings of the Network and*

Distributed System Security Symposium, San Diego, California, USA, February, 2021.

[42] Y. Li, Y. Jiang, Z. Li, and S. T. Xia, "Backdoor learning: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–18, 2022.

[43] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622, San Diego, CA, USA, August, 2020.

[44] Z. Lv, H. Cao, F. Zhang et al., "Awfc: preventing label flipping attacks towards federated learning for intelligent iot," *The Computer Journal*, vol. 65, no. 11, pp. 2849–2859, 2022.

[45] Y. Dong, X. Yang, Z. Deng et al., "Black-box detection of backdoor attacks with limited information and data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16482–16491, Montreal, Canada, October, 2021.

[46] Q. Han, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: an evaluation framework for mitigating dnn backdoor attacks using data augmentation," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pp. 363–377, Auckland, New Zealand, June, 2021.

[47] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3035–3043, apr 2018.

[48] N. McLaughlin, J. Martinez del Rincon, B. J. Kang et al., "Deep android malware detection," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 301–308, Scottsdale, AZ, USA, March, 2017.

[49] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, "A survey of android malware detection with deep neural models," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–36, 2021.

[50] S. Y. Yerima, S. Sezer, G. McWilliams, and I. Muttik, "A new android malware detection approach using bayesian classification," in *Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 121–128, IEEE, Washington, DC; USA, March, 2013.

[51] X. Liu, J. Liu, S. Zhu, W. Wang, and X. Zhang, "Privacy risk analysis and mitigation of analytics libraries in the android ecosystem," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1184–1199, 2020.

[52] G. Adam, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: attacking deep reinforcement learning," in *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020*, OpenReview.net, Addis Ababa, Ethiopia, April, 2020.

[53] K. Mahmood, R. Mahmood, and M. V. Dijk, "On the robustness of vision transformers to adversarial examples," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pp. 7818–7827, IEEE, Montreal, QC, Canada, October, 2021.

[54] X. Dong, A. T. Luu, R. Ji, and H. Liu, "Towards robustness against natural language word substitutions," in *Proceedings of the 9th International Conference on Learning Representations, ICLR 2021*, OpenReview.net, Austria, May, 2021.

[55] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, and Z. Chen, "Sysevr: a framework for using deep learning to detect software vulnerabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 4, pp. 2244–2258, 2022.

[56] D. Zou, S. Wang, S. Xu, Z. Li, and H. Jin, "A deep learning-based system for multiclass vulnerability detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 1–2236, 2019.

[57] Z. Li, D. Zou, S. Xu, H. Jin, H. Qi, and J. Hu, "Vulpecker: an automated vulnerability detection system based on code similarity analysis," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 201–213, Los Angeles, CA, USA, December, 2016.

[58] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "ContractWard: automated vulnerability detection models for ethereum smart contracts," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1133–1144, apr 2021.

[59] J. Song, W. Wang, T. R. Gadekallu, J. Cao, and Y. Liu, "Eppda: an efficient privacy-preserving data aggregation federated learning scheme," *IEEE Transactions on Network Science and Engineering*, vol. 1, 2022.

[60] W. Wang, M. H. Fida, Z. Lian et al., "Secure-enhanced federated learning for ai-empowered electric vehicle energy prediction," *IEEE Consumer Electronics Magazine*, vol. 12, 2021.

[61] Z. Lian, W. Wang, H. Huang, and C. Su, "Layer-based communication-efficient federated learning with privacy preservation," *IEICE - Transactions on Info and Systems*, vol. 105, no. 2, pp. 256–263, 2022.

[62] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.