Prevention of GAN-Based Privacy Inferring Attacks Towards Federated Learning

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

[1] Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, No.3 Shangyuancun, Beijing 100044, China
{hongbo.cao,19125226,wangwei1}@bjtu.edu.cn
[2] School of electronic information engineering, Beijing Jiaotong University, No.3 Shangyuancun, Beijing 100044, China
[3] Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China
{zhuys,wangwq}@rails.cn
[4] Zhejiang Key Laboratory of Multi-dimensional Perception Technology, Application and Cybersecurity, Hangzhou 310053, China
bin_wang@zju.edu.cn
[5] iFLYTEK Co., Ltd,Hefei, China
mqhu3@iflytek.com

Abstract. With the increasing amount of data, data privacy has drawn great concern in machine learning among the public. Federated Learning, which is a new kind of distributed learning framework, enables data providers to train models locally to protect privacy. It solves the problem of privacy leakage of data by enabling multiple parties, each with their training dataset, to share the model instead of exchanging private data with the server side. However, there are still threats of data privacy leakage in federated learning. In this work, we are motivated to prevent GAN-based privacy inferring attacks in federated learning. For the GAN-based privacy inferring attacks, 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 privacy information of victims while maintaining the accuracy of the global model. Comprehensive experimental results demonstrated that our method is effective in the prevention of GAN-based privacy inferring attacks.

Keywords: Federated learning · Inferring attacks · Generative adversarial network · Intrusion detect · Parameter compress

1 Introduction

Deep learning, which is the most popular machine learning method driven by big data, has been widely used in various domain like image recognition [1],

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 39–54, 2022.
https://doi.org/10.1007/978-3-031-24386-8_3

social networks [2], speech technology [3], natural language process [4] and face detection [5]. However, the centralized data storage currently has many problems. First, if all training data is stored and trained in a centralized manner, the transmission of data requires a very large communication cost. Second, training learning algorithms on large dataset requires higher performance computing equipment. Third, many scenarios cannot be relied on because of the absence of trust boundaries. If personal information is submitted to the outside, users will face privacy leakage risks because they cannot control how their data will be used after sharing, which will directly disclose important personal privacy information and may cause serious privacy problems [6]. Therefore, it is important to run a machine learning in a way that protects sensitive data from privacy leakage.

Federated learning (FL) [7,8] is a novel machine learning paradigm to solve this problem. In federated learning, the data owner must participate in the whole learning process instead of relying on a trusted third party. Federated learning was first proposed by Google [7]. It is a server-client architecture consisting of a parameter server and multiple clients. The server and the client carry out multiple rounds of iterative communication and collaborate to train a global model. Private data is stored in a locally isolated device and will not be shared with other parties during the training process, which not only guarantees users' privacy and data security but also solves the problem of data fragmentation and isolation.

Although FL shows superb performance in privacy-preserving and breaks data silos effectively, it's still surprisingly susceptible to GAN-based data reconstruction attacks [9], which is a kind of privacy inference attack [10] in the training phase of FL.

Existing relatd work show that differential privacy (DP) [11] is regarded as one of the strongest defense methods against these attacks. The core idea of DP is introducing random noise into the privacy information, but DP often adds so sufficient noise that the accuracy of the global model is reduced notably.

To address this problem, we focus on the inference attacks toward Non-i.i.d federated learning. In addition, we conduct various experiments to evaluate the privacy leakage that the adversary can get from the parameter of the global model during the training phase and understand the relationship between the reconstruction sample and global model information leakage. Thus, we find parameter compression is an effective defense method against GAN-based reconstruction attacks toward federated learning.

Our contributions can be summarized as the follows:

- We reveal that the gan-based privacy inferring attacks toward federated learning is defensible.
- We propose an efficient defense method to protect sensitive data against inferring attacks toward federated learning.
- We compare our method with the current defense method that adds noise to the parameter and the experiment result shows our method is better.

2 Related Work

2.1 Overview of Federated Learning

It has been well recognized that FL is a peculiar form of collaborative machine learning technique. FL allows the participants 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 process of FL can be expressed as follows.

(1) Client Initialization: The participants 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 participant 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 participants.

Categorization of Federated Learning. Based on the characteristics of the data distribution [10], federated learning can be classified into three general types.

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, which focuses on FL, assumes that the model is trained in HFL.

VFL, which is also called heterogeneous federated learning, is suitable for the situation where the participants have the Non-i.i.d datasets [12]. Meanwhile, sample space is shared between participants who have different label spaces or feature spaces.

FTL [13] is suitable for situations similar to that of traditional transfer learning [14], which aims to leverage knowledge from previously available source tasks to solve new target tasks.

Threats in Federated Learning. FL is vulnerable to adversarial attacks such as unauthorized data-stealing or debilitating global model [15]. The adversary mainly focuses on both the privacy attacks and robustness attacks towards centralized federated learning.

In privacy attacks that often occur in the training phase, 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 participants. The privacy leakage of the sample reconstruction attacks may come from model gradients [16], loss function [17] or model parameters [9]. Furthermore, the sample reconstruction attacks are considered not only on the

client-side but on the server-side [18]. Besides, Fu et al. [19] proposed a label inference attack which is in a special and interesting Non-i.i.d. federated learning setting. Existing relatd work regard differential privacy as an efficient method to defend the privacy inference attack [20]. In the local differential privacy, the FL clients add Gaussian noise to the local gradients or parameters.

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 [21]. The goal of the adversary is to diminish the performance and the convergence of the global model. These misclassifications may cause serious security problems.

Besides, the backdoor attack [22] is known for its higher impact of its capabilities to set the trigger. It's an effective targeted method to attack FL system. Various robust aggregation algorithms are proposed to defend against poisoning attacks towards FL, such as Krum [23], Bulyan [24], Median [25] and Fang [26].

There also exist related work on the prevention of Android malware [27–33], on the detection of software vulnerabilities [34], on the detection of network anomalies [35], or on enhancing the privacy in other scenarios like communications of smart vehicles [36].

2.2 Generative Adversarial Networks

In the field of deep learning, generative adversarial networks (GANs) [37] have recently been proposed, and they are still in a highly developed and researched stage [38]. Various GANs has been proposed. They can be used to generate deepfake face [39], generate image by text [40]. The goal of 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 touching the original samples.

The training of GAN network is a typical game confrontation process of finding the maximum and minimum values. The game between discriminator and generator is shown as in formula 1.

$$\begin{aligned} \min_G \max_D V(G,D) = & E_{x\sim p_{data}(x)} log[D(x)] \\ & + E_{z\sim p_z(z)} log[1-D(x)] \end{aligned} \tag{1}$$

When the discriminator D cannot distinguish between the samples in the original data and the samples generated by the generator G, the training process ends.

Hitaj et al. [9] first proposed a GAN-based reconstruction attack. In the attack, malicious participants in the system steal 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. Without anyone in the system noticing,

Fig. 1. The image recovered by the attacker

the attacker can trick the victim into releasing more information about their training data, and eventually recover the victim's sample data.

For the GAN-based privacy inferring attacks, Yan et al. [41] proposed to detect the GAN-based privacy inferring attacks by setting hidden points on the parameter server side, and adjusting the parameters of the model to make the training model GAN invalid. Since GAN must alternately optimize G and D to achieve the optimal synchronization, G may collapse if the optimal balance between G and D is not reached. Since the learning rate has a great influence on the training process, this method disrupts the training process of GAN and makes it invalid by changing the learning rate.

The GAN-based privacy inferring attacks aim to reconstruct recognizable data images from the victim's personal data information. The GAN effectively learns the distribution of training data. In order to prevent such attacks, Luo et al. [42] proposed an Anti-GAN framework to prevent attackers from learning the true distribution of victim data by adding fake images into the source real image. The victim first inputs the personal training data into the GAN generator, and then inputs the generated fake images into the global model for training of federated learning. Besides, the author designed a new loss function so that the images generated by the victim's GAN not only have classification features similar to the original training data, but also have indistinguishable visual features to prevent privacy inferring attacks.

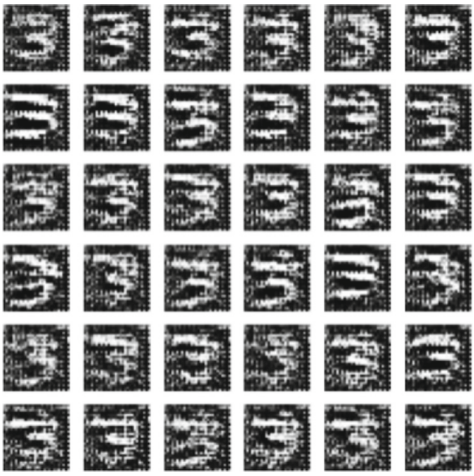

Fig. 2. Defense result when R% = 90%

2.3 Compression

Distributed Stochastic Gradient Descent algorithm has been widely used in the training of large-scale deep learning models, and the communication cost between working nodes has become a new system bottleneck.

Gradient compression is a solution that improves communication efficiency by compressing the gradient of transmission. In general, the gradient change of the model parameters in each iteration is relatively small, most of the parameters are still the same as before. So there will be a lot of redundant parameters during the transmission process, but the attacker can use the redundancy parameters updates to reconstruct the sample. Gradient compression uses this feature to compress the gradient generated in each iteration. This method reduces the amount of gradient in communication and reduces the burden on bandwidth by sending a sparse vector of a subset of important values in the gradient.

There are many optimization algorithms for Gradient Compression. For example, Lin et al. [43] proposed the Deep Gradient Compression algorithm to preserve the model accuracy in the gradient compression process. In order to reduce the gradient sparsification time, Shi et al. [44] proposed an optimal algorithm to find the trade-off between communication cost and sparsification time cost. There are also optimization algorithms of adaptive compression ratios to increase the flexibility of compression schemes [45], etc.

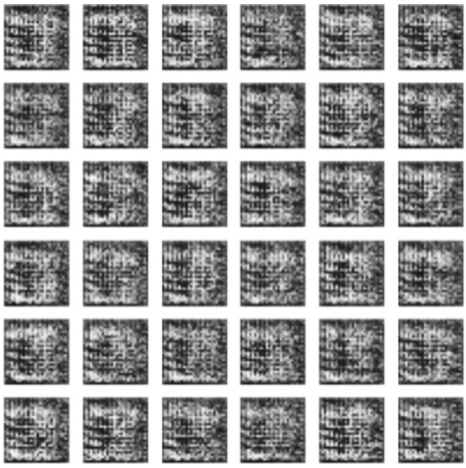

Fig. 3. Defense result when R% = 99%

3 Defense Against GAN-Based Privacy Inferring Attacks

3.1 Threat Model of GAN-Based Privacy Inferring Attacks

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

The attacker pretends to be an honest participant in the federated learning system, but tries to extract information about local data owned by other participants. The attacker builds a GAN model locally. At the same time, the attacker follows a protocol that is agreed upon by all participants. 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 participants. He tricks the victim into revealing more information about his local data.

Adversary A participates in the collaborative deep learning protocol. All such participants 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 participant (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 [42].

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

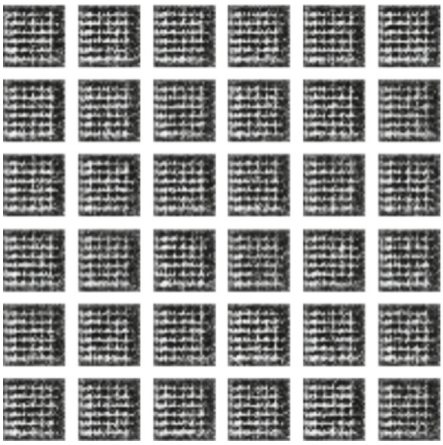

Fig. 4. Defense result when R% = 99.9%

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.

Algorithm 1. Parameter Compression on client C

Require: *parameters* $w = \{w[0], w[1], ..., w[n]\}$
1: **for** $j = 0$ to n **do**
2: $diff \Leftarrow w_t[j] - w_{t-1}[j]$
3: $count \Leftarrow |diff|$
4: $k \Leftarrow count \cdot (1 - R\%)$
5: $w_{compressed}[j] \leftarrow top_k(abs(diff)) + w_{t-1}[j]$
6: **end for**
7: *C submit* $w_{compressed}$ *to server*

3.2 Parameter Compression Method

There is a gradient compression method in distributed learning, which reduces the communication overhead by compressing the gradient in each communication round. Gradient sparsification is a kind of gradient compression. The sparsification algorithm decides to send a small part of the gradient to participate 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

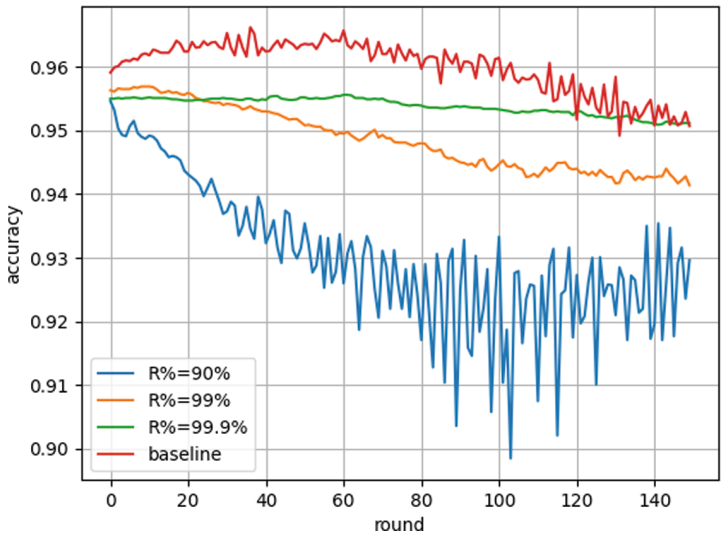

Fig. 5. Test accuracy of global model

maximum change of 1-R% was finally chosen. Usually, the compression ratio is 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 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. Define R% 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.

Fig. 6. Defense result when $noise_{scale}$ is 10^{-4}

3.3 Experiments

Datasets and CNN Architectures 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 [46]. 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 layers are in a feed-forward fully connected manner. The neural networks are trained by Tensorflow.

Results. The defense of GAN-based privacy inferring attacks takes the attack experiment of reconstructing the digital image of "3" as an example. Figure 1 shows the victim's data finally reconstructed by the attacker. It can be seen that the attacker recovers a very clear image.

The results of the parameter compression scheme are as follows. When R% = 90%, the image finally recovered by the attacker is shown in Fig. 2. 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 isn't high enough to prevent information leakage.

When R% = 99%, the attacker eventually recovers an image like Fig. 3. 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 Fig. 4. 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.

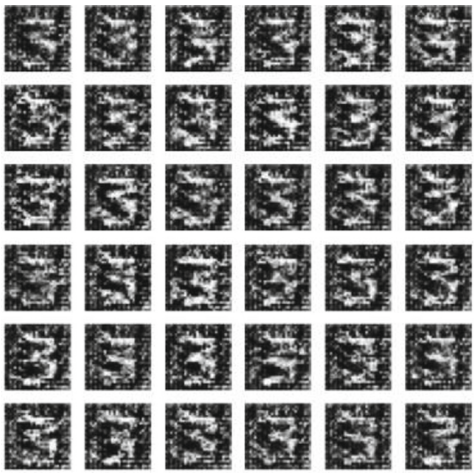

Fig. 7. Defense result when $noise_{scale}$ is 10^{-3}

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 5 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.

3.4 Compare 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.

Experiments. 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 $noise_{scale}$. And $noise_{scale}$'s value is set as 10^{-4}, 10^{-3}, 10^{-2}.

When $noise_{scale} = 10^{-4}$, the noise added is the smallest. It can be seen that it cannot prevent the leakage of data information, as shown in Fig. 6. When $noise_{scale} = 10^{-3}$, the final image recovered by the attacker is shown in Fig. 7. Although the image is more noisy than when $noise_{scale} = 10^{-4}$, there are very few outlines of the number three. When $noise_{scale} = 10^{-2}$, the image finally recovered by the attacker is shown in Fig. 8. At this time, the content of the image is completely invisible. The attacker can not obtain any valuable information about the digital image 3, which indicates that the attack failed.

From the above 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 9 is the accuracy change curve of the global model on the test dataset. When $noise_{scale}$ =10^{-3} and $noise_{scale}$ =10^{-4}, the final accuracy of the global model is similar to that of the baseline, both around 95%. But when $noise_{scale}$ =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.

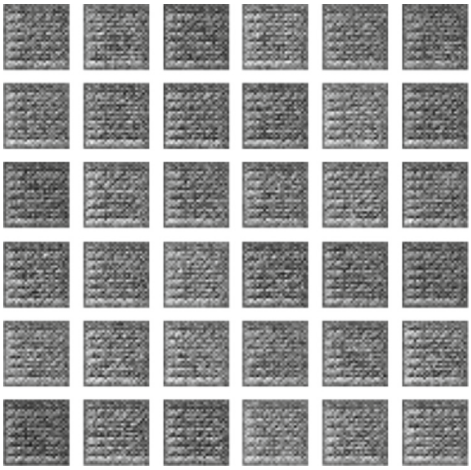

Fig. 8. Defense result when $noise_{scale}$ is 10^{-2}

Analysis of Privacy Protection. From the above experiments, it can be seen that although noise can be added to the parameters when the noise is small, it is not enough to cover up the information of the real samples. When the noise is large, it directly decreases the accuracy of 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. 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 [9]. 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

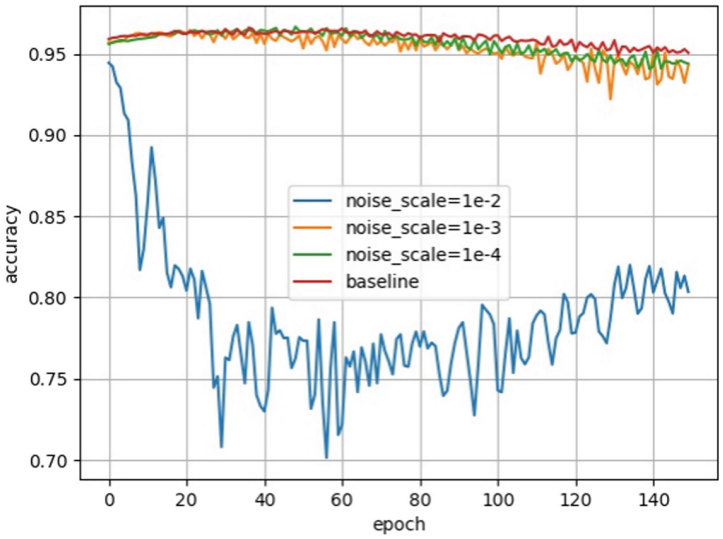

Fig. 9. Test accuracy of global model

compression is an efficient defense method that prevents GAN-based privacy inferring attacks.

4 Conclusion

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.

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. 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.

Acknowledgement. This work was supported in part by National Key R&D Program of China, under Grant 2020YFB2103802, in part by the National Natural Science Foundation of China, under grant U21A20463 and in part by the Fundamental Research Funds for the Central Universities of China under Grant KKJB320001536.

References

1. Yan, K., Wang, X., Du, Y., Jin, N., Huang, H., Zhou, H.: Multi-step short-term power consumption forecasting with a hybrid deep learning strategy. Energies **11**(11), 3089 (2018)
2. Wang, W., et al.: Hgate: Heterogeneous graph attention auto-encoders. IEEE Transactions on Knowledge and Data Engineering, pp. 1–1 (2021). https://doi.org/10.1109/TKDE.2021.3138788
3. Sharma, U., Maheshkar, S., Mishra, A.N., Kaushik, R.: Visual speech recognition using optical flow and hidden markov model. Wireless Pers. Commun. **106**(4), 2129–2147 (2019)
4. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
5. Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., Nanduri, A., Chen, J.C., Castillo, C.D., Chellappa, R.: A fast and accurate system for face detection, identification, and verification. IEEE Trans. Biomet., Behav. Identity Sci. **1**(2), 82–96 (2019)
6. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp. 1310–1321 (2015)
7. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, pp. 1273–1282. PMLR (2017)
8. Liu, M., Ho, S., Wang, M., Gao, L., Jin, Y., Zhang, H.: Federated learning meets natural language processing: A survey. arXiv preprint arXiv:2107.12603 (2021)
9. Hitaj, B., Ateniese, G., Perez-Cruz, F.: 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 (2017)
10. Lyu, L., Yu, H., Ma, X., Sun, L., Zhao, J., Yang, Q., Yu, P.S.: Privacy and robustness in federated learning: Attacks and defenses. arXiv preprint arXiv:2012.06337 (2020)
11. Naseri, M., Hayes, J., De Cristofaro, E.: Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. arXiv e-prints pp. arXiv-2009 (2020)
12. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-iid data: a survey. Neurocomputing **465**, 371–390 (2021)
13. Saha, S., Ahmad, T.: Federated transfer learning: concept and applications. Intelligenza Artificiale **15**(1), 35–44 (2021)
14. Maschler, B., Weyrich, M.: Deep transfer learning for industrial automation: a review and discussion of new techniques for data-driven machine learning. IEEE Ind. Electron. Mag. **15**(2), 65–75 (2021)
15. Liu, P., Xu, X., Wang, W.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity **5**(1), 4 (2022)

16. Zhao, B., Mopuri, K.R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)
17. Sannai, A.: Reconstruction of training samples from loss functions. CoRR abs/1805.07337 (2018), http://arxiv.org/abs/1805.07337
18. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: User-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp. 2512–2520. IEEE (2019)
19. Fu, C., Zhang, X., Ji, S., Chen, J., Wu, J., Guo, S., Zhou, J., Liu, A.X., Wang, T.: Label inference attacks against vertical federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), Boston, MA (2022)
20. Triastcyn, A., Faltings, B.: Federated learning with bayesian differential privacy. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 2587–2596. IEEE (2019)
21. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
22. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics, pp. 2938–2948. PMLR (2020)
23. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
24. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning, pp. 3521–3530. PMLR (2018)
25. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659. PMLR (2018)
26. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 1605–1622 (2020)
27. Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., Zhang, X.: Exploring permission-induced risk in android applications for malicious application detection. IEEE Trans. Inf. Forensics Secur. **9**(11), 1869–1882 (2014)
28. Wang, W., Zhao, M., Wang, J.: Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. J. Ambient Intell. Human. Comput. **1**0(8), 3035–3043 (2018)
29. Fan, M., Liu, J., Wang, W., Li, H., Tian, Z., Liu, T.: DAPASA: detecting android piggybacked apps through sensitive subgraph analysis. IEEE Trans. Inf. Forensics Secur. **12**(8), 1772–1785 (2017)
30. Wang, W., Li, Y., Wang, X., Liu, J., Zhang, X.: Detecting android malicious apps and categorizing benign apps with ensemble of classifiers. Future Gener. Comput. Syst. **78**, 987–994 (2018)
31. Su, D., Liu, J., Wang, W., Wang, X., Du, X., Guizani, M.: Discovering communities of malapps on android-based mobile cyber-physical systems. Ad Hoc Netw. **80**, 104–115 (2018)
32. Wang, X., Wang, W., He, Y., Liu, J., Han, Z., Zhang, X.: Characterizing android apps' behavior for effective detection of malapps at large scale. Future Gener. Comput. Syst. **75**, 30–45 (2017)

33. Liu, X., Liu, J., Zhu, S., Wang, W., Zhang, X.: Privacy risk analysis and mitigation of analytics libraries in the android ecosystem. IEEE Trans. Mob. Comput. **19**(5), 1184–1199 (2020)
34. Wang, W., Song, J., Xu, G., Li, Y., Wang, H., Su, C.: ContractWard: Automated vulnerability detection models for ethereum smart contracts. IEEE Trans. Netw. Sci. Eng. **8**(2), 1133–1144 (2021)
35. Wang, W., Shang, Y., He, Y., Li, Y., Liu, J.: Botmark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. Inf. Sci. **511**, 284–296 (2020)
36. Li, L., et al.: Creditcoin: a privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. IEEE Trans. Intell. Transp. Syst. **19**(7), 2204–2220 (2018)
37. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
38. Hinz, T., Fisher, M., Wang, O., Wermter, S.: Improved techniques for training single-image gans. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1300–1309 (2021)
39. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
40. Ding, M., et al.: Cogview: mastering text-to-image generation via transformers. Adv. Neural. Inf. Process. Syst. **34**, 19822–19835 (2021)
41. Yan, X., Cui, B., Xu, Y., Shi, P., Wang, Z.: A method of information protection for collaborative deep learning under gan model attack. In: IEEE/ACM Transactions on Computational Biology and Bioinformatics (2019)
42. Luo, X., Zhu, X.: Exploiting defenses against gan-based feature inference attacks in federated learning. arXiv preprint arXiv:2004.12571 (2020)
43. Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W.J.: Deep gradient compression: Reducing the communication bandwidth for distributed training. arXiv preprint arXiv:1712.01887 (2017)
44. Shi, S., Wang, Q., Chu, X., Li, B., Qin, Y., Liu, R., Zhao, X.: Communication-efficient distributed deep learning with merged gradient sparsification on gpus. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, pp. 406–415. IEEE (2020)
45. Chen, C.Y., Choi, J., Brand, D., Agrawal, A., Zhang, W., Gopalakrishnan, K.: Adacomp: Adaptive residual gradient compression for data-parallel distributed training. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
46. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)