2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence) | 978-1-6654-6263-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/Confluence56041.2023.10048854

Poisoning Attack based on Data Feature Selection in Federated Learning

Zhengqi Liu*
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
202021080909@std.uestc.edu.cn

Ziwei Liu
School of Computer Science and Engineering
Southwest Minzu University
Chengdu, China
lzw@std.uestc.edu.cn

Xu Yang
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, China
ftosfc@163.com

***Abstract*—Federated learning is proposed as a typical distributed AI technique to protect user privacy and data security, and it is based on decentralized datasets that train machine learning models by sharing model gradients rather than sharing user data. However, while this particular machine learning approach safeguards data from being shared, it also increases the likelihood that servers will be attacked. Joint learning models are sensitive to poisoning attacks and can effectively pose a threat to the global model when an attacker directly contaminates the global model by passing poisoned gradients. In this paper, we propose a joint learning poisoning attack method based on feature selection. Unlike traditional poisoning attacks, it only modifies important features of the data and ignores other features, which ensures the effectiveness of the attack while being highly stealthy and can bypass general defense methods. After experiments, we demonstrate the feasibility of the method.**

***Index Terms*—Federated Learning, Poisoning Attack, Deep Learning, Feature Selection**

I. Introduction

Nowadays, deep learning has become a common method for complex classification problems such as image analysis, speech recognition and text translation [1] [2]. For example, Internet companies use deep learning to analyze the behavior of large groups of users to improve the quality of their services. However, with increasing awareness of privacy and the introduction of corresponding laws which prohibit unauthorized use or collection of users' private data, the application of deep learning in many industries has been hindered. Therefore, it is critical for the deep learning to train robust models with strong performance while protecting user privacy.

Based on the reality, Google proposed a special distributed learning approach named Federated Learning in 2016 [3], which trains models with model sharing instead of data sharing under the coordination of a central server. In federated learning, the client ensures that sensitive private data is kept on the client side throughout the training process and is not accessible to others, only uploading local model after training to obtain a higher quality global model. It seems that this approach meets the current demand for personal privacy, but it is based on the assumption that sharing gradients between nodes does not leak private training data [4].

However, while federated learning has brought increasing attention, it has also raised questions about its security. For example, although its training method avoids direct data sharing compared with traditional machine learning, this training method makes poisoning attacks easier, and uploading gradients does not intuitively reveal whether the training data is normal and whether the gradients have been maliciously modified by other attacks. Most of the existing defense methods are filtered by setting thresholds, but the results are always unsatisfactory in the face of heterogeneous data. Even more information about the training data can be obtained from the passed shared gradient would [5]. In addition, it has been shown that the properties of clients' training data are reflected in the local model gradient [6], and it is feasible to implement reconstruction attacks from shared model gradients [7]. In addition, Zhu et al. [8] proposed a method called Deep Leak from Gradient (DLG), which obtains dummy gradients after randomly generating dummy images, and gradually transforms the dummy images into the original images by continuously optimizing the distance between the dummy gradients and the shared gradients.

Currently, poisoning attacks against distributed machine learning can be divided into two main types. Modifying the data before the model is trained so as to get a faulty training model is called data poisoning attack. If the federal learning scenario is a heterogeneous client involved in training, the heterogeneity of the data can make this attack more difficult

Zhengqi Liu is corresponding author

to detect. Backdoor attack is a type of data poisoning attack. A backdoor attack does not affect the model directly, but generates a high confidence error by planting triggers on the local model with special samples when the model's triggers recognize the special samples. Model poisoning occurs mainly before the local model is uploaded, calibrated, and cropped to share gradients. Since the model poisoning attack only makes small modifications to the normal model, the success rate of this poisoning attack is high, but its impact is less effective compared to the data poisoning attack. In this paper, we modify the data poisoning attack method to enhance the stealthiness of the data poisoning attack and demonstrate the effectiveness of the attack in our experiments.

The main contributions of this paper are as follows:

- We propose a federated learning data poisoning attack method, which first ranks the importance of data features, modifies the feature fields with high weights, and ignores the feature fields with low weights, maintaining the effectiveness of the attack while reducing the difference between the poisoned and normal models.
- We conducted experiments using the MNIST dataset, using the attack detection algorithm and the regular aggregation algorithm on the server side respectively, and proved that our algorithm can effectively bypass the attack detection algorithm and thus poison the global model.

II. Related work

With the development of federated learning, the risk of federated learning being attacked is also exposed. This section mainly details the current poisoning attacks and their defense methods.

A. Poisoning Attack

Poisoning attacks are mainly divided into data poisoning attacks and model poisoning attacks.

Data poisoning refers to an attacker's objective by maliciously modifying data prior to training.Chen et al [9] proposed a backdoor attack method for machine learning, where they used a hybrid injection strategy in their experiments to achieve an attack success rate of over 90% by injecting only a small number of toxic samples into the training set. Jiang et al. [10] proposed a thorough countermeasure model against poisoning attacks, combining either integrity attacks or availability attacks. A trade-off parameter is added to the attack model so that the attack model can trade-off between stealthiness and attack effectiveness to achieve better attack results. Bagdasaryan et al. [11] pointed out the problem of backdoor attacks in federated learning, where the backdoor may be forgotten after averaging by the federated, and designed a model-replaced backdoor poisoning attack attack method that guarantees accuracy on both the main federated learning task and the backdoor task, and under different assumptions, the joint learning effect of the attack is evaluated and shown to be significantly better than label flipping. Xie et al. [12] proposed a distributed backdoor attack method where they decompose the triggers into local triggers and embed the training tasks into different malicious attackers accordingly, and this training method has higher stealthiness.

Model poisoning differs from data poisoning in that the attacker does not need to modify the data, but instead makes malicious modifications to the model gradient after training is complete. This attack is much less targeted and aims mainly at reducing the performance of the federated average model [13]. Bhagoji et al. show that because participants in federated learning are usually very geographically dispersed and difficult to authenticate, this makes the federated learning model more vulnerable to malicious attacks. Instead, attacks on the global model can be achieved by model poisoning with only one malicious client, preventing the model from making correct classifications.

In terms of affecting the global model performance, model poisoning is less influential and more concealed compared to data poisoning because it is modified and clipped from the normal model. In contrast, data poisoning, because it is a direct modification of the data, makes the model more different from the normal model and is more easily detected by the detection algorithm.

B. Defense method

The defense against data poisoning attacks focuses on the identification of malicious clients and the detection of uploaded model parameters. Cao et al. [14] found the relationship between the number of malicious samples and the number of attackers on the effectiveness of the attack by analyzing the influencing factors of poisoning attacks. And proposed a filtering defense mechanism called Sniper. the server runs Sniper during communication to filter the update parameters of the attacker and distinguish the user type even if there are multiple attackers in the federated learning system. Blanchard et al [15] proposed an individual aggregation rule that limits the range of participant upload gradient vectors, and a Krum scheme to filter clients, and experimentally demonstrated that Krum can effectively resist model poisoning attacks. Yin et al. [16] extended the regression algorithm to federated learning based on the reweighted aggregation rule of residuals, and improved the filtering efficiency by using the median model uploaded by the client as the standard filtering model.

Most of the above defense methods against poisoning attacks by restricting the uploaded client models through thresholds, which can cause a greater threat if the above defense mechanisms can be bypassed by data poisoning attacks. To solve this problem, this paper proposes a data poisoning attack that can bypass the detection algorithm while effectively implementing the attack.

III. Methodology

A. Problem Description

The architecture of the federated learning is shown in Fig. 1. There is a server and n clients. Each client $client_i$ trains the local model with local data and then uploads the gradient g_i $(1 <= i <= N)$ to the server, which updates the global model

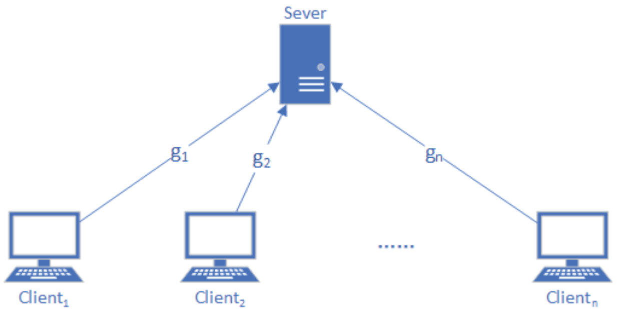

Fig. 1. Architecture of federated learning

based on the federated learning average aggregation algorithm. The aggregation algorithm is shown in (1).

$$W_{t+1} = W_t + \frac{1}{N}\sum_{n=1}^{N} \alpha \cdot g_i \tag{1}$$

where α denotes the weight of each client and $\sum \alpha = 1$. The server completes the aggregation algorithm, obtains the global model W_{t+1}, and sends it to each participating client. Throughout the training process, the clients keep the local data private and all clients can only access the global model W_{t+1} and the gradient W for each round.

In this paper, our goal is to add attackers in the federated learning training process to achieve the effect of contaminating the global model by modifying the training data, and to remain undetected during the attack.

B. Our Algorithm

In this section, we focus on how to improve the performance of the attack with as little impact on the data as possible. In most of the poison detection methods about federated learning there is a threshold mechanism to filter the uploaded models. The purpose of label flipping attack is to reduce the performance of the global model, but because it is a change of the label, which leads to a large difference between the trained classifier and the normal classifier, it is difficult to pass the threshold filtering mechanism of the server. Equation (2) is a common distance filtering equation:

$$Distance_i = \|M_G - M_i\| = \sqrt{\sum (M_G - M_i)^2} \tag{2}$$

where M_G and M_i denote the global model and the local model of client i. From the equation (2), it can be seen that if the model poisoning attack modifies the model less than the dis threshold, the detection can be bypassed. Although the modification of the model by a data poisoning attack is unpredictable, the possibility of bypassing this detection method exists as long as it affects the data information as little as possible. We first perform feature selection on the data using Sklearn's RFECV method from the open source feature selection library [17], and modify the data with higher weights. The modified data are then fed into a local model for training, resulting in a poisoned local model.

The RFECV method implements feature selection in two stages:Recursive feature elimination(RFE) and Cross Validation(CV).

In the RFE phase, the initial feature set is all available features, modeling is performed using the current feature set, then the importance of each feature is calculated, and the least important feature (or features) is removed and the feature set is updated. The modeling continues with the remaining features until the importance ratings of all features are completed.

In the CV stage, different numbers of features are selected in turn, and the selected feature set is cross-validated according to the feature importance determined in the RFE stage. The features are ranked according to MSE and the number of features with the highest average score is selected.

RFECV finds the optimal number of features by cross-validation again. If reducing a feature causes a performance loss, the feature will not be removed, and this method is very suitable for single-model features.

IV. Experiments

We implemented our algorithm on the FedML [18] framework (Python 3.7). Simulation experiments are performed in federated learning using Lasso Regression to compare the effectiveness of the proposed method for attacks with/without defense mechanisms, respectively.

A. Setup

The dataset used in the experiments is MNIST [19], containing a total of 69035 samples, where the ratio of training set to test set is 6:1. We randomly assigned these data to 30 clients.

For feature selection, we use Lasso Regression as an iterative model. The features are ranked based on the weight coefficients and the worst features are excluded, and then the process is repeated on the remaining features until all the features are traversed. For Cross Validation, we use the K-fold cross validation method and set K=5. We calculate the MSE_i of the model on the test set according to equation (3) and use the average value as MSE.

$$MSE_i(y, \hat{y}) = \frac{1}{5}\sum_{i=1}^{5}(y_i - \hat{y})^2 \tag{3}$$

$$MSE = \frac{1}{5}\sum_{i=1}^{5} MSE_i \tag{4}$$

B. Control experiments

We set the attacker ratio in the simulation experiments to $frac15$ and $frac13$, respectively, and the attackers behave the same as normal clients except that they use malicious datasets. On the server side, the data used for verification is from the test set part of MNIST, which is independent of the client data.

In our experiments, we implement the data poisoning attack described in the previous sections. We use RFECV to rank the features of 10 classes of data in the MNIST dataset, while keeping the labels unchanged, inverting the high weight features, and reducing their feature values. The Federated Learning process is performed for 40 iterations, each set of experiments is repeated 10 times, and the results of all experiments are averaged.

C. Result

Our use of MNIST for heterogeneous datasets, although the accuracy of training results is lower compared to models trained on independent homogeneous datasets, after several simulation experiments, we can determine that the results of model training do not affect the execution of the attack.

Fig. 2. shows the accuracy of simulated heterogeneous MNIST dataset training when the number of clients is 30 and the ratio of attackers is $frac15$. In this case, the yellow and green lines represent the federated learning system without defense mechanism, and with distance-based checking mechanism, respectively. As a control, we use the blue line as the accuracy rate that the global model should have after normal training. It can be seen that the performance of the global model degrades significantly after attack, and our attack can effectively bypass the distance checking mechanism of the server.

Fig. 3. shows the accuracy of the simulated heterogeneous MNIST dataset training with a number of clients of 30 and a proportion of attackers of $frac13$. It can be seen that the increase of attackers brings worse impact on the federated learning system.However, the defense mechanism plays a greater role at $frac15$ compared to $frac13$, although the model is still unusable at this point.

We simulated poisoning attacks on the federated learning system in our experiments using 1/5 and 1/3 proportions of attackers, and the global model decreased from 78.15%, to 61.57% and 32.86%, respectively, after the attacks. Although the distance checking mechanism can mitigate the effect of the attack, it still cannot perform effective defense.

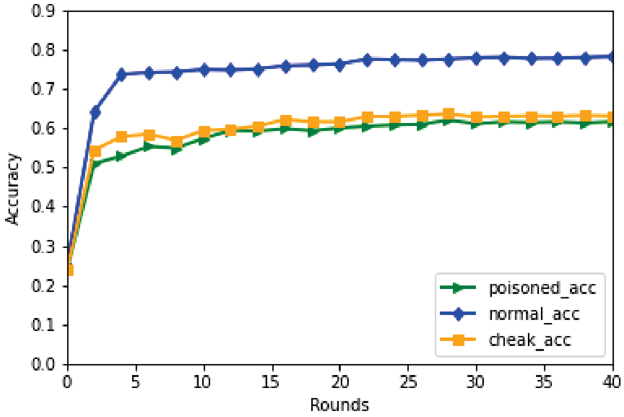

Fig. 2. Attackers are 1/5 of the total clients

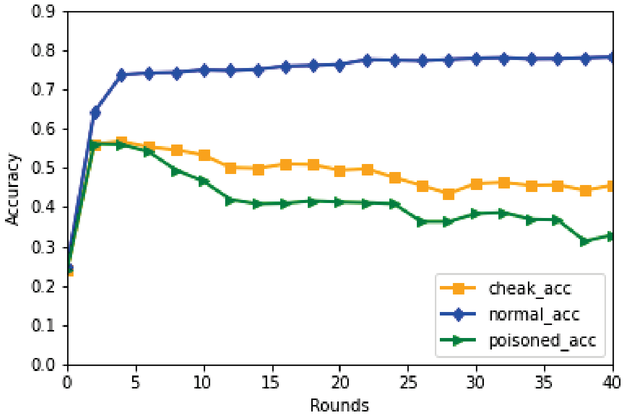

Fig. 3. Attackers are 1/3 of the total clients

V. Conclusion

Federated learning has been proposed as a privacy-preserving solution, and it is widely believed to be secure enough. However, as the research progresses, scholars continue to challenge the security of federated learning. In this paper, we propose a data poisoning attack method based on feature selection and validate it in experiments on the MNIST dataset. The shortcoming is that we have only completed the validation of our idea in MNIST. In the future, we will conduct experiments on more network models to improve the generality of our method and study the corresponding defense strategies of our attack method.

References

[1] A. Jochems, T. M. Deist, I. El Naqa, M. Kessler, C. Mayo, J. Reeves, S. Jolly, M. Matuszak, R. Ten Haken, J. van Soest, *et al.*, “Developing and validating a survival prediction model for nsclc patients through distributed learning across 3 countries,” *International Journal of Radiation Oncology* Biology* Physics*, vol. 99, no. 2, pp. 344–352, 2017.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, *et al.*, “Towards federated learning at scale: System design,” *arXiv preprint arXiv:1902.01046*, 2019.

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, 2016.

[4] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.

[5] Y. Aono, T. Hayashi, L. Wang, S. Moriai, *et al.*, “Privacy-preserving deep learning via additively homomorphic encryption,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[6] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” pp. 691–706, 2019.

[7] Y. Aono, T. Hayashi, L. Wang, S. Moriai, *et al.*, “Privacy-preserving deep learning: Revisited and enhanced,” pp. 100–110, 2017.

[8] L. Zhu and S. Han, “Deep leakage from gradients,” pp. 17–31, 2020.

[9] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” *arXiv preprint arXiv:1712.05526*, 2017.

[10] W. Jiang, H. Li, S. Liu, Y. Ren, and M. He, "A flexible poisoning attack against machine learning," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2019.

[11] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020.

[12] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2019.

[13] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[14] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 233–239, IEEE, 2019.

[15] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, pp. 5650–5659, PMLR, 2018.

[17] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM computing surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[18] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[19] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.