Research on robustness of federated learning based on pruning optimization

Lei Zhao*, Yitong Chen
State Grid Jiangsu Electric Power Co., Ltd, Nanjing 210022, Jiangsu, China

ABSTRACT

Federated learning has more flexible data ownership participants, therefore its data (sample feature vector or label) is more likely to be changed, and it is more vulnerable to data poisoning by malicious users, resulting in the final global model not getting the expected effect. This paper focuses on this data poisoning defense problem and applies the traditional centralized machine learning pruning optimization method to each client of federated learning. Each client needs to execute before each global iteration. Pruning optimization algorithm to remove abnormal data. The experimental results indicate that when the discrepancy between abnormal and normal samples is significant, the pruning optimization algorithm effectively eliminates the outliers, thereby minimizing their impact on the final federated learning model.

Keywords: Federated learning, FedAvg algorithm, data poisoning, outlier removal, pruning optimization

1. INTRODUCTION

Federated learning can overcome the challenge of data isolation by enhancing data utilization through collaborative modeling among data participants without the need for data sharing[1]. As shown in Figure 1, The training process in the traditional federal learning structure is typically segmented into three stages:

(1) Initialization: All end users on their local devices receive a pre-assigned, well-optimized machine learning model. The terminal may choose to participate in the learning protocol and align on the same machine learning and model training objectives.

(2) Local training: In a specific communication round, federated learning participants begin by downloading the global model parameters from the central server. They then train the model using their private training data, generate local model updates (i.e., model parameters), and transmit these updated parameters back to the central server.

(3) Model aggregation: The global model for the subsequent round is derived by aggregating all the model updates from various training samples and performing linear weighting calculations. Throughout the federated learning process, these steps are iteratively executed to optimize the current global model. The iteration process concludes when the global model parameters satisfy the convergence criteria.

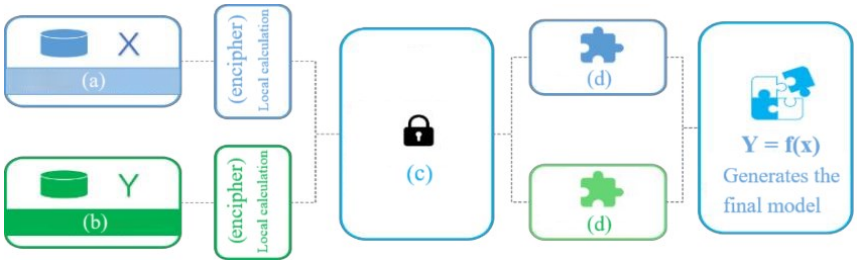


Figure 1. Federated learning system architecture. (a): The party that owns X sample; (b): The party that owns Y sample; (c): Continuously interact with intermediate calculation results under encryption protection, such as gradients, step sizes, etc.; (d): Update model parameters.

However, the application of federated learning is not smooth, and there are two kinds of problems: data security and model security. This paper mainly discusses the data poisoning problem under the model security problem in federated learning.

*zhaolei_1109@163.com

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), edited by Yang Yue, Proc. of SPIE Vol. 13395, 133952G · © 2024 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3049006

This paper addresses the issue of data poisoning during the training phase of the federated learning model and employs a pruning optimization algorithm[2] to identify outlier data points.

Specifically, before the initialization of federated learning, each client needs to conduct data statistics on the local data. When new data points are added and existing data changes, the pruning optimization algorithm is used to determine whether to retain the newly added data and the changed data. The final experimental results indicate that as the discrepancy between abnormal and normal sample increases, the pruning optimization algorithm can remove it more, which makes the impact on federated learning smaller[3].

2. RELATED WORK

In federal learning, each participant is an independent individual, and the central server lack the capability to verify whether a participant is data is normal or anomalous. As a result, if the attacker poisons the data or the model from within the federal learning, only using a small number of toxic samples, there will be more than 90% of the attack success rate, and hidden dangers can be buried in the generated model. The training values of model parameters are guided to the desired results, which reduces the sample accuracy and performance of model prediction. Data poisoning refers to the act of attackers contaminating the training dataset by introducing incorrect labels or biased data, thereby degrading the quality of data. This, in turn, compromises the final trained model, undermining its availability or integrity. Jiang et al.[4] proposed an attack method where in the attacker manipulates the parameter values of the learning model to align with their desired values while simultaneously causing the model to produce incorrect predictions for certain test samples. Chen et al.[5] adopted the hybrid auxiliary injection strategy, and a more than 90% attack success rate was achieved by injecting toxic samples into the training set. Nelson et al.[6], according to the optimization gradient produced by the support vector machine algorithm, predicted the direction of its objective function, and used the gradient ascent strategy to significantly improve the misclassification rate of SVM classifier. To enhance the scope of attacks, Biggio et al.[7] proposed a novel poisoning algorithm grounded in the concept of anti-gradient optimization. This approach is capable of targeting the gradient-based training processes across a broader spectrum of learning algorithms, including neural networks and deep learning architectures.

As adversarial attack and defense in machine learning becomes a hot topic, a large number of researches on adversarial attack and defense appear[8] It was first discovered in 2014 that by introducing calculated perturbations into the original image, a classifier that initially correctly classify the image could be induced to misclassify the perturbed image, even though the magnitude of the perturbation was imperceptible to the human eye[9]. In 2014, Goodfellow et al.[10] proposed the concept of adversative samples. Pang et al.[11] proposed understanding black-box predictions via influence functions. Adversative sample is formed by consciously adding subtle interference to the data set, whose existence will cause the model to wrongly give predictions with high confidence. For example, putting a special pair of glasses on a real person can be misidentified by facial recognition system as another person. Add some graffiti to a stop sign and it will be misread as a speed limit sign by traffic sign recognition systems. If these attack methods are used to interfere with automatic driving, face recognition and other applications, the consequences will be unimaginable.

3. PRUNING OPTIMIZATION ALGORITHM

A clean training dataset D_* with n labeled instances $< X_*, y_* >$, where $y_* \in \mathbb{R}$. The dataset was subsequently corrupted in two ways: the eigenvectors were added with noise, and the adversary added n_1 malicious instances (eigenvectors and labels) to mislead learning. Therefore, $\alpha = \frac{n_1}{n+n_1}$. $\gamma = \frac{n_1}{n}$ is defined as the corruption rate, that is, the ratio of corrupted to clean data. Now assuming the opponent knows everything about the learning algorithm. The purpose of the learner is to learn a model that is similar to the real model on corrupted data sets. Assuming that X_* of basis B is low-rank, the real model is an associated low-dimensional linear regression.

Formally, the observed training data are produced as follows.

(1) True value: $y_* = X_* w^* = U w_U^*$, where w^* is the weight vector of the real model, w_U^* represents its low-dimensional form, while $U = X_* B$ denotes the low-dimensional embedding of X_*.

(2) Noise: $X_0 = X_* + N$, where N is the noise matrix of $\|N\|_\infty \leq \varepsilon$; $y_0 = y_* + e$, where e is Gaussian noise that is independently and identically distributed, with a mean of zero and a variance of δ.

(3) Damage: In order to obtain $< X, y >$, the attacker adds n_1 data points $\{x_a, y_a\}$ of the adversarial design to design the prediction performance of low- low-dimensional linear regression to the greatest extent.

Li et al. proposed a pruning regression algorithm. In order to predict $y = X_* w + e$, we assume that $w_U = Bw$. Because of $X_* = U_* B$, the prediction problem of w in the high-dimensional space is transformed into the prediction problem of w_U in the low-dimensional space, so $y_* = Uw_U + e$. After getting the predicted value $w_U^{\wedge}$, you can convert it back to get $w_U^{\wedge} = Bw_U^{\wedge}$. It is important to observe that this closely resembles traditional principal component regression. However, in order to trick the learner into generating a wrong prediction of $w_U^{\wedge}$, and then a wrong prediction of $w^{\wedge}$, the opponent may destroy the n_1 rows in U. Pruning regression Algorithms 1 and 2 solve this problem.

Algorithm 1. Robust principal component regression

input data set *D*

$\boldsymbol{B} = \text{findBasis}(D)$

$w = \text{learnLinearRegression}(D, \boldsymbol{B})$

Algorithm 2. Pruned principal component regression

input: $\boldsymbol{X}$, $\boldsymbol{B}$, y

(1) project $\boldsymbol{X}$ to the space formed by $\boldsymbol{B}$ to get $\boldsymbol{U} \leftarrow \boldsymbol{X}\boldsymbol{B}^{\mathrm{T}}$

(2) Get $w_U^{\wedge}$ by solving the following minimization problem

$$\min_{w_U} \sum_{j=1}^{n} \{ (y_i - u_i w_U)^2, i = 1, \ldots, n + n_1 \}$$

where *z(j)* denotes the *j*-th smallest element in the ordered sequence

(3) return $w^{\wedge} \leftarrow \boldsymbol{B}\, w_U^{\wedge}$

From an intuitive perspective, the first n_1 instances reflecting the disparity between the largest observed response y_i and the predicted response $u_i w_{\mathrm{U}}$ are subtracted from the training. Here, u_i represents the *i*-th row of matrix *U*. Given that the variance of these differences is minimal (the variance of random noise $y - xw^*$ is δ), it is more probable that these instances with the largest differences are adversarial in nature.

4. PRUNING OPTIMIZED FEDERATED LEARNING ALGORITHM

The algorithm used for federated learning in this study is referenced from References[12,13]. FedAvg is atypical algorithm of the traditional federated learning optimization model. Many federated optimization algorithms and privacy protection methods for federated learning are developed based on FedAvg. Such as FedProx algorithm proposed by Li et al.[14], FEDDUALAVG algorithm proposed by Yuan et al.[15], and so on. Within the FedAvg algorithm, each participant uses the Stochastic Gradient Descent (SGD) algorithm for local training, with identical learning rates η and the same number of local iterations *E* across all participants. After each participant has performed *E* local iteration training, a parameter average aggregation calculation is performed as the initial parameter of the next local iteration training.

To enhance the robustness of federated learning, we implement Algorithms 1 and 2 on the clients involved in the federated learning process, as shown in Figure 2.

In order to delete the abnormal or malicious data of the participants, we need to perform a pruning optimization operation before each global iteration of the client in the FedAvg algorithm, and then clear the abnormal client training sample data, as shown in Algorithm 3.

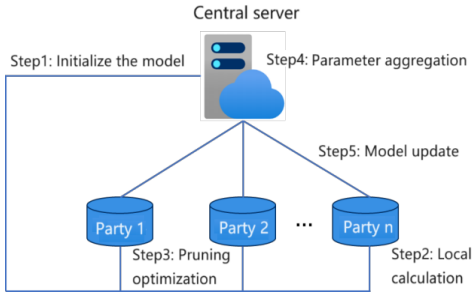


Figure 2. Federal learning process.

Algorithm 3. Federated learning FedAvg algorithm based on pruning optimization

Server:

(1) input: number of participants K, learning rate η , iteration round T, local iteration number E, model initialization parameter w_0 , total number of participants N, participant's influence on the model p_k , participant number $k = 1, ..., N$;

(2) for each iteration t ($t = 1, ..., T$);

(3) the server randomly selects K of the N participants and sends w_{t-1} to the client k;

Client:

(4) perform Algorithm 1 robust principal component regression, Algorithm 2 pruned principal component regression;

(5) receive the model parameter w^{t-1} sent by the server, the number of local iterations E, and the learning rate η;

(6) according to the learning rate η , use the SGD algorithm to perform iterative training E times, and calculate the new model parameter $w_k^{(t)}$ and send it to the server;

Server:

(7) receive model parameters from the client for aggregation operation $w^{(t)} = \sum_{k=1}^{K} p_k \, w_k^{(t-1)}$;

(8) judge whether $w^{(t)}$ meets the termination condition, if not, proceed to the next cycle;

(9) output w.

5. EXPERIMENT

5.1 Experimental setup

We conduct experimental evaluations on the FedAvg algorithm based on pruning optimized federated learning and real-world federated data sets. To facilitate a more in-depth analysis of the impact of data poisoning on federated learning, varying levels of data noise variance δ are applied across different devices.

Synthetic data: In order to generate synthetic data, we adopted the same approach as Shamir et al.[16] and Lin et al.[17]. In particular, for device k, we generated a sample (X_k, Y_k) according to model $y = \text{argmax}(\text{softmax}(Wx + b))$, $x \in \mathbb{R}^{60}$, $W \in \mathbb{R}^{10\times60}$, $b \in \mathbb{R}^{10}$. In the model we created, $W_k - N(u_k, 1)$, $u_k - N(0, \partial)$, $x_k - N(v_k, \sum)$, the diagonal elements of the covariance matrix $\sum$ are all $j^{-1.2}$. Each element of the average vector v_k comes from $N(B_k, 1)$, where $B_k \in N(0, \beta)$. Among them, ∂ is used to control the models of different devices, and β is used to control the differences in data samples of different devices.

Real data: We also used real data, as shown in Table 1. These data sets are frequently used in the research field of federated learning. We studied convex classification on MNIST. In order to add disturbances to the data, we are selecting several devices and adding some disturbances. The added disturbances are denoted by δ. We then conducted experiments on FEMNIST with 62 categories, and also selected equipment to add noise during the experiment.

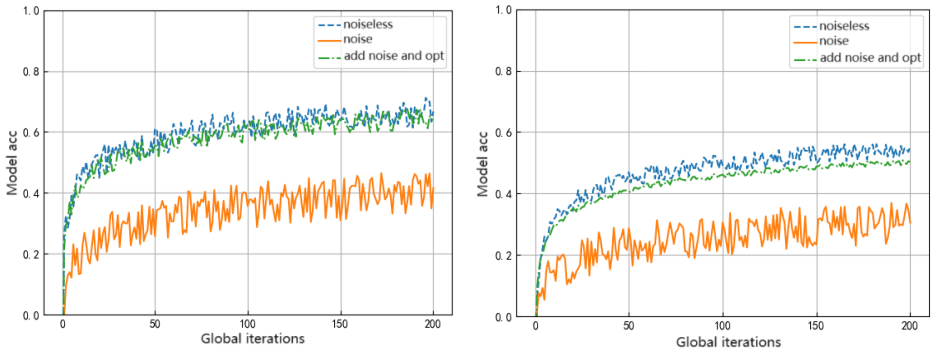
Table 1. Real federal datasets.

Datasets	Equipment	Sample
MINIST	1,000	69035
FEMNIST	200	18345

We conduct experiments on the FedAvg algorithm based on pruning optimized federated learning. We record the size of the noise added and whether the pruning optimization algorithm is added to the client to affect the FedAvg algorithm.

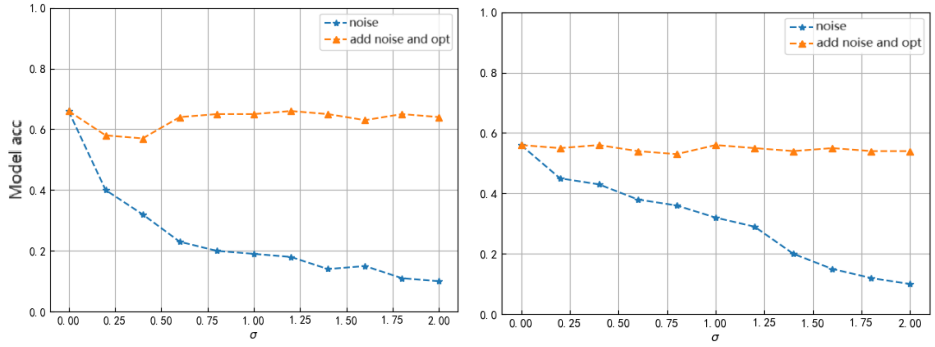
5.2 Experimental results and analysis

To compare the accuracy, 10% of the devices are selected to add the noise of δ=0.2, and compare the accuracy changes of no noise added, noise added, and noise added and the client running pruning optimization algorithm locally. As shown in Figure 3, Figure 3(a) and Figure 3(b) respectively depict the accuracy of the models trained on the MNIST and FEMNIST datasets. It is evident that as the number of global iterations increases, the impact of clients adding noise on the final model becomes more significant. In the experiment on the MNIST data set, the model accuracy will eventually be reduced to 39% without noise, and when the client performs the pruning optimization algorithm to remove some data noise, the final model accuracy will only lose about 1%; In the experiment on the FEMNIST data set, the client that adds noise has a great influence on the final model, and the final model accuracy is 50% of that without noise. When the client performs the pruning optimization algorithm to remove some noisy data, the model is accuracy is only reduced by 2% compared to the model trained without noise. The experimental results in Figure 4 show that when the data-poisoned client runs the pruning optimization algorithm, the local poisoning samples of the participating parties in the federated learning will be cleaned up, which greatly guarantees the accuracy of the federated learning model.



(a) (b)

Figure 3. Accuracy test on the MNIST and FEMNIST datasets of noise variance δ =0.2.



(a) (b)

Figure 4. The effect of adding noise on model accuracy.

Considering the impact of the newly introduced toxic sample on the original dataset, the noise variance δ is utilized in the experiment to quantify the value discrepancy between the toxic sample and the original sample. We consider the impact of the value of δ on the accuracy of the model and still choose 10% of the federal learning participants' equipment for data poisoning. Figure 4 is a statistical graph of the experimental results done on the MNIST and FEMNIST data sets. From Figures 4a and 4b, it can be reflected that as the interference of the poisoned sample on the sample increases, the accuracy of the model shows a downward trend. When the data-poisoned client executes the pruning optimization algorithm, the accuracy of the model changes little and remains basically stable.

6. CONCLUSION

Based on the basic application scenarios and potential problems of federated learning, this paper specifically analyzes the methods to deal with the data poisoning problems encountered in the actual application of federated learning. Based on the traditional centralized machine learning data poisoning defense measures, this paper proposes a method of applying the pruning optimization algorithm to the FedAvg algorithm to prevent data poisoning in federated learning. This method is mainly used to remove abnormal sample points in federated learning so that federated learning can train the model on a clean data set. The experimental results indicate that when the noise is pronounced, the pruning optimization algorithm can quickly clean up the abnormal data so that the final model will not have too much influence on the prediction accuracy. However, the risks and challenges faced in the actual application field are often more complicated. If a user participating in federated learning knows the details of the model, he may design a class of samples with minimal noise based on the characteristics of the model. It is difficult to be detected by the pruning optimization algorithm, but it has a great destructive effect on federated learning. The poisoning designed by malicious users for the detailed information of the model is one of the challenges in future federated learning.

ACKNOWLEDGEMENT

This paper was supported by the project No.: JS202011.

REFERENCES

[1] Li, M., Wang, W. and Zhou, Z. H., "Exploiting remote learners in Internet environment with agents," Science in China Series F: Information Sciences, 53(1), 64-76 (2010).
[2] Steinhardt, J., Koh, P. W. and Liang, P., "Certified defenses for data poisoning attacks," Proceedings of the 31st International Conference on Neural Information Processing Systems, 3520-3532 (2017).
[3] Yang, A., Ma, Z., Zhang, C., et al., "Review on application progress of federated learning model and security hazard protection," Digital Communications and Networks, 9(1), 146-158 (2023).
[4] Jiang, W., Li, H., Liu, S., et al., "A flexible poisoning attack against machine learning," ICC 2019-2019 IEEE International Conference on Communications (ICC), 1-6 (2019).
[5] Chen, X., Liu, C., Li, B., et al., "Targeted backdoor attacks on deep learning systems using data poisoning," arXiv preprint arXiv: 1712.05526, (2017).
[6] Biggio, B., Nelson, B. and Laskov, P., "Poisoning attacks against support vector machines," arXiv preprint arXiv: 1206.6389, (2012).
[7] Muñoz-González, L., Biggio, B., et al., "Towards poisoning of deep learning algorithms with back-gradient optimization," Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 27-38 (2017).
[8] Szegedy, C., Zaremba, W., Sutskever, I., et al., "Intriguing properties of neural networks," arXiv preprint arXiv: 1312.6199, (2013).
[9] Wang, Z., Toussaint, P. J., Evans, A., et al., "Exploring the brain characteristics of structure-informed functional connectivity through graph attention network," bioRxiv, (2023).
[10] Goodfellow, I. J., Shlens, J. and Szegedy, C., "Explaining and harnessing adversarial examples," arXiv preprint arXiv: 1412.6572, (2014).
[11] Koh, P. W. and Liang, P., "Understanding black-box predictions via influence functions," International Conference on Machine Learning, 1885-1894 (2017).
[12] Konečný, J., "Stochastic, distributed and federated optimization for machine learning," arXiv preprint arXiv: 1707.01155, (2017).

[13] Li, D., Guo, Y., Liu, D., et al., "Client-edge-cloud hierarchical federated learning based on generative adversarial networks," 2023 IEEE International Conference on Knowledge Graph (ICKG), 160-167 (2023).
[14] Li, T., Sahu, A. K., Zaheer, M., et al., "Federated optimization in heterogeneous networks," arXiv preprint arXiv: 1812.06127, (2018).
[15] Yuan, H., Zaheer, M. and Reddi, S., "Federated composite optimization," International Conference on Machine Learning, 12253-12266 (2021).
[16] Shamir, O., Srebro, N. and Zhang, T., "Communication-efficient distributed optimization using an approximate newton-type method," International Conference on Machine Learning, 1000-1008 (2014).
[17] Lin, S., Yang, G. and Zhang J., "A collaborative learning framework via federated meta-learning," 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), 289-299 (2020).