Model Poisoning Attack In Federated Learning Via Adversarial Examples

Liang Yuan
School of software
Yunnan University
Kunming, China
yuan_2019go@163.com

Huajie Hu
School of software
Yunnan University
Kunming, China
huhuajie404@email.ynu.edu.cn

***Abstract*-The emergence of federated learning framework sets protects user privacy and has solved the Isolated Data Island problem. However, existing federation attack methods mostly use label flipping for data poisoning attacks, while federated backdoor attacks require patching the target samples. The attacker can also arrange other generative networks at local nodes but has limited data reconstruction capability. Thus, this paper proposes a new poisoning attack method named adversarial example poisoning attack (AEPA). The attacker uses the distributed global model to create adversarial examples and uses the original clean label for federation training and attacks against the specific target class in the dataset. In the evaluation process, we conducted extensive experiments on the CIFAR-10 dataset. We also explored the toxicity of the adversarial examples under different generation methods, compared our method with the label-flipping attack, and finally showed the effectiveness of AEPA.**

Keywords-Federated learning, Poisoning attack, Adversarial examples

I. INTRODUCTION

Federated learning is an emerging privacy-preserving framework in which many participants can collaboratively train a joint global model without sharing and uploading individual datasets. Several current practical applications of FL include visual object detection [1], training with fragmented healthcare data sources with privacy-preservation within the biomedical space[2], and next-word prediction during search [3].

There are two roles named global server and local participants in federated learning, and a typical training process is as follows: first, the server initializes the model parameters and distributes them to each participant; then, each participant uses the distributed parameters and its local dataset to train the model and uploads the local updates; next, the global server uses the aggregation algorithm, such as the federated averaging algorithm [4]. The above process is repeated until the global server converges.

Although many federated attack methods have emerged in recent years, they still have shortcomings. Compared to data reconstruction methods using generative networks [5], AEPA can generate adversarial samples of RGB datasets and does not require additional networks. Compared to federated backdoor attacks [6], AEPA targets clean samples of benign participants belonging to the attacker-chosen class, which have a higher probability of occurrence. The poisoning persistence of AEPA is higher compared to label flipping.

Accordingly, we briefly summarize our attack method in Figure 1 At first, in each communication round, the global server distributes model G_t to all selected local participants. Then each participant trains it on its local datasets to generate the new model L_i^{t+1}. Still, the attacker uses clean local examples belonging to the attacker-chosen class and G_t to generate adversarial examples, which will be mixed with originals from other classes and submits the poisoned model L_m^{t+1}. Finally, the global model G_{t+1} will be replaced by the scaled model L_{m}^{t+1} after federated averaging.

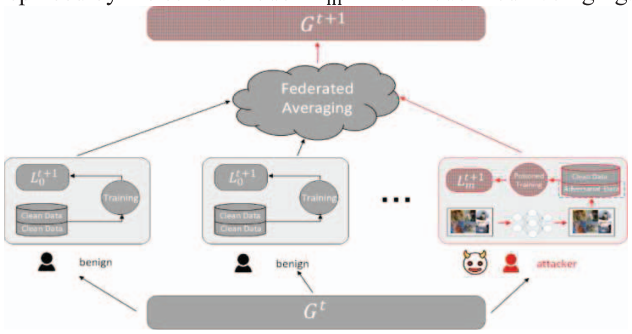


Figure 1. An illustrating of AEPA in federated learning.

In this paper, we design a model poisoning attack method based on adversarial examples, called adversarial example poisoning attack (AEPA). The attacker expects the attacked global model to misclassify clean samples of the target class with high confidence while correctly classifying samples of other classes. In conclusion, our main contribution is three-fold.

1) We propose a new adversarial example-based federated attack and apply the adversarial examples as poisoned datasets in the federation training phase, with more substantial poisoning persistence than label flipping.

2) The attacker can manufacture adversarial examples using the global model sent down when convergence is achieved. We also explore the neutrality of the adversarial examples under different generation methods.

3) We conducted extensive experiments on the Cifar-10 and achieved over 80% accuracy on the main task and around 98% on the target task, and our attack lasted for certain epochs.

II. RELATED WORK

A. Federated Learning

Federated learning is becoming increasingly popular due to its privacy-preserving features. Federated Averaging (Fed-Avg) was the very first FL algorithm where updates uploaded by all selected participants were aggregated via a weighted averaging. There were two types of poisoning attacks in federated learning: data poisoning and model poisoning. In data poisoning, the attacker could arbitrarily manipulate a subset of the local training set to generate poisoned updates [7], but could not modify the local learning procedure[8]. In model poisoning, the attacker could directly modify the local updates, such as backdoor attack with embedded triggers and poisoning attack using GAN. Our attack falls under the category of model poisoning attacks.

B. Attacks Adversarial Examples

A common idea for creating adversarial examples is adding a tiny amount of well-tuned additive perturbation on the originals, which will cause a pre-trained classifier to misclassify it as an irrelevant class. Intuitively, although the adversarial examples and originals look extremely similar in the eye of humans, they look significantly different to neural networks. There are plenty of ways to generate adversarial examples, including white-box methods where the attacker has the knowledge of the structure, such as FGSM[9] algorithm with a single iteration along the gradient direction of the loss function, IFSM[10] algorithm with multiple iterations and PGD algorithm[11] with iterations from random points, and parameters of a given model and black-box methods that emphasize transferability of the adversarial examples.

III. METHOD

In this section, we describe the threat model, introduce the attacker optimization objective when generating adversarial examples, and finally introduce our unconstrained poisoning attack and constrained poisoning attack.

A. Threat Model

The attacker's goals can be summarized as follows. The attacker can create high-quality adversarial examples using the distributed global model and local dataset. The attacker wants the poisoned global model that misclassifies clean samples of the target class with high confidence while correctly classifying examples of other classes.

We consider a realistic attacker with the following capabilities. The attacker has full knowledge of the model structure, can collaborate and share the local dataset, and controls the local training procedure. However, the attacker does not access and control any aspect of the central server and benign nodes.

B. Generation of target poisoning examples

The adversarial examples can be used as poisoned training sets because when adversarial perturbations are added to the correct direction of the originals, their depth characteristics change significantly. The gradient-based optimizer can effectively find these directions and greatly impact the model parameters during the training phase. Inspired by [12], the optimization objective of the attacker in generating target poisoning sample is as follows:

$$\min_{\delta \in S} \left[\sum_{i \in D_{target}} \mathcal{L}\big(L_{mal}(x_i^*; w^*), g(y)\big) \right], x_i^* = x_i + \delta_i \quad (1)$$

The above equation optimizes the perturbations δ_i added to the originals. Here, S denotes the constraint set of the perturbations. We craft our poisons with 250 steps of PGD on this loss-maximization objective. Our attacks are bounded by ℓ_∞ with $\varepsilon = 8 / 255$ on CIFAR-10. We update the attacker model L_{mal} using the distributed global model G^t. D_{target} denotes the clean local examples belonging to the target class, and x_i^* denotes the target adversarial examples obtained by adding the tiny perturbations δ_i on the original x_i. w^* represents the distributed global model parameter, whose parameters are fixed during the generation period, and we call L_{mal} the crafting model. $g(\cdot)$ denotes a permutation on the label space.

C. Adversarial example poisoning attack

Another critical point for constructing AEPA is that the attacker substitutes the target adversarial examples for the clean local examples belonging to target class. Then the poisoning data will be mixed with the clean examples belonging to other classes to form the new dataset D_i, which will be injected into the federated training process with their original labels y_i. The optimization objective of the attacker in training phase is as follows:

$$w^* = \arg\max_{w}\left(\textstyle\sum_{i \in D_{adv}} \mathbb{1}(L_{mal}(x_i^*) = y_i; \gamma) + \sum_{i \in D_{cln}} \mathbb{1}(L_{mal}(x_i) = y_i; \gamma)\right) \quad (2)$$

The above equation maximizes the attacker's local model parameter w. We call this AEPA. The attacker can choose an optimal scale factor γ to result in better model parameters w, with which the attacked global model G^{t+1} is able to misclassify clean examples of the target class with high confidence and correctly classify samples from other classes in the testing phase. Here, D_{adv} is the generated adversarial sample of the target category and D_{cln} is the clean sample of the other categories. Both satisfy $D_{adv} \cap D_{cln} = \emptyset$ and $D_{adv} \cup D_{cln} = D_i$. The attacker chooses to perform a single attack when the global model tends to converge in order to reduce the impact of benign updates. $\mathbb{1}(\cdot)$ indicates the indicator function.

IV. EXPERIMENTS

A. Introduction to the experimental environment

We use CIFAR-10 [13] image classification datasets to conduct our attacks. For the experiments on CIFAR-10, we use ResNet18 [14] as a convolutional neural network.

Our experiments used the PyTorch framework. Specifically, benign participants trained two local epochs. The attacker performed the same training procedure in the non-attack round but trained 20 local epochs in the attack round. There were 50 nodes in the experiment, 5 nodes were randomly selected to participate in the federation training, and 5 nodes were selected as the malicious ones. The scaled factor is 30, and the attacker attacked when the global model converged at 1500 rounds. In generating adversarial examples, our attacks are constrained by the ℓ_∞ norm with $\epsilon = 8 / 255$. In addition, we use a Dirichlet distribution[15] with hyperparameter 0.9 to partition the training set, repeat each experiment five times, and report the average results.

The target task accuracy represents the global model's misclassification success rate for the target class, while the main task accuracy represents the correct rate for the whole test set. The following experiments show the change of the global model accuracy after a single attack.

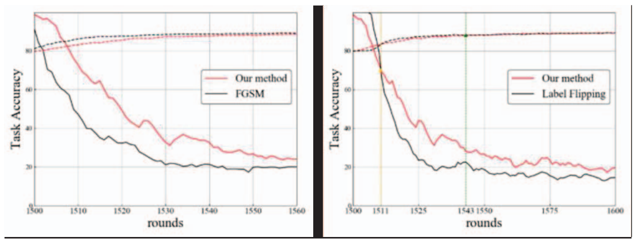
B. Experiment results

We compare our AEPA against these methods:

1) FGSM: the attacker uses only the FGSM algorithm to generate target class adversarial examples. In addition, FGSM and AEPA use $\varepsilon = 8 / 255$ for the same noise constraint.

2) Label Flipping: the attacker only labels class 5 as class 3 during federation training and uses the misclassification of clean test samples belonging to class 5 as the target task.

Figure 2 shows the accuracy changes of the main and the target task after at 1500th round, where the dashed line represents the main task and the solid line represents the target task. Experiment shows that AEPA produces much igher toxicity in the adversarial examples than FGSM. Although FGSM adds perturbations based on gradients, it only iterates once, so the toxicity is lower than AEPA which employs multiple iteration. Comparison with Label Flipping sh ws that the target accuracy of AEPA decreases more sl wly because the adversarial perturbation of AEPA is added along the gradient, so it has more stronger poisoning persistence.



(1) Comparison with FGSM
(2) Comparison with Label Flipping

Figure 2. Comparison experiments.

We explore the extent to which different times of generating the target poisoning examples affect the global model. The attacker creates adversarial samples at different times but performs federated training in the 1500th round to pload poisoning updates. Figure 3 shows the change in global model accuracy for the following 100 rounds after the poisoning attack. The results show that as the global model converges, the accuracy and persistence of the target task increase. We analyze that well-trained models can add more effective adversarial perturbations.

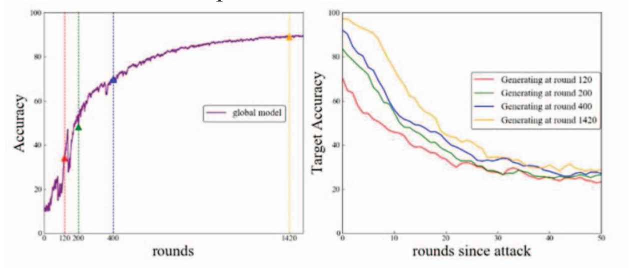


Figure 3. Effects of generating time.

V. CONCLUSION

This work proposes the Adversarial Example Poisoning Attack (AEPA), a practical and effective attack algorithm. The attacker makes adversarial examples of the target class and joins it in the federation poisoning training process. We conduct extensive experiments on CIFAR-10 to demonstrate the effectiveness of our AEPA. The experiments show our approach achieves more substantial poisoning persistence than the label-flipping method.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62162067 and 62101480 in part by the Yunnan Province Science Foundation for Youths under Grant No.202005AC160007 and the fundamental research of Yunnan Province under Grant No.202001BB050076. and Open Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant No. 2020SE310.

REFERENCES

[1] Liu Y, Huang A, Luo Y, et al. Fedvision: An online visual object detection platform powered by federated learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(08): 13172-13179.

[2] Xu J, Glicksberg B S, Su C, et al. Federated learning for healthcare informatics[J]. Journal of Healthcare Informatics Research, 2021, 5: 1-19.

[3] Yang T, Andrew G, Eichner H, et al. Applied federated learning: Improving google keyboard query suggestions[J]. arXiv preprint arXiv:1812.02903, 2018.

[4] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

[5] Zhang J, Chen J, Wu D, et al. Poisoning attack in federated learning using generative adversarial nets[C]//2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And

Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, 2019: 374-380.

[6] Xie C, Huang K, Chen P Y, et al. Dba: Distributed backdoor attacks against federated learning[C]//International conference on learning representations. 2020.

[7] Tolpegin V, Truex S, Gursoy M E, et al. Data poisoning attacks against federated learning systems[C]//Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25. Springer International Publishing, 2020: 480-501.

[8] Sun G, Cong Y, Dong J, et al. Data poisoning attacks on federated machine learning[J]. IEEE Internet of Things Journal, 2021, 9(13): 11365-11375.

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

[10] Kurakin A, Goodfellow I J, Bengio S. Adversarial examples in the physical world[M]//Artificial intelligence safety and security. Chapman and Hall/CRC, 2018: 99-112.

[11] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks[J]. arXiv preprint arXiv:1706.06083, 2017.

[12] Fowl L, Goldblum M, Chiang P, et al. Adversarial examples make strong poisons[J]. arXiv preprint arXiv:2106.10807, 2021.

[13] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. 2009.

[14] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[15] Minka T. Estimating a Dirichlet distribution[J]. 2000.