ECE 661 Final Project Report
Poisoning Attacks to Federated Learning

Jingyu Peng
MEMS
Duke University
jy.peng@duke.edu

Libo Zhang
ECE
Duke University
libo.zhang@duke.edu

Shucheng Zhang
MEMS
Duke University
shucheng.zhang@duke.edu

Abstract

In federated learning, model training is divided into several client devices. Each client device has local model with its own data. After the local devices finish training, the center device collects model parameters from clients and build global model with aggregation techniques. However, standard federated learning can be easily attacked by attackers. They can attack the local training datasets and local model parameters so that the global model will get wrong parameters from the local devices and there will be huge drop of testing accuracy. In our project, we aim at implementing federated learning and implementing two kinds of poisoning attack: data poisoning attack and local model poisoning attack, to test the robustness of federated learning. Additionally, we discuss the defense methods against the two kinds of attack.

1 Background

Federated learning builds machine learning models where training is conducted at the edge, and there is no need to collect all users' data to centralized model[1]. Without sharing data, federated learning has the advantage of alleviating complicated legal issues including data privacy, data access rights and data security. Several federated learning models have been proposed so far trying to achieve good accuracy and efficiency. The centralized federated learning arranges one central server to manage all clients during model training as well as to control the algorithmic implementations step by step, while the decentralized federated learning arranges local clients to coordinate with the purpose of acquiring the global model[2].

However, research conducted on how to improve the reliability of federated learning under poisoning attacks is rare[3]. Currently, there are two major types of poisoning attacks, which are data poisoning attacks and local model poisoning attacks[4]. For data poisoning attacks, before model training, the attacker would inject poisoned data samples into some local clients and turning them into malicious participants, therefore distorting the federated learning process. Typical data poisoning attacks include targeted label flipping attack and backdoor attack, both methods enable attackers to manipulate the class labels of the original training set[1, 5]. Another very strong attacking method is the local model poisoning attack, where the attacker directly distorts local clients' model parameters during federated learning[4]. The distorted local clients then became malicious participants and their poisoned parameters will be sent to the global model for federated averaging, therefore jeopardizing the global model's performance.

Both data poisoning attacks and local model poisoning attacks can distort federated learning heavily and impose negative effects on the global model's testing accuracy, therefore it is very important to thoroughly study poisoning attacks against federated learning, carefully evaluate experimental attacking results, and correspondingly propose potential poisoning attack detecting and defending methods.

ECE 661 (2021 Fall) final report.

2 Method

Datasets and Model. Our projects were implemented on the Cifar-10 dataset and all local models are classic CNN / ResNet20 models. The CIFAR-10 originally contains 60000 images from 10 categories, where each example is 32×32 RGB image. The architecture of classic CNN is shown in Table 1.

Table 1: CNN architecture

Layer type	Size
Convolution + ReLU	$3 \times 6 \times 5$
Max pooling	2×2
Convolution + ReLU	$6 \times 16 \times 5$
Max pooling	2×2
Fully connected + ReLU	400×120
Fully connected + ReLU	120×84
Fully connected + Softmax	84×10

2.1 Federated Learning

Federated Learning enables many devices to train a machine learning model jointly. One of the common algorithm on independently and identically distributed data (IID data) is FedAvg[6]. However, local data in different devices are usually non-independently and identically distributed (non-IID) and non-IID data can influence the accuracy of FedAvg a lot[7]. There have been some studies[8] trying to address the drift issue in Federated Learning. Here we mainly discuss the algorithm FedProx[8] because it is a popular approach, and it can improve the performance of Federated Learning on non-IID data.

Algorithm 1 Federated Learning Algorithm, including FedAvg/FedProx, and four types of aggregation methods: Mean(FedAvg), Median, Trimmedmean, Krum. The K is numer of clients; E is the number of epoch; η is the learning rate

Server executes:
initialize w_0
for each communication round $t = 1, 2, ...$**do**
choose random set of clients S_t
for each client $s \in S_t$ **do**
$w_{t+1}^s \leftarrow$ LocalUpdate(s, w_t)
$w_{t+1} \leftarrow$ Aggregation of w_{t+1}^s
LocalUpdate(s, w)**:**
For FedAvg: $\ell(w; b)$
For FedProx: $\ell(w; b) + \frac{\mu}{2}\|w - w_t\|^2$
for each batch b **do**:
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

2.2 Data Poisoning Attacks

We use targeted label flipping attacks as the data poisoning attack method. To ensure good accuracy, we use ResNet20 for both the global network and the local client networks. We choose the number of local clients as 100 and we utilize all 100 participants with independent and identically distributed training data. To implement data poisoning attack, we first set a malicious participant percentage to turn some of the local clients into malicious participants, which means they will have poisoned data during federated learning. For malicious participants' training data, we change all source class's labels into target class's labels. In experiments, we have 6 percentage values [0%, 20%, 40%, 60%, 80%, 100%], where 0% represents the standard, non-poisoned ResNet20 federated learning model and 100% means all clients are attacked. We set the source class as "airplane (label 0)" and the target class as "bird (label 2)". We evaluate the global net testing accuracy as well as class recall (accuracy to classify each class) when analyzing the experimental results.

2.3 Model Poisoning Attacks

Threat model Like all attack methods, our attack goal is to manipulate the training process to decrease the testing accuracy rate. To achieve this, we assume the attacker has control c worker devices and can craft their model parameters. To ensure it is not easy for the attacker to control the whole global model, we say c is less than 50%. Finally, we suppose the attack knows the aggregation rules of the learning process and be fully knowledgeable of the training data. This means the attacker knows the local training dataset and local model on every worker device, so it is the upper bound of the attacks' threats for a given setting of federated learning.

Attack method
(1) Attack trimmed mean. We assume that $w_{max,j}$ and $w_{min,j}$ are the maximum and minimum of the jth local model parameters on the benign worker devices.We also define s_j as the changing direction of the jth global model parameters compared with the last iteration. Specifically, s_j=1 means the parameters of jth global model increase upon the previous iteration, while $s_j = -1$ means the decreasing of global model parameters. Theoretically, the basic attack strategy is that if $s_j = 1$, we will craft c numbers of compromised worker devices and make their value smaller than $w_{min,j}$, otherwise, if $s_j = -1$, the value of those devices will be larger than $w_{max,j}$. To be detailed, the crafting local model parameters will be chosen from the following interval.
if $s_j = 1 : [w_{min,j}/b, w_{min,j}]$, $w_{min,j} > 0$ $[bw_{min,j}, w_{min,j}]$, $w_{min,j} \leq 0$
if $s_j = -1 : [w_{max,j}, bw_{max,j}]$, $w_{max,j} > 0$ $[w_{max,j}, w_{max,j}/b]$, $w_{max,j} \leq 0$
(2) Attack median. We use the same attacks for trimmed mean to attack the median aggregation rule, because they show similarities in the experiment.
(3) Attacking Krum and Bulyan. The attack to Krum and Bulyan[4] can be transformed into the optimization problem:
$\max \lambda$ subject to $w'_1 = Krum(w'_1, ...w'_c, w_{c+1}, ..., w_m)$ and $w'_1| = w_{Re} - \lambda s$
After solving λ in the problem, we can get the crafted local model w'_1. Then, the remaining $c-1$ compromising work devices will be crafted as the same as w'_1 to support w'_1 to be chosen as the global model with Krum's rule. The λ can be got from the following function. With the upper bound, we use a binary search to find λ.

3 Experiment results

3.1 Federated Learning

Experiment Setup. In our experiment, we use CNN and ResNet20 as our model and we use CIFAR-10 as our dataset. In exploring the effectiveness of federated learning, we experimented with the influence of different hyperparameters on the effect of federated learning and the performance of federated learning on IID data and non-IID data.

Impact Factors of Federated Learning. We evaluate the ratio of chosen clients C and the batch size of training dataset B. We have a total of 100 clients. In the experiment, we test the $C = 0.1$ and $C = 0.5$, the $B = 50$ and $B = 128$.

In our experiment, we find that ratio $C = 0.5$ can improve the global model's performance and batch size $B = 50$ can improve the global model's performance a lot, the result is shown in Figure 1. Then we come to know that more clients' participation can bring better performance and smaller batch size of data can improve the model's performance. It is strategy when trying to get better performance of Federated Learning.
We then conduct experiment on non-IID data. We evaluate the influence of local update iteration. Here we allocate for each local model with dataset that only contains two classes. The iteration n varies :$n = 2, 3, 5, 10$. The result is attached in the Figure 6 in appendix.
We then evaluate the influence of the non-IID data. We try to find the relationship between the degree of non-IID and the performance. Here we change the data. For each device, we allocate dataset which contains $1, 2, 3$ classes. The result is attached in the Figure 7 in appendix.

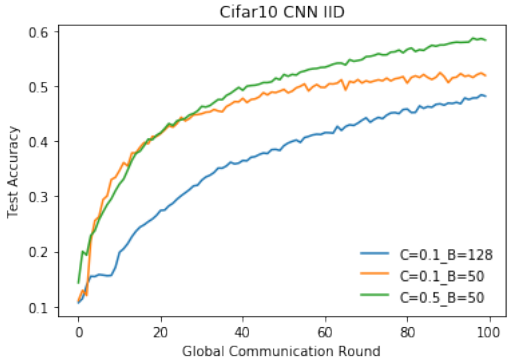

Figure 1: Cifar10 CNN IID

3.2 Data Poisoning Attacks

In this section, the first three parts should be developed with clear and strong evidence of experimental results, while the last one involves general discussion.

Federated learning is vulnerable to data poisoning attacks. Data poisoning attacks affect the global model testing accuracy negatively, meaning that increasing the percentage of malicious participants will decrease the global model testing accuracy. In specific, we can also find that if the percentage ranges from 40% to 60%, the federated learning process is very noisy with accuracy fluctuating drastically. It seems that the global model is struggling to tell which data sample is correct (non-poisoned) and which is incorrect (poisoned). When the malicious percentage goes above 80%, we can observe that the training process is no longer noisy, but the testing accuracy significantly decreases about 9%, which means the global model almost completely fails to recognize the source class "airplane". The result is shown in Figure 2

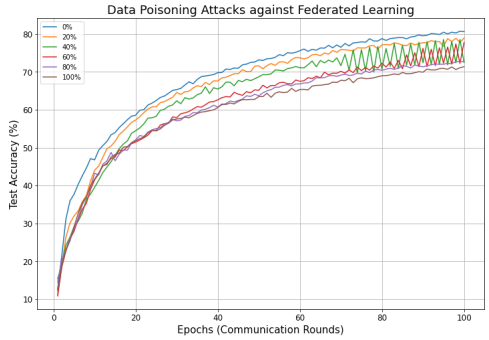

Figure 2: Data Poisoning Attacks against Federated Learning

Data poisoning attacks can be targeted in supervised federated learning. Targeted label flipping attacks can significantly jeopardize the source class recall, while the target class recall and all other classes' average recall are not heavily influenced. We can also confirm that when malicious percentage goes above 80%, the global model completely fails to recognize the source class "airplane". The result is shown in Figure 3.

Data poisoning attack timing affects federated learning model vulnerability Attackers must inject poisoned data to turn some local clients into malicious participants during the last dozens of federated learning epochs (communication rounds). If the attacks only occur during the early period of training, we can observe that the global model's source class recall can recover in a very short period and can still reach relatively good accuracy compared with non-poisoned model. The result is shown in Figure 4.

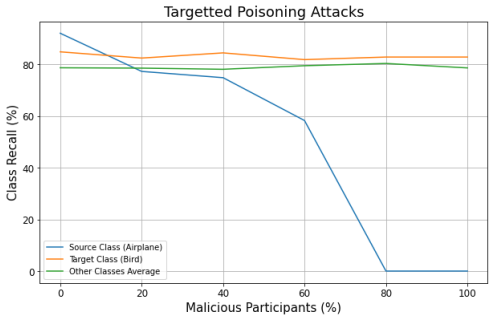

Figure 3: Targeted Poisoning Attacks

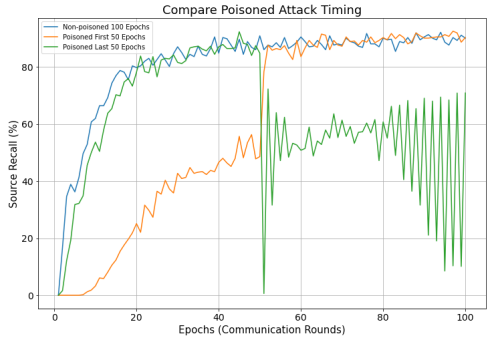

Figure 4: Poisoning Attack Timing Comparison

Potential data poisoning attack detection and defense methods To detect, we may compare local clients' model gradients[1] during federated training process. For each class, we can extract the gradients between the last hidden layer and the output layer in specific to this class node. After removing the mean and scaling to unit variance, we use principal component analysis to reduce the gradients' dimension to 2 for better visualization. If a small group of projected gradients is clearly deviating from the large group, then the participants having such gradients could be malicious. To defend, we could use clustering or regression methods to exclude these malicious gradients when averaging weights to the global model, and the class recalls could recover quickly according to our observation and analysis for the last part.

3.3 Model Poisoning Attacks

key Parameters. In the before experiments, we find decreasing the number of work devices can improve the test accuracy, so to promise the prediction accuracy and effect of attacking, we choose 20 work devices and each of them will be used in each epoch. In our experiments, we changed c, Iter and IID (Non-IID) respectively to study their influence on different attacks.

Our attacks are effective. The image here shows the test accuracy rate of four aggregation rules under different degree attacks. For mean aggregation, we deploy a trimmed-mean-attack on it, because both mean and trimmed mean calculate the average in the last step which means this attack method will be effective on them. From the results, the traditional federated learning with mean aggregation rule performs the worst and even becomes unusable and unavailable, which proves to mean is not robust under adversarial settings. Also, our attacks result in higher error rates in other situations. The result is shown in Figure 5.

Impact of the percentage of attacked worker devices. We use 4, 5, 6, and 7 attacked local models which represent 20%, 25%, 30%, and 35% model attack rate respectively on the three advanced aggregation rules. The result shows our attack increases the error rates significantly as we compromise more work devices. Specifically, in the trimmed mean and the median situations, when we attack 35% local model parameters, the error rate can increase 30% at most, while in the Krum situation, the increase is not obvious (It just achieve around 10% under 8 attacked model).

However, our attack is not as effective as paper[4], though we use the same methods. The reason might be our model is much easier than the model used in paper[4] and we do not use some delicate methods to increase the accuracy rate which may be non-robust under attacking. Our datasets are not as complicated as the paper[4] might be another reason because this will cause the model to be harder to attack. The experiments between different datasets in paper[4] prove this and show the dataset's dimension is another crucial factor of the attack effect.

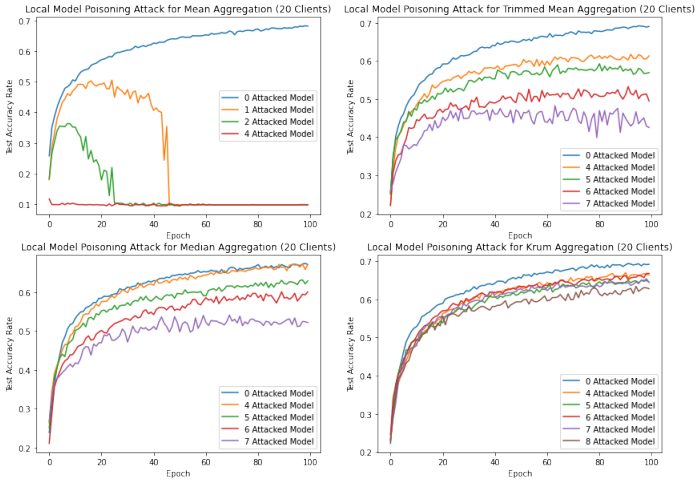

Figure 5: Poisoning Attack Timing Comparison

4 Conclusions

1. We find that larger ratios can improve the global model's performance and smaller batch size can improve the global model's performance a lot. We also find local updating three times in one round brings the best performance, while a small or large number brings worse performance. It shows that appropriate choice of local update is necessary. We find training on IID data brings better performance compared with training on non-IID data and different extend of non-IID data brings different performance. The highly extend of non-IID will bring bad performance.
2. Data poisoning attacks affect the global model testing accuracy negatively. Increasing the percentage of malicious participants will decrease the global model testing accuracy. Targeted label flipping attacks can significantly jeopardize the source class recall, while the target class recall and all other classes' average recall are not heavily influenced. Attackers must inject poisoned data to turn some local clients into malicious participants during the last dozens of federated learning epochs.
3. We prove local model poisoning attacks can easily influence the performance of federated learning. To increase the error rates of the learned model, the attacker can compromise more worker devices. On the other hand, using datasets with a less non-IID degree and reducing iterations on local models can make the global more robust. For defense strategy, the byzantine-robust federated learning model performs well under the same degree of attack compared with traditional learning, while ERR and LFR generalized from data poisoning attack only have limited success in this situation.
4. To detect, we may compare local clients' model gradients[1] during federated training process. To defend data poisoning, we could use clustering methods to exclude these malicious gradients when averaging weights to the global model.
5. To defend model poisoning attack, there are two ways:
1) Byzantine-Robust Aggregation Method[4]: As we mentioned before, it is less likely for an attacker to influence the global model significantly by manipulating only a few local models due to the more advanced aggregation methods (Trimmed mean, median, and Krum).
2) Error Rate based Rejection (ERR)[4]: This method is based on RONI. It will remove the local models which have a large negative impact on the error rate of the global model. Loss Function based Rejection (LFR)[4]: Similarly, this method which is inspired by TRIM will remove the local model which results in a large loss.

References

[1] Tolpegin, V., Truex, S., Gursoy, M. E., & Liu, L. (2020, September). Data poisoning attacks against federated learning systems. In European Symposium on Research in Computer Security (pp. 480-501). Springer, Cham.

[2] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... Zhao, S. (2019). Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977.

[3] Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L., Liu, J. (2021). Data poisoning attacks on federated machine learning. IEEE Internet of Things Journal.

[4] Fang, M., Cao, X., Jia, J., & Gong, N. (2020). Local Model Poisoning Attacks to Byzantine-robust Federated Learning. In 29th {USENIX} Security Symposium ({USENIX}Security 20) (pp. 1605-1622).

[5] Li, Y., Wu, B., Jiang, Y., Li, Z., & Xia, S. T. (2020). Backdoor learning: A survey. arXiv preprint arXiv:2007.08745.

[6] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[7] Li, Q., Diao, Y., Chen, Q., & He, B. (2021). Federated learning on non-iid data silos: An experimental study. arXiv preprint arXiv:2102.02079.

[8] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems, 2, 429-450.

A Timeline and task allocation

Table 2: Timeline and task allocation

	Part	
Name	Description	Time
Jingyu Peng	FL and impact factors	Nov 4 - Nov 23
Libo Zhang	Data poisoning attack	Nov 20 - Dec 6
Shucheng Zhang	Model poisoning attack	Nov 20 - Dec 6

All team members discuss the defense of poisoning attacks from Dec 1 - Dec 6 together.

B Implementation detail

CNN model architecture is shown below:

Table 3: CNN architecture

Layer type	Size
Convolution + ReLU	$3 \times 6 \times 5$
Max pooling	2×2
Convolution + ReLU	$6 \times 16 \times 5$
Max pooling	2×2
Fully connected + ReLU	400×120
Fully connected + ReLU	120×84
Fully connected + Softmax	84×10

C Additional experiment results

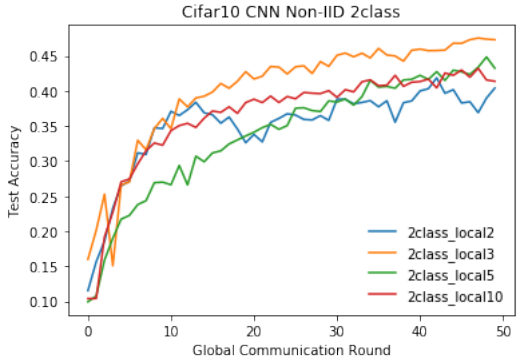

Figure 6: Cifar10 CNN Non-IID 2class

The performance of Federated Learning is bad when the local update iteration is too large or too small. Instead, an appropriate choice of local update iteration, like three, performs better.

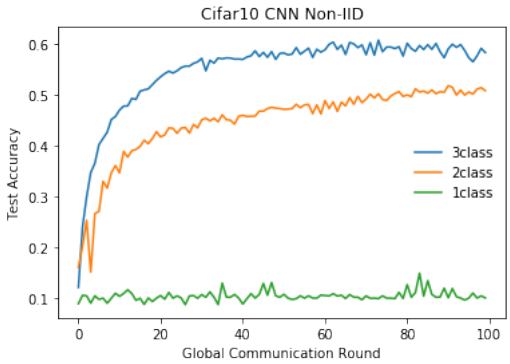

Figure 7: Cifar10 CNN Non-IID

The more unbalanced non-IID (higher degree non-IID) the data is, the worse performance the Federated Learning has. When it is the one class situation, the accuracy is around "random guess", but the accuracy rise to 60% when it is the three class situation.

Impact of using IID and non-IID datasets. The training data for federated learning is more possible to be non-IID because its learning is not centralized. To study the training data type's influence on the attack, we used a 20% attack rate on trimmed mean and the models are trained by IID or non-IID data. The result shows training with non-IID data is much easier to be attacked. One possible reason is that the local model could be more diverse by using non-IID data, which leaves more room for attacking.

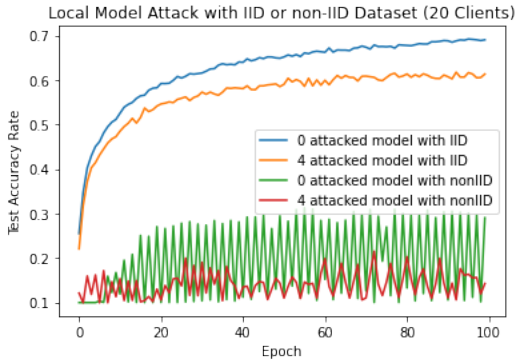

Figure 8: Model Poisoning Attack with IID or non-IId Dataset

Impact of local model training iteration. We also study the local model iteration's influence on the attack effect. The image shows decreasing iteration could weaken the attack to some degree, though it also decreases the accuracy rate. The reason might be that fewer iterations may reduce its deviation in the wrong direction.

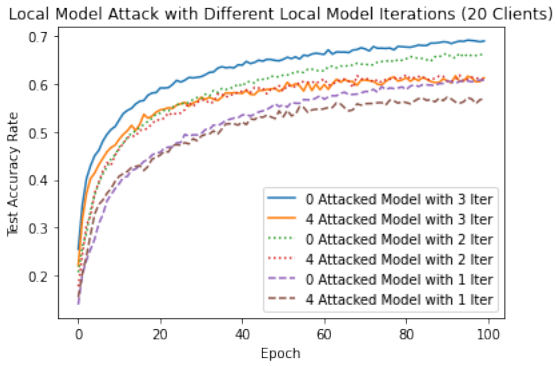

Figure 9: Model Poisoning Attack with Different Local Model Iterations