CS6203 Project
Distributed Backdoor Attacks Against Federated Learning
-attack and defend

Fu Yujian
A0206942E
e0427770@u.nus.edu

Ye Qiyuan
A0209641H
e0452951@u.nus.edu

I. Introduction

Backdoor attacks aim to change the sample data through the attack algorithms so that the machine learning model will make an incorrect prediction on the targeted sample data. In today's society, much more data stored in different organizations, to make better use of those different organizations' data, people prefer to use some up to date machine learning technology, such as federated learning, federated learning can combine different organizations' data to train a better model under the condition of not worrying about data privacy, but it will also bring new vulnerabilities. Besides, facing federated learning, the backdoor attack's special attack way can exploit those vulnerabilities. A backdoor attack is triggered only when the model gets a specific input, which then causes the neural network to produce faulty output, under federated learning condition, the server will assign the same model to local clients and receive results from clients, so a backdoor attack is very covert and not easy to be detected. Therefore, analyzing attack ways of backdoor and finding useful defend ways are significant to get a more effective and safer model.

In the paper [6], it proposes a novel attack way of Backdoor attacks named, Distributed Backdoor Attacks,DBA. DBA decomposes global triggers into several local triggers, assigning them to attackers, and embedding them into different training sets. Compared with the standard centralized backdoor attack, a distributed backdoor attack has a better performance. But, in this paper, the investigate factors and variables are not important factors that result in the incorrect prediction of Federated learning. Therefore, comparing different results by changing significant key values of distributed backdoor attacks and centralized backdoor attacks can highlight more convincing results, which is convenient for us to find useful defend way to restrain this new attack way.

II. Background

A. 1. Federated learning

Federated Learning is a new machine learning technique published by google in 2016, the target is to ensure information security, protecting the privacy of terminal data and personal data, and ensuring legal compliance during big data exchange, efficient machine learning is carried out among multiple participants or multiple computing nodes.

Federated learning can be regarded as a distributed learning process, which performs n updates rounds in several clients synchronously. For example, having a weighted average of k clients update, n_k should be the training data size of n different examples. The update equation should be:

$$w_{g,t+1} = w_{g,t} + \sum_k \frac{n_k}{n} \cdot \Delta_{k,t} \tag{1}$$

In common, Federated Learning has two forms: one is FEDSGD, another is FEDAVG. In FEDSGD, the server will send the model to each client and each client will return every SGD update to the server. Compared with FEDAVG, the difference is that clients will perform multiple iterations locally before sending updates to the server, which is more efficient in communication. However, because of the diversity of data, whatever FEDSGD or FEDAVG, they all have new vulnerabilities [2], which leads to the dataset can be manipulated by some special attack ways.

B. 2. Backdoor attack

Federated learning is generically vulnerable to model poisoning [2]. Attackers follow the rules that federated learning gives malicious clients direct chances to have an impact on the joint model, embedding overpowering attacks to influence the weights of the joint model.

In Fig.1, the attackers manipulate several clients to train their backdoor data by federated learning average algorithm, then submit the results back to the resulting model, which replace the answer of federated averaging and influence the weights of joint model.

C. 3. DBA

Backdoor attacks target to control subsets of training data by setting adversarial triggers, so that machine learning model

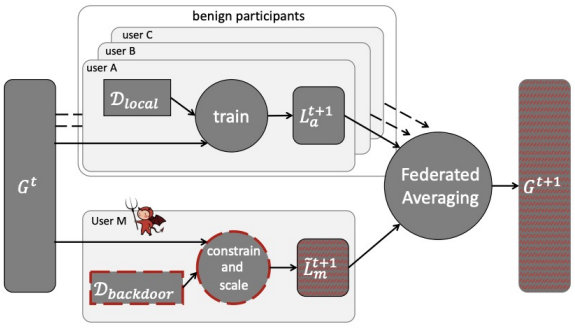


Fig. 1: Backdoor Attack

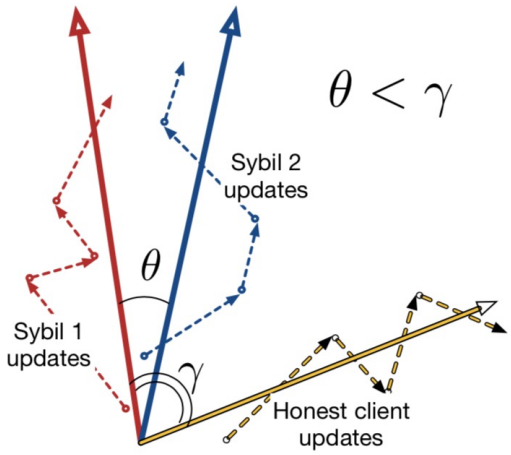


Fig. 2: gradient updates from three clients

such as Federated learning will make incorrect predictions on the targeted dataset which has embedding triggers.

Compared with standard centralized backdoor attacks, which inject poisonous triggers by global triggers, the novel distributed backdoor attacks decompose the global triggers into several local triggers and assign those local triggers to different attackers. For example, a global trigger can be divided by trigger factors, such as trigger size, trigger gap, and trigger location. The final function of a distributed backdoor attack is the same as a centralized attack, but the performance in the original paper [6], shows distributed backdoor attack has a better attack success rate than a centralized attack.

A key result in original paper, [6] Fig.3 shows comparison between DBA and CBA. It uses attack A-M and attacks A-S, totally two different attack ways and two different datasets, including LOAN, MINST, CIFAR, and TINY-IMAGENET.

Attack A-M represents the attack ways that attackers select multiple rounds and accumulated malicious updates to one successful attack. To avoid detecting by other defend ways, attack A-M weakens their updates.

Attack A-S represents the attack ways that attackers embed backdoor trigger and achieve successfully attack by one shot. To avoid overpowering by other updates, attack A-S scale malicious updates to ensure backdoor attack survive.

In Fig.3, it clear to see one global trigger by the distributed backdoor attack has lower performance than one global trigger by the centralized backdoor attack.

D. 4. Defend

1) Foolsgold: When we finish comparing distributed backdoor attacks and centralized backdoor attack, we still want to use the newest defend way to test distributed backdoor attack. Considering about several local triggers that distributed backdoor attack held. We select Foolsgold, [1], a novel defend way against multiple attackers by cosine similarity. The principle of Foolsgold is to distinguish poisonous and honest client by cosine value. Poisonous clients can achieve differences by manipulating the magnitude of the gradient, but they can not easily manipulate the direction of the gradient. Because attackers need to update poisonous several times to overpower other updates and they need to insist on one same direction. If choosing a different direction, malicious updates are hard to survive. If they focus on one shot, it is easy for protectors to detect. Therefore, Foolsgold which measures cosine value can evaluate the efficiency of attack ways. Fig.2

2) GeoMedian: The most common knowing of Geometric median is to find a point in Euclidean space which minimizes the sum distances of sample data points. In paper [4], it proposes a secure average oracle to aggregate individual device updates [3]. According to robustness to corrupted updates is a desirable property in general for distributed optimization, especially for federated learning. It designs a secure robust aggregation oracle which can be easily implemented using calls to a regular secure average oracle, relying on its built-in privacy guarantees, which besides the causes of corrupted updates. The technique named secure multiparty computation is the most efficient in general for linear functions. Therefore, this way can efficiently provide static and adaptive data corruption, as well as update corruption.

III. Problem

According Fig.3 results, it indeed shows that distributed backdoor attack has better attack success rate than centralized backdoor attack, but the original experiment considers incredible factors. The experiment keeps injected pixels as the same value for comparison. But it is meaningless for federated learning because, in federated learning, the server randomly selects the subsets of local clients, controlling the same injected pixel can not ensure distributed backdoor attack and centralized backdoor attack have the same function of global triggers, the real variable influences performance of backdoor attacks is the number of attackers. Whatever distributed backdoor attack and centralized backdoor attack, the number of attackers decides the number of malicious updates, it directly affects the weights in federated learning. If we want to test the efficiency of distributed backdoor attack and centralized backdoor attack, we should assign a fixed number of attackers, in order to control the same function of global triggers.

In addition, although the global trigger of distributed backdoor attack has better performance than the global trigger of centralized backdoor attack over four datasets. But we can not make a comparison between global triggers. According to the definition, distributed backdoor attacks need to consider all local triggers because of decomposing. We should combine all local triggers of distributed backdoor attack, comparing those with the global trigger of a centralized backdoor attack. If we do like this, one colorful local trigger in Fig.3 of distributed backdoor attack indeed has worse performance than one global trigger of centralized backdoor attack, but combing all local trigger of distributed backdoor attack, the results nearly the same as the global trigger of a centralized backdoor attack. If we treat the local trigger as a global trigger, the accuracy is not good enough. Therefore, we need to do more researches on the distributed backdoor attack and centralized backdoor attack when we fix the number of attackers. Then, we can find the significance and influence of local trigger and global trigger in these two different backdoor attacks.

IV. Experiments

A. Main Target

There are two main targets that we are going to check in this project: firstly, we want to confirm the importance of the number of attackers and the number of pixels used as triggers: as mentioned in the DBA paper [6], the authors keep the number of pixels to be the same in their experiments. Our main target here is to demonstrate that the most important element in a distributed backdoor attack is the number of attackers. Besides, we notice that in federated learning, keeping the number of pixels to be the same is not natural since each worker is allowed to take full control of their training data and the training process. Therefore, the attackers can inject any trigger they like as long as the trigger is not discovered in usage and there is no reason for them to maintain the number of trigger pixels to be the same in the attack. Secondly, we are going to explore whether there are solutions to this distributed attack problem, especially when there are multiple attackers.

B. Experiment Setting

Here we introduce the settings that we are going to implement evaluation to the distributed attacker. In DBA, there are two kinds of attack mechanisms: single-shot attack and multi-shot attack. In a single-shot attack, each attacker only injects the attack once in the whole training process. Undoubtedly, the attack will be "forgotten" by the global model after update iterations. In the single-shot setting we are mainly evaluating the robustness of the attack, i.e. how long can the attack keep its impact on the global model. In a multi-shot attack setting, each attacker launches the attack in each iteration, which means the malicious update will be accumulated and will not be weakened by the benign workers. In real federated learning scenarios, the workers are randomly selected for participation in each iteration and the attack result may be largely different to single-shot and multi-shot settings due to the randomness in participant selection while they can serve as the lower bound and upper bound respectively of the attack performance.

We conduct our experiments on two datasets for the evaluation: MNIST and CIFAR10 datasets. MNIST is a dataset with 60000 handwriting number images in 10 classes and CIFAR10 is a dataset with 60000 images with different objects. Both datasets are used for image classification. In the attack process, firstly, the whole dataset is partitioned into subsets based on the Dirichlet distribution are the subsets are assigned to workers for training. We randomly mark a label, e.g. 2, as the malicious label and the attacker(s) will inject global trigger or local trigger to all the images with the marked label. The main goal of attackers is to mislead the global to model to misclassify the images with the trigger to another attacker-chosen label, i.e. 3. In the evaluation on attack performance part, we report the accuracy number on both *main task* and *target task*. The *main task* denotes the image classification on MNIST and CIFAR10 with the correct labels and the *target task* denotes the successful rate that the attackers mislead the global model to classify the triggered images to attacker-chosen label. The metric for main task and target task are: $Accuracy_{main} = N_{correct}/N_{test}$ and $Accuracy_{target} = N_{success}/N_{triggered}$. Here $N_{correct}$ is the number of test images that are classified correctly and N_{test} is the total number of test images. $N_{success}$ is the number of triggered images that are misclassified by the global model with an attacker-chosen label. $N_{triggered}$ is the total number of triggered images for testing.

As we are considering triggers on images, injecting trigger(s) can be implemented by setting selected pixels on the image to be zero thus we can control the number and position of the trigger pixels easily. In a distributed attack, the global trigger is partitioned into several disjoint local triggers and each attacker will keep one of the local triggers.

C. Experiment Results

Result Check. Firstly, we check the result reported in the DBA paper. In a backdoor attack, the main target for the attacker is to improve the attack accuracy while minimizing the impact of the main accuracy. In Fig.4 we report the result of comparison between centralized setting and distributed setting. In a centralized attack, we use the global trigger in one attacker, and in a distributed attack, there are four attackers with four local triggers respectively. Each local trigger holds a quarter of the number of pixels in the global trigger. Given the number of pixel in global trigger as P_{NUM}, for clear comparison, the (number of attacker, number of pixels in each attacker) tuple for comparison is: centralized: (1, P_{NUM}), distributed: (4, $P_{NUM}/4$). From the result curves, we can observe that the distributed setting is much more effective than a centralized attack. In a single-shot setting, after all the attackers have launched the attack, the global model "forgets" the malicious update of centralized attackers much quicker (in fewer rounds) than the distributed attackers. In a multi-shot setting, the success rate of distributed attackers increases much quicker than the success rate of the centralized attacker

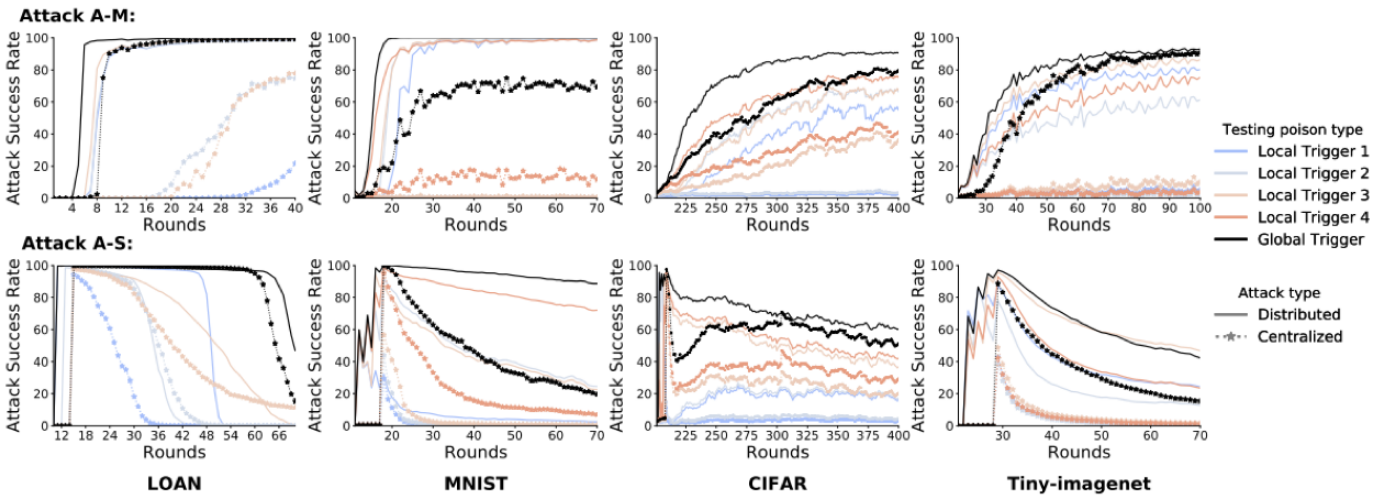
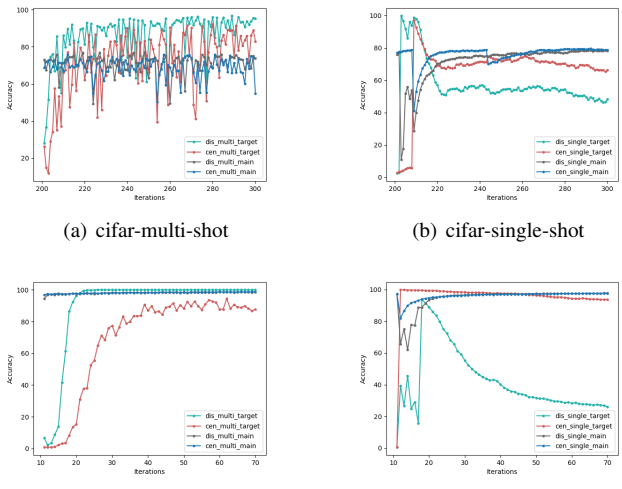


Fig. 3: Comparison between DBA and CBA

and they achieve a much higher performance upper-bound. Besides, both attack mechanisms make a little impact on the main result. Such results confirm the statement in DBA paper [6] that distributed attack is much more efficient than centralized attack in terms of training iterations and maximum success rate.



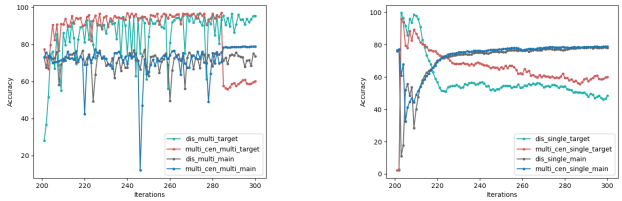
(a) cifar-multi-shot (b) cifar-single-shot

(c) mnist-multi-shot (d) mnist-single-shot

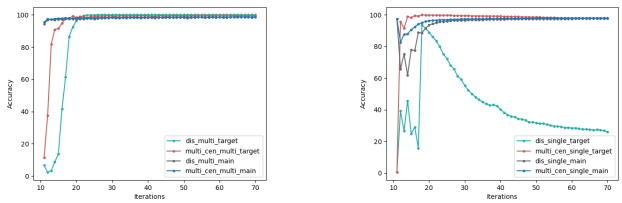
Fig. 4: distributed - centralized

Result Validation. Although we have confirmed the effectiveness of distributed setting in previous parameter setting, as we mentioned in section III, we believe the main reason for this result lies in the number of the attacker rather than the partition mechanism introduced by the distributed attack. In this part, we design experiments for validating this idea. Firstly, we increase the number of attackers in the centralized setting to the same number of the attacker in the distributed setting and each multi-centralized attacker keeps the global trigger. The experiment setting tuple here is: multi-centralized: (4, P_{NUM}), distributed: (4, $P_{NUM}/4$). Although the total number of pixels in a multi-centralized setting is four times than the distributed setting, such comparison is meaningful: as we mentioned, each worker in federated learning can take full control of the training data (images) and the whole training process, the attackers will use the origin trigger if it achieves better attack performance. The result is shown in figure . From the result curves, we perform a similar analysis to the centralized-distributed comparison. On both datasets, the multi-centralized setting has a slightly higher upper-bound on attack success rate and faster saturation speed on the multi-shot setting. In a single-shot setting, the descent is much faster in a distributed setting. Both parameter setting does not affect the accuracy on the main task much. Such experimental results demonstrate that with a certain number of malicious in control, using the global trigger on all malicious workers is a better choice than using local triggers as mentioned in DBA.

From the observation on multi-centralized advantages, we further analyze the relationship between the number of pixels and the number of attackers in the parameter setting. Following the experiment design in the DBA paper, we keep the total number of pixels as the same in different settings while changing the number of attackers. To be more specific, the number of the attacker is reduced to half, i.e. 2, and each corrupted worker uses the local trigger with half pixels of the global trigger. The experiment setting tuple here is: (half-attacker:(2, $P_{NUM}/2$), distributed: (4, $P_{NUM}/4$)). From the results in Fig.6, the distributed setting is the better choice for the attack in single-shot and multi-shot experiments. From the results on the three settings: centralized, half-attacker, distributed we can find an attack performance improvement with the increment of the number of attackers as the number of pixels used in whole is the same. Our origin idea that the number of the attacker is one of the dominant elements in attack performance can be illustrated with this trend.

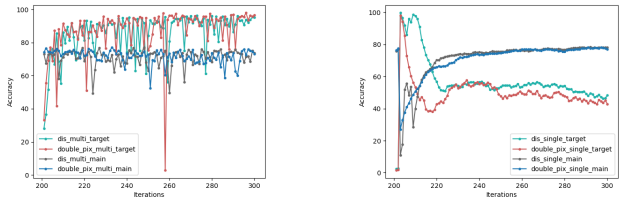


(a) cifar-multi-shot (b) cifar-single-shot

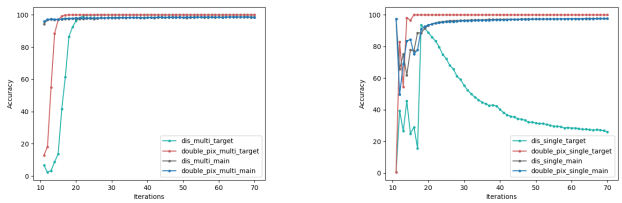


(c) mnist-multi-shot (d) mnist-single-shot

Fig. 5: distributed - multi-centralized

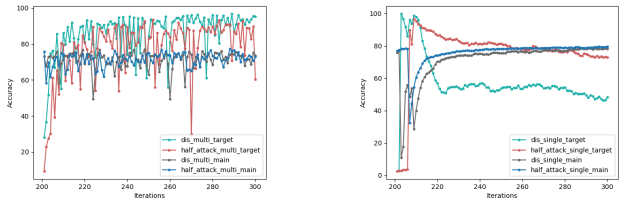


(a) cifar-multi-shot (b) cifar-single-shot

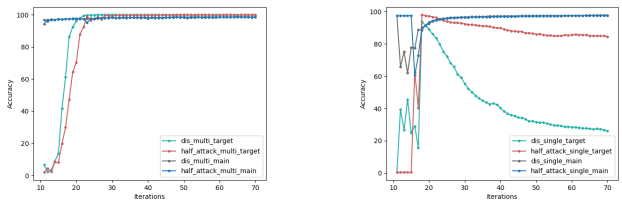


(c) mnist-multi-shot (d) mnist-single-shot

Fig. 7: double-pixel



(a) cifar-multi-shot (b) cifar-single-shot



(c) mnist-multi-shot (d) mnist-single-shot

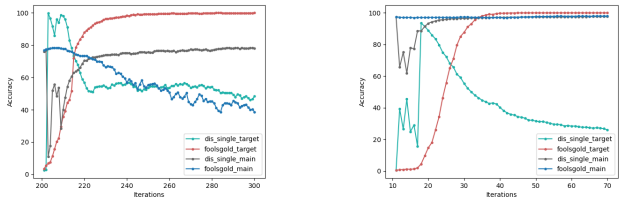
Fig. 6: half-attack

Besides the influence of the number of corrupted workers, we also want to evaluate the impact of the number of pixels in the distributed attack. Intuitively, the target task of attackers can be viewed as a trigger detection task to the global model. Specifically, if the (local or global) trigger is detected, then the image should be classified as attacker-defined labels. Thus, the attack success rate should be higher as the task difficulty can be reduced with larger triggers. With such guess, we conduct the experiments with double pixel setting: (double-pixel:(4, $P_{NUM}/2$)) and show the result in Fig.7. From the red curve denoting the performance of the double-pixel setting, it is slightly better than the original distributed setting on target task accuracy and convergence speed (over iterations). Therefore, for the attackers, using larger triggers helps improve the attack performance while the trigger might easier to be discovered.

As a summary of this performance validation part, the most important characters in carrying out an effective attack to a federated learning system are the number of malicious workers participating in training and the number of pixels used for constructing the trigger. The partitioning mechanism proposed in the distributed setting in the DBA paper cannot introduce an improvement in the attack performance.

Defense From the experiments on attack performance analysis, the backdoor attack with multiple malicious participants in a distributed setting is much more dangerous than with only one attacker. The defense to such an attack remains to be an open challenging problem due to the difficulty in distinguishing the corrupted model weight updates from the benign worker updates. Attempting to solve this problem or at least mitigate such attack, we test the performance of two state-of-art defense algorithms, GeoMedian (RFA) [4] and FoolsGold [1] on distributed attack setting.

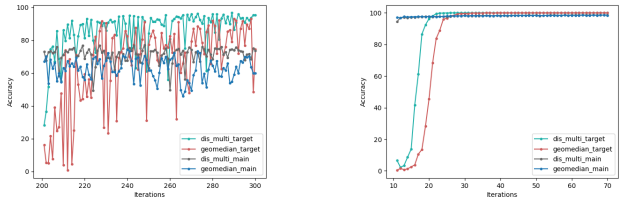
Fig.8 and Fig.9 present the defense performance on two datasets on the multi-shot setting. As the defense algorithms work on the weight aggregation phase, testing the performance in a single-shot setting is not meaningful since in most iterations there is no malicious update.



(a) cifar-foolsgold (b) mnist-foolsgold

Fig. 8: foolsgold

There are several important observations we can get from



(a) cifar-Geomedian (b) mnist-Geomedian

Fig. 9: Geomedian

the performance of GeoMedian and FoolGold: Firstly, they cannot introduce limitations on the best target accuracy that attackers can achieve although they both mitigate the attack by limiting the convergence speed of the target task. Secondly, both algorithms harm the performance of the main task on the CIFAR10 dataset, which is very unpleasing since we do not want to sacrifice the model accuracy on the main task. Especially, Foolsgold even helps the attackers' target task to get higher accuracy

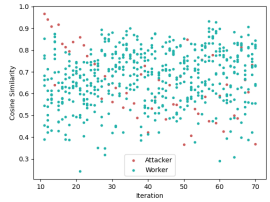
There is also an idea introduced in [5] by adding noise to the update weight to prevent the attacker target task achieves high accuracy. However, according to our experiments, such an approach will also make a huge impact on the performance of the main task and in some cases, the global model cannot achieve convergence.

Further Exploration. Motivated by the defense algorithms, we made many attempts to design the algorithm to serve as a good solution to this problem. The key idea of defending against backdoor attack is to distinguish the weight update from malicious attacker based on its distribution. However, this task can be very difficult so far as there left much space for attackers to hide in the large scale of deep learning model weights. Solving such a problem requires deep insight into the distribution of parameters in the deep learning model. To our knowledge, all previous approaches to defending the backdoor attack take all the weights from a worker as a whole to decide whether it is malicious or not. We have an idea that the discriminative weights in different parts of the model are not the same. To illustrate this, we compute the cosine distance of parameters in different parts of a deep learning model between the mean update of all workers (malicious workers + benign workers). We consider the model used in MNIST dataset with two convolutional layers and two fully connected layers. The cosine similarity is computed as $CosineDistance = \frac{MeanWeights \cdot WorkerWeights}{|MeansWeights||WorkerWeights|}$. The cosine distance distribution with the training iteration in a centralized setting and distributed setting are presented in figure 10 and figure 11. The cosine distance are normalized to $[0, 1]$. The larger cosine similarity value represents a higher similarity to the mean value of the update wight of all participant workers. There are 10 participants in each iteration while in the distributed setting there are 4 attackers in each iteration and the centralized setting, there is only one attacker. From the similarity distribution result, we can figure out several interesting conclusions: firstly, the bias is not discriminative and they seem to be distributed randomly over benign workers and malicious workers. Such a result is quite natural since the bias is not a determining factor for extracting and processing the images' features. Therefore, in the backdoor defense algorithms, the bias distribution is not as helpful as the distribution of weights in the fully connected layer and the convolutional layer. Secondly, there is a clear trend on the training iteration that the update weight of malicious attackers becoming less similar to the mean weights in both the fully connected layer and the convolutional layer. With such a finding, it shows that discovering the attackers at the late training process will be much easier if we are using cosine distance. Finally, the weights in FC layer1 are the most discriminative and should be emphasized in the defense period. As they show clear similarity to the mean value of the update weight and the weight from the attackers are not, it would be easier to differentiate them from each other with cosine-related metrics. With these figures, we want to demonstrate the idea that the weights in a deep learning model should be considered differently in terms of the defense to backdoor attack. However, as the model structure changes widely among different tasks, it can be very hard to give an algorithm that can handle all the deep learning models.

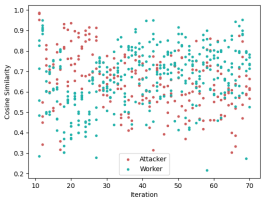
V. Conclusion

In this project, we are motivated by the backdoor challenge, which is a difficult open problem in the federated learning field. With the doubt on the findings of the DBA paper, we decided to further explore this problem. In the beginning, we implement a federated learning framework and several state-of-art defense algorithms for experiments on the distributed backdoor attack. With experiments on the most important element in the DBA algorithm: the number of attackers and the number of pixels used in the trigger, we demonstrate that the statement in the DBA paper is not completely correct and in general federated settings, improving the number of pixels and attackers will achieve a much better attack performance. The trigger partition mechanism introduced in [6] is not the most important factor to either the attack or the defense. We confirmed the importance of the number of attackers in the backdoor attack in both single-shot and multi-shot setting. When we were trying to integrate NLP tasks into the DBA framework, we find it to be extremely difficult since the trigger in NLP tasks such as word prediction, cannot be partitioned.

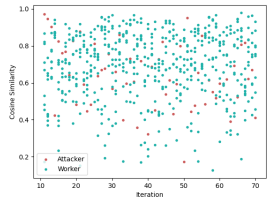
With the performance of the defense algorithm, FoolsGold and GeoMedian, we find both algorithms cannot solve the distributed backdoor problem and these defense algorithms may bring harm to the performance of our main task. We need an algorithm with deeper insight into the deep learning model parameter distribution to handle this attack effectively. In the final part, we present our thinking on this problem and point out the difference of importance of weights in different parts of the global model. However, extending such discovery to larger and more complex models remains to be a daunting task.



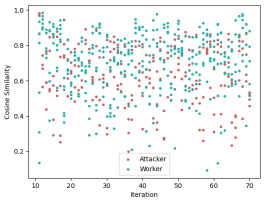
(a) cen - conv1.weight



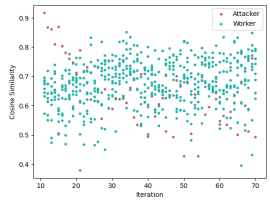
(b) dis - conv1.weight



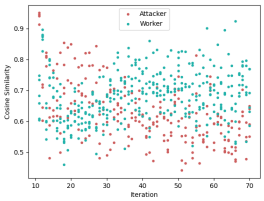
(c) cen - conv1.bias



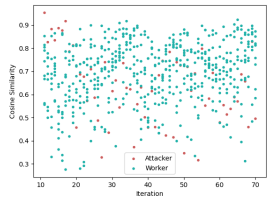
(d) dis - conv1.bias



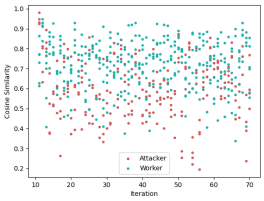
(e) cen - conv2.weight



(f) dis - conv2.weight

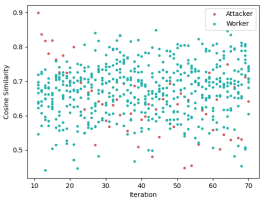


(g) cen - conv2.bias

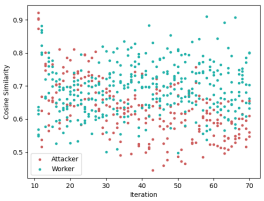


(h) dis - conv2.bias

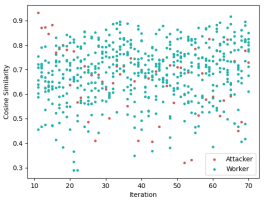
Fig. 10: Convolutional Layer Cosine Similarity



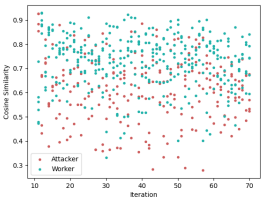
(a) cen - fc1.weight



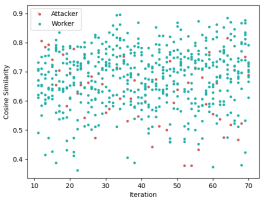
(b) dis - fc1.weight



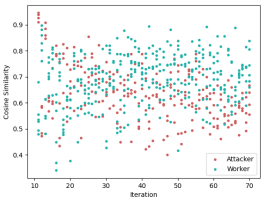
(c) cen - fc1.bias



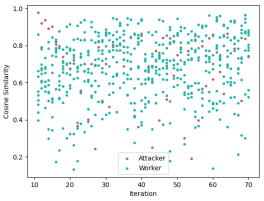
(d) dis - fc1.bias



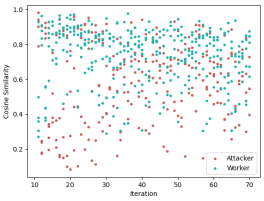
(e) cen - fc2.weight



(f) dis - fc2.weight



(g) cen - fc2.bias



(h) dis - fc2.bias

Fig. 11: FC Layer Cosine Similarity

Although we do not agree with the effectiveness of the partition mechanism introduced in DBA, this can still be an interesting problem to be discussed. For example, given a certain number of malicious workers and the specific global trigger, assigning the same global trigger to all attackers may not be the most efficient way. If we can figure out a partition or pre-processing mechanism to the origin trigger to optimize the attack performance, then we can implement a stronger attack.

REFERENCES

[1] Ivan Beschastnikh Clement Fung, Chris J.M. Yoon. Mitigating sybils in federated learning poisoning. *Machine Learning*, 2018.

[2] Yiqing Hua Deborah Estrin Vitaly Shmatikov Eugene Bagdasaryan, Andreas Veit. How to backdoor federated learning. *Cryptography and Security (cs.CR); Machine Learning (cs.LG)*, 2018.

[3] B. Kreuter A. Marcedone H. B. McMahan S. Patel D. Ramage A. Segal K. Bonawitz, V. Ivanov and K. Seth. Practical secure aggregation for privacy-preserving machine learning. *ACM SIGSAC Conference on Computer and Communications Securit*, page 1175–1191, 2017.

[4] Zaid Harchaoui Krishna Pillutla, Sham M. Kakade. Robust aggregation for federated learning. *Machine Learning,Cryptography and Security*, 2019.

[5] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[6] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. *International Conference on Learning Representations*, 2019.