Distributed Backdoor Attack in Federated Learning

Wang Ruizhe
118010301@link.cuhk.edu.cn
School of Science and Engineering

I. Introduction

Increasingly, phones and tablets are the primary computing devices for most of the people [1]. As the computation power of the devices is enhancing and the fact that they are frequently carried, an extremely large amount of data is generated with much of which private in nature [2]. Nowadays, it is increasingly attractive to locally store the data, leverage enhanced local resources on each device and push network computation to the edge [3]. At the same time, the increasing awareness and concern of privacy lead to more significant risks, and responsibilities of storing data in a centralized location. Those facts result in a growing interest in federated learning (FL) [2], which explores training statistical models directly on remote devices.

Federated learning enables mobile phones to learn a shared prediction model under the coordination of a central server while keeping all the training data on the device, decoupling the ability to do machine learning from the need to store the data in the cloud [4]. Under this approach, the local data of each client will not be directly accessed by the central server. Compared with traditional centralized machine learning, federated learning outperforms in terms of data privacy and security.

II. Overview of Federated Learning

A. Definition of Federated Learning

According to [5], federated learning distributes the training of a deep neural network across n participants by iteratively aggregating local models into a joint global model.

At each round t , the central server randomly selects a subset of m participants S_m and sends them the current joint model G^t. Each selected participant updates this model to a new local model L^{t+1} by training on their private data and sends the difference $L_i^{t+1} - G^t$ back to the central server. The central server averages the received updates to obtain the new joint model:

$$G^{t+1} = G^t + f(L_i^{t+1} - G^t) \tag{1}$$

where f is the aggregation function.

B. Application of Federated Learning

As is shown in I, federated learning can be applied in scenarios where data cannot be directly aggregated for training machine-learning models owing to factors such as intellectual property rights, privacy protection, and data security [6]. For example, the healthcare industry, financial technology company, insurance sector, blockchain technology, and Internet of Things (IoT).

C. Challenges of Federated Learning

Even though the application of FL is promising, it still faces many challenges. Concluded in [2], [3], [7], the core challenges is included as the following.

- **Systems heterogeneity** The storage, computational, and communication capabilities of each device in federated networks may differ due to variability in hardware, network connectivity, and power.
- **Statistical Heterogeneity** Devices frequently generate and collect data in a highly non-identically distributed manner across the network. This paradigm violates frequently used i.i.d. assumptions in distributed optimization.
- **Limited Communication** Federated networks potentially comprise a massive number of devices, and communication in the network can be slower than local computation [8].
- **Privacy concerns** Communicating model updates throughout the training process can nonetheless reveal sensitive information, either to a third-party or the central server [9].
- **Security concerns** Attacks on FL come from either the privacy perspective when a malicious participant or the central server attempts to infer the private information of a victim participant, or the robustness perspective when a malicious participant aims to compromise the global model.

In this study, we mainly focus on the security issues of federated learning.

III. Security Problems of Federated Learning

A. Reason for Vulnerability of Federated Learning

1) FL in a larger landscape has numerous clients that are open for attackers to exploit model parameters and training data. Access to the global model may be further vulnerable to data reconstruction attacks [10].
2) The data is stored locally, only the local gradients are available on the server side. As a result, the examination of local data is disabled and many defense methods targeted at data and developed in centralized machine learning are disabled [7].
3) The aggregation algorithm of the central server may not be able to identify abnormality with client updates, or cannot be robust enough to defense against different types of attacks.

B. Attack to Federated Learning

The category of the attack to federated learning, based on the goal of the attack, can be defined as follow:

- *non-targeted attack* (Byzantine attack) where the adversary aims to destroy the convergence and performance of the global model [11], [12], which may eventually result in a denial-of-service attack [7]
- *targeted attack* (Backdoor attack) where the adversary aims to implant a backdoor trigger into the global model so as to trick the model to constantly predict an adversarial class on a subtask while keeping good performance on the main task [13].

Based on the way of implementation, we can define two types of attack:

- **Data Poisoning** The data poisoning mentioned here mainly refers to dirty-label poisoning [7], where the adversary can introduce a number of data samples it wishes to be misclassified by the model with the desired target label into the training data. For instance, swap the label of part of the samples to the one that the adversary desire or modify individual features or small regions of the original training dataset.
- **Model Poisoning** The attacks aim to thwart the learning of the global model or hide a backdoor trigger into the global model. After training the local data, the adversaries modify the model and upload that to the central server. In a non-targeted attack, the adversary uploads arbitrarily malicious model updates to cause the failure of the global model [11]. In a targeted attack, the adversary ambitiously attempts to substitute the new global model with a malicious model, causing the global model to converge to the one that the adversary desire [5].

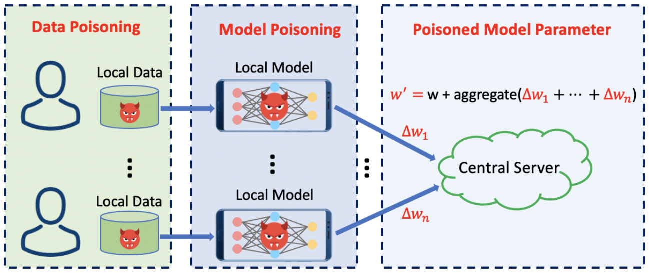


Fig. 1. Data v.s. Model Poisoning Attacks in FL

In this study, we mainly focus on targeted attack to federated learning, which is known as backdoor attack.

IV. Backdoor Attack of Federated Learning

A Backdoor attack is a targeted attack using both data and model poisoning. The most significant feature of it is making the learned models behave differently on certain targeted sub-tasks while maintaining good overall performance on the primary task [14]. For example, in image classification, the adversary can control the model to misclassify the ”white cat” as a car while ensuring that other cats are correctly classified.

We mainly focus on 3 papers exploring the backdoor attack of federated learning.

A. How To Backdoor Federated Learning

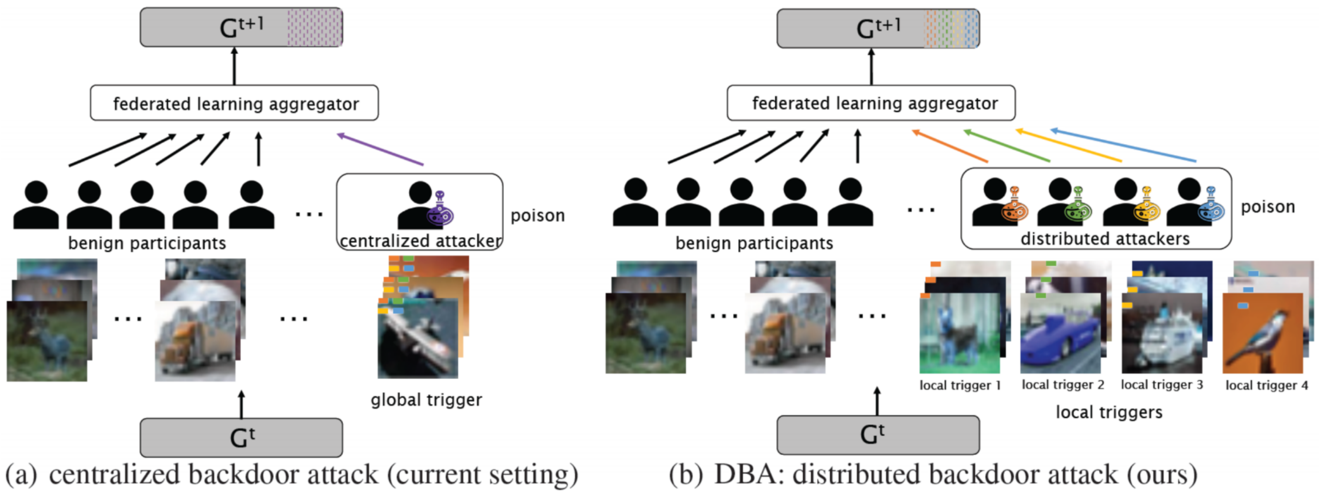
Since backdoor attack is a poison attack, the key point here is how the attackers obtain the poisoned model and upload it to the central server. In [5], the author presents 2 approaches of backdoor poisoning attack.

- **Naive approach** The attacker can simply train its model on backdoored inputs with each training batch including a mix of correctly labeled inputs and backdoored inputs. However, it does not work against federated learning as aggregation cancels out most of the backdoored model's contribution and the joint model quickly forgets the backdoor.
- **Model replacement** The attacker ambitiously attempts to substitute the new global model G^{t+1} with a malicious model X in Eq.(1):

  $$X = G^t + f(L_i^{t+1} - G^t) \tag{2}$$

  To obtain X, the local model is trained with poisoned data injected with backdoors. Besides, before updating, the backdoored model should be scaled:

  $$\tilde{L}^{t+1} = \gamma(X - G^t) + G^t \tag{3}$$



(a) centralized backdoor attack (current setting)

(b) DBA: distributed backdoor attack (ours)

Fig. 2. Overview of Centralized Backdoor Attack and Distributed Backdoor Attack on Federated Learning

where γ is the scaling factor. Therefore, by uploading the poisoned model $\tilde{L}^{t+1}$, the new joint model, after aggregation, will become closer to the adversary desired backdoored model X instead of correct model G^{t+1}.

However, the value of γ should be chosen carefully. Too large γ will negatively affect the performance of global model on primary task while too small γ will result in implanted backdoors being quickly forgot by global model.

B. Can You Really Backdoor Federated Learning

The attacking approach in IV-A only focuses on the one-time attack of a single attacker. In [14], the author looks into the sampling of multiple adversaries in the overall learning process. Two sampling schemes are investigated: (1) *random sampling* attack following hypergeometric distribution and (2) *fixed frequency* attack where a single adversary appears in every f rounds. As in [5] the poisoned model needs to scale up before uploading, this paper provides a *norm bound* setting of the adversary to control the scaling under a proper range and guarantee stealthiness.

Besides, the paper also presents a feasible defense method called *norm thresholding of updates*. The approach is to set a threshold M for the norm of the model. The server simply needs to ignore updates whose norm is above the threshold. However, if the adversary knows the threshold, it can always control the model under the norm bound to make sure the malicious update will not be ignored. Under this scenario, the norm-clipping approach is given in the paper:

$$\Delta w_{t+1} = \sum_{k \in S_t} \frac{\Delta w_{t+1}^k}{\max(1, ||\Delta w_{t+1}^k||_2/M)} \tag{4}$$

where Δw_{t+1} is the aggregate update and Δw_{t+1}^k is the local update of client k.

In addition, week differential privacy is also introduced in the defense. By adding Gaussian noise to the aggregate update, it is more difficult for the adversary to detect the threshold.

C. DBA: Distributed Backdoor Attacks Against Federated Learning

In both of the papers above, no matter how many adversaries participate in the poisoning attack, the tasks among them are the same. On the contrary, [15] investigated the cooperation of different adversaries implementing different backdoor sub-tasks to achieve the same backdoor attack as previous experiments. This process is defined as Distributed Backdoor Attack (DBA). Relatively, the original setting where all adversaries carry the same backdoor task is defined as Central Backdoor Attack (CBA).

The description of process of DBA and CBA is given in Fig. 2 from [15]. Here the backdoor attack is a data poisoning attack. In both schemes, the adversary adds a specific pattern to the upper left corner of the image, which is the **backdoor trigger** that the adversary implants. The difference is that, in a centralized attack, the adversary uses a single global trigger while in distributed attack, each adversary uses a local trigger that is part of the global one. The combination of all local triggers is exactly the global trigger.

Our study is mainly based on the work in this paper. Thus we will introduce more details of DBA and the improvement of existing works in the following sections.

V. Distributed Backdoor Attack

A. Formulation of Backdoor Attack

- **General Formulation of Backdoor Attack**
  Denote poisoned dataset as S^i_{poi} and clean dataset as S^i_{cln}. They satisfy $S^i_{poi} \cap S^i_{cln} = \emptyset$ and $S^i_{poi} \cup S^i_{cln} = D_i$, where D_i is the local dataset or attacker i. The adversarial objective for the attacker in round t with target label τ is

$$w_i^* = \arg\max_{w_i}\Big(\sum_{j \in S^i_{poi}} P[G^{t+1}(R(x^i_j, \phi)) = \tau] + \sum_{j \in S^i_{cln}} P[G^{t+1}(R(x^i_j)) = y^i_j]\Big) \tag{5}$$

  where function R transforms clean data in any class into backdoored data that have an attacker-chosen trigger pattern using a set of parameters ϕ.
- **Formulation of Distributed Backdoor Attack**
  In DBA, we consider M attackers with corresponding M small local triggers. Each attacker m_i independently performs backdoor attack on their local models. In this scenario, the formulation can be decomposed into

$$w_i^* = \arg\max_{w_i}\Big(\sum_{j \in S^i_{poi}} P[G^{t+1}(R(x^i_j, \phi^*_i)) = \tau; \gamma; I] + \sum_{j \in S^i_{cln}} P[G^{t+1}(R(x^i_j)) = y^i_j]\Big), \forall i \in [M] \tag{6}$$

  where $\phi^*_i = \phi, O(i)$ is the geometric decomposing strategy for the local trigger pattern off attacker m_i and $O(i)$ entails the trigger decomposition rule fort m_i based on global trigger ϕ. I is poison round interval and γ is scale actor to manipulate the update before submitting to the aggregator.

B. Experiments in the Paper

- *CBA vs. DBA*: In this experiment, the author investigates two attack schemes: (1) **single-shot** where each adversary only upload poisoned model once in the entire training process; and (2) **multi-shot** where each adversary upload poisoned model in every round of the training process. Note that in multi-shot, the poisoned model is not scaled before uploading to avoid influencing the performance of the global model. The result shows the superiority of DBA over CBA.
- *Robustness of DBA*: In this experiment, two robust aggregation algorithms presented by [16] and [17] are applied to test the performance of DBA. The result shows that DBA can successfully perform the backdoor attack under those defense algorithms while CBA fails.

Finally, the author looks into the effect of trigger factors. Several parameters in the attacking process are investigated to explore the relationship with the attack. The parameters include scale factor, trigger location, trigger gap (distance between different local triggers), poison interval, poison ratio, and data distribution.

C. Discussion of the Paper

In the experiment of [15], it is verified that DBA outperforms CBA in terms of robustness under defense aggregating algorithm and accuracy of backdoor task. However, the accuracy of the global model on primary task is not given in the paper. Apart from high accuracy, the backdoor attack must maintain the accuracy of the global model on the primary task simultaneously. This part of the information is needed in our investigation. Besides, we know that the defense is usually conducted on the central server, and it has no knowledge of the backdoor task of the adversaries. Therefore, the test of the global model over clean data may give the valuable clue to find out whether the model is poisoned. And it is helpful in the design of the corresponding defense algorithm.

As is mentioned in IV-B, [14] gives the schedule of each clients being selected. The multi-shot in [15], on the other hand, designs every adversary to update the poisoned model every round until the end of the entire learning. Hence, more schedules in multi-shot can be investigated to give a better understanding of the more realistic scenario in the backdoor attack of federated learning. And it can be implemented by modifying the poison interval I in Eq. (6) and scale factor that is to maintain the stealthiness of backdoor attack.

Additionally, the paper also mentioned the effects of poison interval. It notes that attack performance is poor when all distributed attackers submit the scaled updates at the same round or the poison interval is too long. We will also check this result through experiment and find out the real situation under those scenarios

D. *Dataset and Sample Poisoning*

The dataset applied in this study is MNIST, which is a database of handwritten digits with a training set of 60,000 examples and a test set of 10,000 examples. For each sample, it is an image centered in 28x28 pixels. Therefore it can be extended as a vector with 784 entries.

The backdoor trigger of the MNIST sample is implanted as shown in fig. 3. Each black rectangular with a size of 1x4 pixels on the upper left corner of the image is the local trigger of each adversary. The four local triggers together form the local trigger. According to [15], the trigger factor should be the feature with the lowest importance in the figure to maintain the stealthiness, which explains the location of the backdoor triggers.

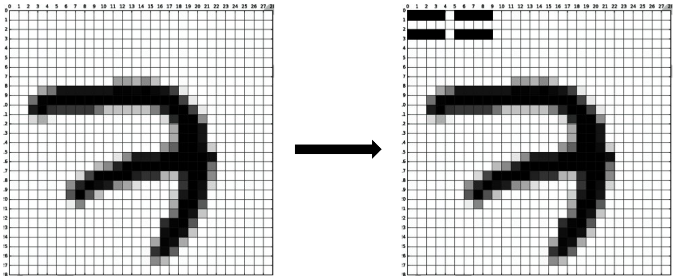


Fig. 3. Poison trigger of MNIST sample

With the implanted trigger, the adversary can attack the global model and make it regard ’7’ in fig.3 as the number other than ’7’ that it wants if an image with the trigger is tested.

E. *Experiment Settings*

According to [5], the poisoned attack should be conducted when the global model starts to converge. Therefore, the models are pretrained for 10 rounds with totally clean data to create the same initial model. After that, the training of the model will continue until round 70. Besides, due to the convergence of the global model, the same schedule of attacks will give the same final result no matter which round the attack starts, all the attacks in the experiment are conducted in round 11 - 30. Other detailed settings of the training and test process are the same as [15].

In the experiments, we evaluate the performance of attack through the test of models over 3 types of data: (1) Clean data, (2) Data implanted with global backdoor trigger, and (3) Data implanted with different local triggers. The goal of the backdoor attack is to keep high accuracy in the test of clean data while achieving high accuracy in the test of data implanted with the global trigger.

VI. Experiment Results & Analysis

A. *Single-Shot Concentrated DBA*

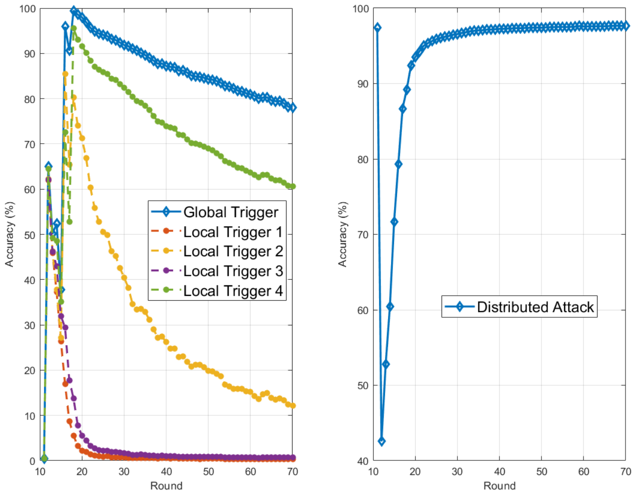


Fig. 4. Result of Distributed Backdoor Attack - Single-Shot

Firstly, we reproduce the most representative result in [15] as a baseline for comparison. Four attackers conduct a single-round poison attack at round 12, 14, 16, and 18, respectively.

The left graph shown in Fig. 4 is the reproduced result of the paper, which tests the global model on poisoned data (local trigger & global trigger). The curves accord with our expectation: each time a poisoned model is aggregated, the accuracy of the global trigger rises. When all local triggers are updated, the accuracy of the global model reaches the maximum and slowly drops, meaning that the global model is being slowly corrected by the clean data.

Besides, we checked the test result on clean data presented in the right graph of Fig. 4. And there is an interesting result not mentioned in the paper. Even though there are 4 attacks from different adversaries, the primary accuracy does not drop 4 times. It acts like one single shot at round 12. We explain that the attacks of different adversaries are too concentrated so that the global model is not recovered yet. Although different adversaries implant different local triggers, we can still observe that local triggers 2-4 are activated when local trigger 1 is implanted. Also, from the perspective of the global model, it vector of parameters is shifted to the targeted model. As in the setting, we set the gap of modified patterns in different triggers very small, the vector of parameters of models with different target models are close to each other. As a result, later local triggers will not significantly affect the performance of the global model on primary tasks. Nevertheless, the effect caused by the first adversary is still too huge due to the large-scale factor, which violates the stealthiness of backdoor attack.

B. Single-Shot Disperse DBA

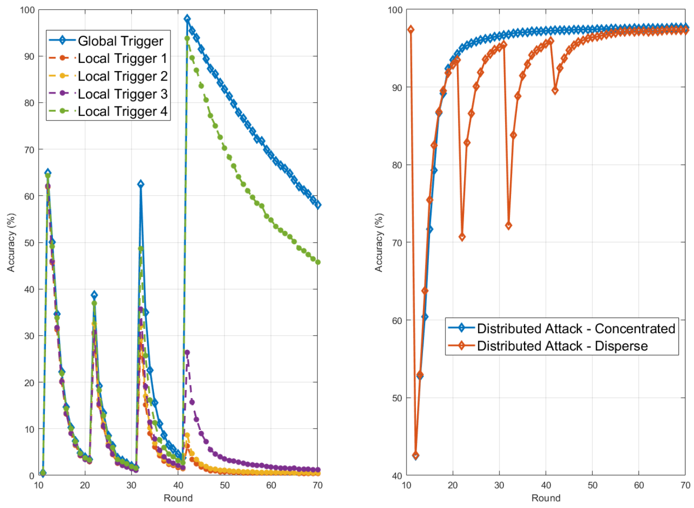


Fig. 5. Single-Shot Disperse DBA vs. Single-Shot Concentrated DBA

In VI-A, there is only 1 round between each attack. Here, we extended this interval to 9 rounds, and the result is given in Fig.5. The performance of the global model is shown on the right graph. Here we can see 4 significant drops of the test on clean data in disperse attack. As the interval between each attack becomes larger, the global model can partially recover the performance and nearly correct the backdoor model (left graph) with benign updates.

Apart from verifying conclusions in the paper, we find out in the left graph that after all adversaries conducted the backdoor attack, the backdoor accuracy decays much faster compared to concentrate attack. As the first 3 attacks are nearly corrected by the global model, the overall global trigger is more like the last local trigger instead of the combination of four triggers. As is shown in the graph, after the last attack, the accuracy of triggers 1-3 becomes quite low, and local trigger 4 is close to the global trigger.

C. Multi-Shot Continuous DBA

Next, we consider the case where each adversary conducts multiple rounds of attack. In [15], the multi-shot attack is to set every adversary to attack every round from round 11 to 70, with all the poisoned updates not scaled. We found two drawbacks of this setting. First, every round attack is an ideal assumption of the adversary. Just as discussed in V-C, to obtain a more realistic multi-shot attack scenario, we need to have different settings. Second, in the figure in [15], the backdoor accuracy has already reached the maximum around round 30. Therefore, in this session, we set the continuous attack only in round 11-30 and check the robustness of the attack after all the poison update.

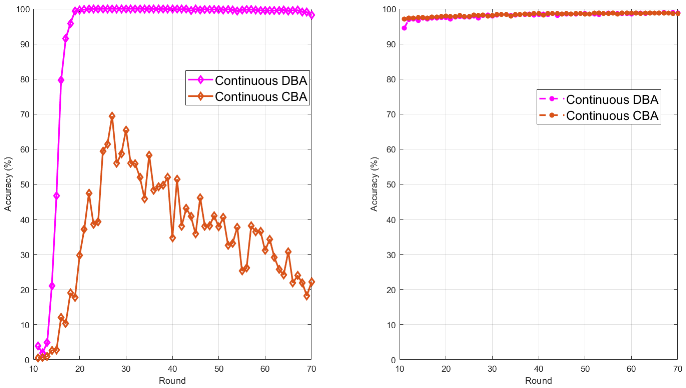


Fig. 6. Continuous DBA vs. Continuous CBA

The result is shown in fig.6 with the left and a right graph showing backdoor accuracy and overall accuracy, respectively. It verifies that continuous backdoor attack without scale is successful in terms of stealthiness. For centralized attacks, the backdoor accuracy slowly increases during attack rounds and drops after attacks are finished. Distributed attack, on the other hand, maintains nearly 100% accuracy even after all the attacks. We can see that continuous distributed attack can successfully achieve the goal of the backdoor attack.

D. Multi-Shot Fixed Frequency DBA

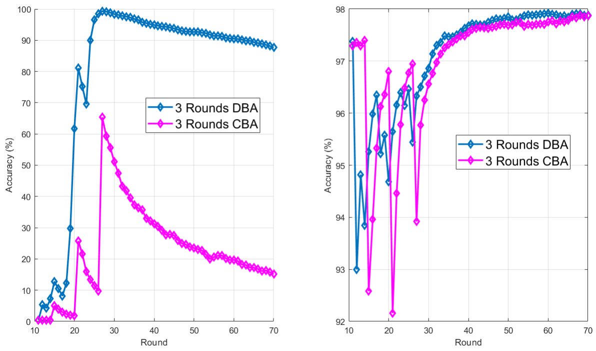


Fig. 7. 3 Collaborative DBA vs. 3 Collaborative CBA

Finally, we consider combining multi-shot DBA with fixed frequency sampling attack mentioned in [14]. The attack is scheduled as follow

- For DBA, four adversaries conduct their poison attack one by one. And that is denoted as one time of the collaborative attack. There are 3 times of collaborative attacks in total at round 12-15, 18-21, and 24-27.
- For CBA, a single adversary will conduct a poison attack at round 15, 21, 27 to correspond with DBA.

Note that the total number of attacks is increased compared with single-shot, we decrease the scale factor to 30% of the original value. And the result is shown in fig.7. Similar to VI-C, DBA can maintain relatively higher accuracy over CBA. And there are 3 significant drops in the main accuracy, which is in line with the result in previous experiments.

VII. Discussion

By the experiments in VI, we again verify the superiority of DBA over CBA from a different perspective of view. However, apart from multi-shot continuous DBA, we can observe at least one sudden drop of the main accuracy when the poison attack occurs. According to [14], boosted attacks are likely to produce updates with large norms, and thus the performance of global is seriously affected. Even though we use the settings in the paper, the attack result is not that satisfactory. We speculate that norm thresholding of updates [14] mentioned in IV-B can handle the single-shot attack. Besides, we propose another similar defense scheme: if the norm of the model is too large, i.e., exceeds the threshold, we abandon this update and aggregate the rest models.

However, the 2 approaches mentioned above are only effective in single-shot attack where the scale factor needs to be large. They cannot filter multi-shot attack with small-scale factors. DBA has been proved to successfully conduct the backdoor task in existing defense algorithms. A new robust algorithm should be designed to defend against this malicious attack.

VIII. Conclusion & Future Works

In this independent study, we first review the basic definition and process of federated learning and the challenges of FL in terms of privacy and security. Next, we focus on the security issues of FL and especially the newly-proposed distributed backdoor attack on FL. Several experiments are designed to improve the framework.

In the next step, we need to try the experiment to use 2 methods mentioned in VII to defense the single-shot attack. Additionally, a new aggregation algorithm should be developed to defense against the multi-shot attack.

References

[1] J. Poushter *et al.*, "Smartphone ownership and internet usage continues to climb in emerging economies," *Pew research center*, vol. 22, no. 1, pp. 1–44, 2016.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

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

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

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[7] L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *arXiv preprint arXiv:2012.06337*, 2020.

[8] C. Van Berkel, "Multi-core for mobile phones," in *2009 Design, Automation & Test in Europe Conference & Exhibition*, pp. 1260–1265, IEEE, 2009.

[9] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[10] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[11] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

[12] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signsgd with majority vote is communication efficient and fault tolerant," *arXiv preprint arXiv:1810.05291*, 2018.

[13] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.

[14] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?," *arXiv preprint arXiv:1911.07963*, 2019.

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

[16] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

[17] C. Fung, C. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning. arxiv 2018," *arXiv preprint arXiv:1808.04866*, 2018.