DBA-DFL: Towards Distributed Backdoor Attacks with Network Detection in Decentralized Federated Learning

Bohan Liu [1] **Yang Xiao** [2] **Ruimeng Ye** [2] **Zinan Ling** [2] **Xiaolong Ma** [3] **Bo Hui** [2]

Abstract

Distributed backdoor attacks (DBA) have shown a higher attack success rate than centralized attacks in centralized federated learning (FL). However, it has not been investigated in the decentralized FL. In this paper, we experimentally demonstrate that, while directly applying DBA to decentralized FL, the attack success rate depends on the distribution of attackers in the network architecture. Considering that the attackers can not decide their location, this paper aims to achieve a high attack success rate regardless of the attackers' location distribution. Specifically, we first design a method to detect the network by predicting the distance between any two attackers on the network. Then, based on the distance, we organize the attackers in different clusters. Lastly, we propose an algorithm to dynamically embed local patterns decomposed from a global pattern into the different attackers in each cluster. We conduct a thorough empirical investigation and find that our method can, in benchmark datasets, outperform both centralized attacks and naive DBA in different decentralized frameworks.

1. Introduction

Federated learning (FL) (McMahan et al., 2017; Kairouz et al., 2021; Bai et al., 2024) is a promising paradigm for collaborative training machine learning models over large-scale distributed data. It preserves the privacy of local data in each client and enjoys the advantage of efficient optimization as the local clients conduct computations independently and simultaneously (Andrew et al., 2024). Based on the communication architecture, existing FL frameworks can be classified into two categories: centralized FL and decentralized FL (Li et al., 2023b). Specifically, in centralized FL, the server updates the global model by aggregating the information from parties (McMahan et al., 2017; Li et al., 2020b; Wang et al., 2024; Hamer et al., 2020). In decentralized FL, the communications are performed among the parties and every party can update the global parameters directly (Bornstein et al., 2023; Li et al., 2020a; Marfoq et al., 2020; Shi et al., 2023; Dai et al., 2022)

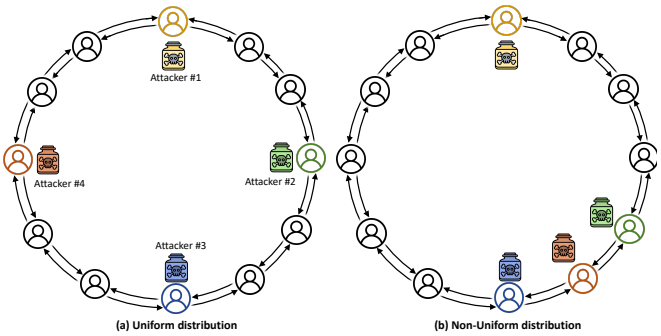


Figure 1. Location of Attackers

Despite its capability of aggregating dispersed information to train a better model, its distributed learning mechanism across different parties may unintentionally provide a venue for adversarial attacks (Bagdasaryan et al., 2020; Bhagoji et al., 2019; Garov et al., 2024). Specifically, adversarial agents can perform data poisoning attacks on the shared model by manipulating a subset of training data and uploading poisoned local models such that the trained model on the tampered dataset will be vulnerable to the data with a similar trigger embedded and data with specific patterns will be misclassified into some target labels (Dai & Li, 2023; Zhuang et al., 2024; Zhang et al., 2023b).

Due to the nature of the distributed learning methodology in FL, it is intuitive to have several adversarial parties attack FL simultaneously. DBA (distributed backdoor attacks) (Xie et al., 2020) is an attack strategy to decompose a trigger pattern into local patterns and embed local patterns to different adversarial parties respectively. Compared with embedding the same global trigger pattern to all adversarial parties, DBA is more persistent and effective, as the local trigger pattern is more insidious and easier to bypass the robust aggregation mechanism in the centralized FL framework.

[1]Carnegie Mellon University [2]University of Tulsa [3]Clemson University. Correspondence to: Bohan Liu <bohanli2@andrew.cmu.edu>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025. *Copyright 2025 by the author(s).*

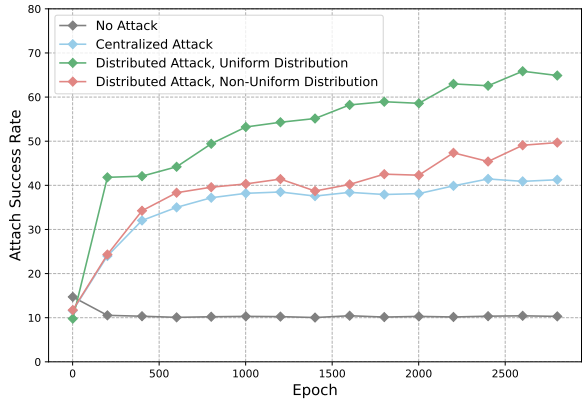


Figure 2. Attacks on D-PSGD

However, DBA has not been investigated in the decentralized FL. Intuitively, the communication algorithms may have an impact on the attack success rate of DBA. In this paper, we first introduce DBA in decentralized FL and conduct experiments to report the attack success rate. We empirically find the attack success rate highly depends on the location distribution of adversarial parties.

In Figure 2, we compare the attack success rate of two scenarios: (1) uniform distribution of adversarial parties on the topology and (2) non-uniform distribution of adversarial parties on the topology. As shown in Figure 1, the location distribution of adversarial parties can be non-uniform on the topology of the communication network. We especially found that while directly applying DBA to decentralized FL, the attack success rate highly depends on the distribution of attackers. Specifically, Figure 2 compares the attack success rate of two scenarios on D-PSGD (Lian et al., 2017) and CIFAR-10. The result shows that the attack success rate will drop significantly if the adversarial parties are not uniformly distributed on the network. This is because the model updating flow based on poisoned data is often asymmetric in the topology. Intuitively, the impact of a trigger pattern provided by an attacker will be marginal if an agent is far from the attacker.

In this paper, we aim to achieve a high attack success rate regardless of the locations of adversarial agents. First, we propose to detect the network by predicting the distance between any two attackers on the network. Specifically, we observe that the sequence of prediction accuracy of elaborated data varies differently on agents with different distances to an attacker. Based on this observation, we use the sequence to predict the distance between any two attackers in the early stage of FL. With the estimated distance, we leverage the clustering algorithm to organize the attackers in different clusters. Lastly, we develop an algorithm to dynamically decompose global trigger patterns into different adversarial agents to maximize the attack success rate. Our method has addressed the distinctive framework of decentralized FL and achieved a higher attack success rate.

We experiment with multiple decentralized FL frameworks and standard datasets to verify the effectiveness of the proposed method. We propose the following contributions:

- This work is the first to study distributed backdoor attacks on decentralized FL.
- We empirically find that while directly applying DBA to decentralized FL, the attack success rate depends on the distribution of attackers in the topology.
- We propose a method to detect the network of the decentralized FL by estimating the distance between any two agents. An algorithm is developed to dynamically organize distributed backdoor attacks.
- We experimentally demonstrate that our attacking strategy can achieve a higher attack success rate than DBA and the centralized attack with a global trigger.

2. Preliminary

Federated Learning. Centralized FL is a distributed learning framework with the following training objective:

$$\min_{w} F(w) := \frac{1}{N}\sum_{i=1}^{N} f_i(w_i) \tag{1}$$

There are N parties in the framework, each of whom trains a local model $f_i(w)$ with a private dataset $D_i = \{\{x_j^i, y_j^i\}_{j=1}^{J}\}$ where $j = |D_i|$ and $\{x_j^i, y_j^i\}$ represents each data sample and its corresponding label. At round t, a central server sends the current shared model parameterized with w to N parties. Each local party will copy w to its local model w_i. The parameter of a local model w_i will be updated with a loss of prediction $l(\{\{x_j^i, y_j^i\}_{j=1}^{J}\}, w_i)$. By running an optimization algorithm such as stochastic gradient descent, a local party can obtain a new local model w_i^{t+1}. After several rounds, the server implements an aggregation algorithm to combine the local models or model updates into a global model.

Different from centralized FL where a server communicates coordinates with all parties, decentralized FL, local parties only communicate with their neighbors in various communication typologies without a central server, which offers communication efficiency and better preserves data privacy compared with centralized FL. Denote the communication topology in the decentralized FL framework among clients is modeled as a graph $G = \mathcal{V}, \mathcal{E}$, where $\mathcal{V}$ refers to the set of clients, and $\mathcal{E}$ refers to the set of communication channels, each of which connects two distinct clients. The client adopts multi-step local iterations of training and then sends the updated model to the selected neighbors. Decentralized FL design is preferred over centralized FL in some aspects

since concentrating information on one server may bring potential risks or unfairness (Li et al., 2023b).

Backdoor attack The objective of a backdoor attack is to mislead the trained model to predict any input data with an embedded trigger as a wrong label. In federated learning, an adversarial client can pretend to be a normal client and manipulate the local model. By sending the updates to the global server, the global model would achieve a high attack success rate on poisoned data. Specifically, the training objective for an adversarial client i at round t with local dataset D_i and the target label τ is:

$$w_i^* = \arg\max_{w_i}\Big(\sum_{j\in S_{\text{poi}}^i} P[F(w, R(x_j^i)) = \tau] + \sum_{j\in S_{\text{cln}}^i} P[F(w, x_j^i) = y_j^i]\Big), \tag{2}$$

where S_{poi}^i is the index set of poisoned data samples and S_{cln}^i is the index set of clear data samples. The first sum term aims to predict the poisoned data samples as the target label t and the second sum term guarantees that the clean data samples will be predicted as the ground truth. The function $R(\cdot)$ transforms a clean data point into poisoned data by adding a trigger pattern parameterized by ϕ.

3. Method

3.1. Analysis of DBA in Decentralized FL

Assume there are N clients forming an unknown topology (e.g., ring and clique ring). A rational setting is that the adversarial clients are only aware of their neighbors and have no information ((e.g., locations) about other adversarial clients and the overall communication topology. In decentralized federated learning, each client follows a pre-defined algorithm to communicate with its neighbors, receiving model parameter information from all neighbors and aggregating it locally. Different from centralized federated learning, there is no central server to balance all parameters and each client's model is directly influenced by its neighbors. Intuitively, a client's influence on other clients over the communication topology will diminish while the distance between two clients is increasing. For example, if an adversarial client conducts backdoor attacks on the local model, the attacking effects could be marginal for a client far from the adversarial client. This is because the model updates based on the poisoned data can be canceled out along the long chain of model updates on the topology.

Accordingly, the communication algorithms of decentralized FL may have an impact on the attack success rate of DBA. We empirically find that, while directly applying DBA to decentralized FL, the attack success rate highly depends on the location distribution of adversarial clients. As shown in Figure 2, compared with the scenario where the adversarial parties are uniformly distributed on the topology, the attack success rate will drop significantly if the adversarial parties are not uniformly distributed on the network. In decentralized federated learning, the effectiveness of DBA significantly decreases due to the absence of a central server that aggregates the effects of distributed attacks. Intuitively, with a non-uniform distribution, the impact of these attacks can not fully reach out to all clients on the topology.

Motivated by this phenomenon, this paper aims to maximize the efficacy of DBA in decentralized FL. Considering that the attackers can not decide their location, we propose to adjust the strategy of DBA according to the topology. Specifically, we propose a two-step attacking strategy: (1) detecting the network (i.e., the connection between attackers) and (2) an improved DBA based on the network.

3.2. Topology Detection

Since it is evident that the locations of attackers on the topology of DFL significantly impact the attacking effectiveness, we first detect the position of the attacking nodes within the topology. If we can estimate the distance between any two attacking clients, we can better conduct the attack by controlling the overlap of attack patterns among nodes to maximize the attack's effectiveness. Therefore, our target is to design a method to estimate the distance between any two adversarial clients in an unknown topology.

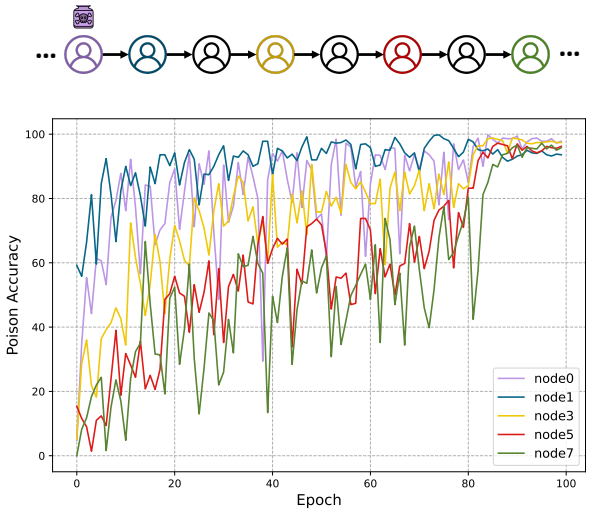


Figure 3. Sequences

In this paper, we refer to the attacking actions of adversarial clients as "signals" and the poison accuracy as "signal strength" (i.e., the accuracy of predicting an image as the attacker's desired category). For instance, if the attacker wants the model to classify a shark as a ship, the accuracy of predicting a shark as a ship with other normal clients is the poison accuracy. The higher the poison accuracy, the higher the signal strength. As the attacker initiates the attack, the signal propagates through the topology, affecting the model in each client by combining the attacker's attacking signal

and other nodes' normal signals based on local data. Since the update of the model for a normal (i.e., non-attacker) client could cancel out some impact of the attacking signal, the signal strength detected by a client could become weaker along the propagation path in the topology. Therefore, the poison accuracy on a client is influenced by its position in the topology, more precisely by its distance from the attacker. From the perspective of the training process, the poison accuracy of a client forms a sequence that varies from epoch to epoch. We remark that this sequence can be used to estimate the distance from the client to the attacker.

To further justify that the attacking signals become weaker along the propagation path, we visualize the sequence of poison accuracy for 5 clients in the training process of a decentralized FL (Amiri & Gündüz, 2020) using CIFAR-10. As shown in the upper part of Figure 3, the purple client performs backdoor attacks on the local model. Specifically, on the purple client (node 0), we assign "ships" as the label of a shark image for local training. Note that "shark" does not belong to any of the 10 classes in CIFAR-10. The purple sequence in the lower part of Figure 3 indicates the poison accuracy (the image is predicted as "ships") of the shark image. Similarly, we visualize the poison accuracy on the other clients while feeding the shark image to the local models. We can observe that the sequence gap between a client and the attacker (node 0) increases as the distance to the attacker increases. It indicates that such sequences can reveal the distance between a client and an attacker.

Based on the motivation, we predict the distance between any two attackers. Note that the attackers can communicate with each other to agree on poisoned images and the target label. Denote $\mathcal{A}$ as the set of attackers. For each attacker $i \in \mathcal{A}$, we assign a distinctive image z^i as the "signature" of attacker i. The attacker will train the model to predict $\check{x}^i$ as a random label $\tau \in \mathcal{Y}$ in the domain:

$$w_i^* = \arg\max_{w_i}(P[f(w_i, z^i)) = \tau] + \sum_{j \in S_{\text{cln}}^i} P[f(w_i, x_j^i) = y_j^i]), \tag{3}$$

Denote s_i as the sequence of poison accuracy for z^i on attacker i. For any other attacker $i' \in \mathcal{A}$ $(i \neq i')$, we predict its distance to attacker i by feeding the sequence difference $s_i - s_{i'}$ into a pre-trained LSTM model. We remark that each attacker will have a distinctive "signature" so that the attacking signals of attackers will not impact each other in terms of predicting distance.

To per-train an LSTM model $G(\cdot)$ for distance prediction, we set the distance of each direct connection on the topology as 1. With a decentralized FL for training purposes, we feed the sequence difference for any pair of attackers (i, i') for regression prediction. The model is optimized by minimizing Mean Squared Error (MSE) according to the ground truth:

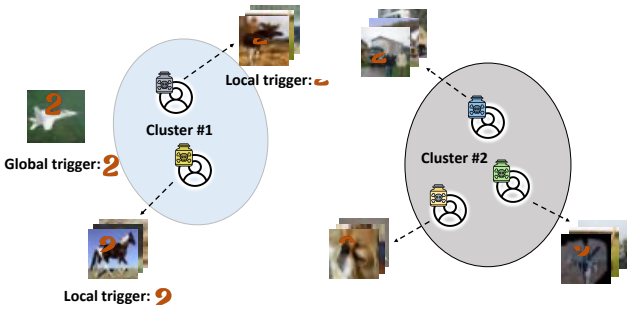


Figure 4. Distributed Patterns

$$\text{MSE} = \frac{1}{N} \sum_{(i,i'), i \neq i'} (G(s_i - s_{i'}) - d_{i,i'})^2, \tag{4}$$

where $d_{i,i'}$ is the ground truth distance and N is the number of pairs. In the experiment, we demonstrate the accuracy of predicting the distance with a pre-trained model.

4. DBA based on the detected network

Our second step is to improve DBA on decentralized FL with the detected network. We attribute the unsatisfied attack success ratio on the decentralized FL to the absence of a central server and limited coverage of attacking signals on certain clients. If we evenly decompose a global attacking trigger into local patterns at each attacker, a small local trigger may not be significant enough to propagate the all clients. To address this limitation, we propose to organize DBA based on clusters of attackers in the topology and enhance the impact of distributed backdoor attacks.

Denote M as the distance metric predicted with a pre-trained model. Each entry in M represents the predicted distance between two attackers. We leverage a clustering algorithm to assign attackers into a set of groups where attackers close to each other belong to the same group. Figure 4 shows two clusters of attackers. Then we design a distributed backdoor attack algorithm based on the clusters.

Dynamic distribution of local triggers within clusters. Suppose there are K clusters in the decentralized FL topology. As illustrated in Figure 4, we decompose a global trigger evenly into local triggers in each cluster C_k. All attackers in a cluster only use parts of the global trigger to poison the training data. For example, the attacker highlighted with blue in Cluster #1 poisons a subset of the training data only using the upper part of the global trigger and the attacker with the yellow sign uses the lower part of the global trigger to poison the data. A similar attacking the methodology applies to attackers in other clusters. We define each decomposed trigger used for each attacker as the local trigger. Considering m attackers in cluster C_k with m small local triggers. Each DBA attacker mi indepen-

dently performs the backdoor attack on their local models by solving:

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

where ϕ^i_k denotes the local trigger for client i in cluster C_k.

Algorithm 1: DBA with network detection

```
t = 0;
Assign a distinctive poison signature out of the domain
  for each attacker;
while t < ΔT do
    for i ∈ A do
        for i ∈ A do
            Compute the poison accuracy s_i for attacker
              i' concerning z_i;
    t+=1;
For any pair of two attackers, predict the distance d_{i,i'}
  from i to i' with G(·), s_i, and s_{i'};
Clustering attackers into K groups with the distance
  matrix M;
while t < T do
    for k = 0; k < K; k+ = 1 do
        Randomly assign decomposed local patterns to
          all attackers i in Cluster C_k;
        for i ∈ C_k do
            Each attacker i uses Eq. (5) to attack the
              local model;
    t+=1;
```

Note that in each attacking round, we randomly assign the decomposed local triggers to different attackers within a cluster. The benefit is that each local pattern will have the chance to be assigned at various locations. It further maximizes the overall influence of the attacking trigger.

Algorithm 1 outlines the workflow of our attacking scheme. In the early stage of learning ($t < \Delta T$), the sequence of poison accuracy will be used for predicting the distance between any two attackers. Based on the distance matrix, we can leverage any clustering algorithm based on distance to group attackers. Then in each attacking round, the global trigger will be decomposed and randomly assigned to all attackers within each cluster. Each attacker will conduct backdoor attacks with the assigned local trigger. Our cluster-based on backdoor attacks and dynamic distribution of local triggers can enhance the impact of DBA.

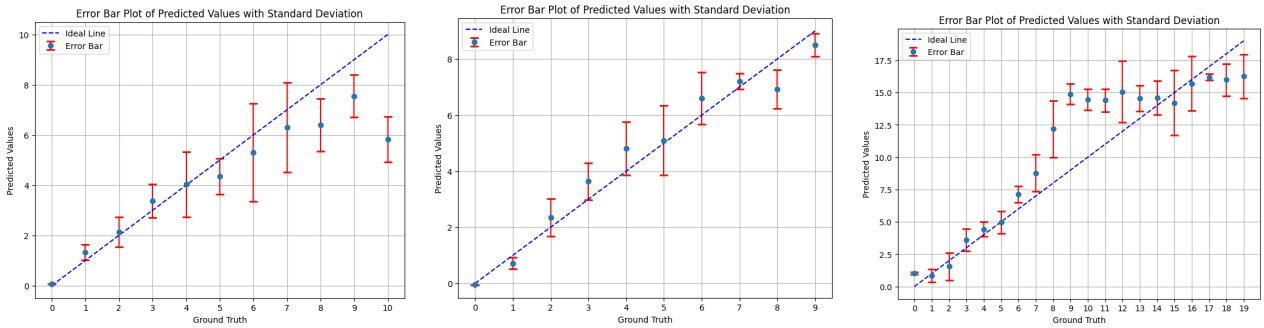
5. Experiments

Experimental Setup We follow DBA (Xie et al., 2020) to set up the experiment. We introduce two popular decentralized FL algorithms: DSGD (Amiri & Gündüz, 2020) and Swift (Bornstein et al., 2023). All training parameters are configured as the standard value in the corresponding paper. We evaluate the performance of predicting distance on two typologies: Ring and Grid. To compare with DBA and centralized backdoor attack (Bagdasaryan et al., 2020), we report the attack success rate (ASR) on two datasets: CIFAR-10 and MNIST. We use the poison accuracy of the first 100 epochs to predict distance. On each topology, there are 40 clients by default. We follow DBA to set up the attacking trigger.

Distance prediction. In the experiment, we randomly assign pairs of clients as attackers with specified ground-truth distance and leverage an LSTM model to predict distance. The experiments are repeated 20 times to combat randomness. As shown in Figure 5, we report the error of the predicted distance on two typologies with different numbers of clients. On the ring topology, we can observe the prediction error for Swift is smaller than DSGD. We attribute it to the rapid synchronization of model updates in Swift. The observations still hold for the grid topology. Also, we can see that the error increases while the ground-truth distance is increasing. This is because the attacking signal becomes weak if the distance is long and the model can not distinguish it from the signal of a non-attack client. The result indicates that our distance prediction method is accurate.

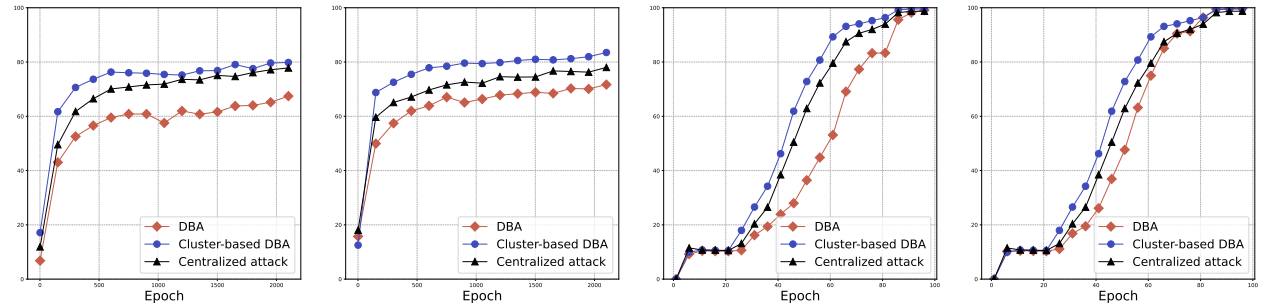
Attack success rate. Following DBA, we evaluate the attack success rates of different attacking methods using the same global trigger. The ratio of backdoor pixels in the global triggers is 0.964 for MNIST and 0.990 for CIFAR-10. For a fair comparison, we set the total number of backdoor pixels in the training dataset to be the same across different attacking methods. Specifically, we poison more data in DBA and centralized attack so that the total number of poison pixels equals that of our cluster-based DBA by including more data in S^i_{poi}. We randomly select 10 clients as attackers and cluster the attackers into 3 groups. In Figure 7, we use "Cluster-based DBA" to denote our method. We report the average attack success rate for two topologies on CIFAR-10 and MNIST. We can see that the centralized attack outperforms DBA in terms of attack success rate. This is against the motivation of DBA. It further justifies the necessity of improving DBA in decentralized FL. By applying our backdoor attack method to the decentralized FL, we can observe that our attack success rate is higher than both DBA and centralized attacks in all settings.

Attacking DFL with defensive mechanisms. To showcase the effectiveness of the proposed attack under defense mechanisms, we introduce two defensive mechanisms (Zhang



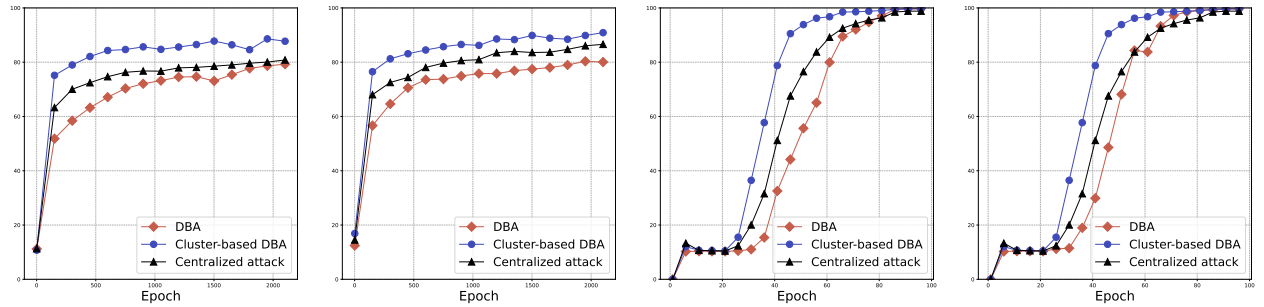
(a) Ring topology (DSGD) (b) Ring topology (Swift) (c) Grid Topology (Swift)

Figure 5. Distance Prediction



(a) Swift on CIFAR-10 (b) DSGD on CIFAR-10 (c) Swift on MNIST (d) DSGD on MNIST

Figure 6. Attack Success Rate (Ring Topology)



(a) Swift on CIFAR-10 (b) DSGD on CIFAR-10 (c) Swift on MNIST (d) DSGD on MNIST

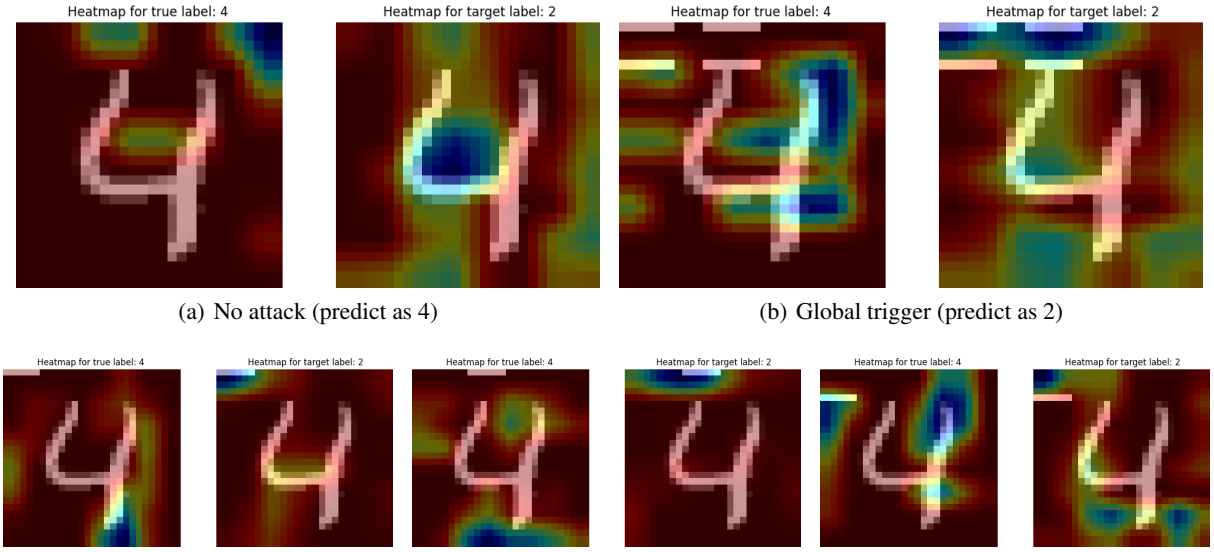
Figure 7. Attack Success Rate (Clique Ring Topology)

et al., 2023b; Jia et al., 2023) in Table 1. To the author's best knowledge, there is no defense mechanism designed specifically for decentralized FL in the literature. The possible reason is that a decentralized framework itself is a defense mechanism. Many defensive strategies based on client selection such as Krum (Blanchard et al., 2017) are not suitable for DFL. To introduce the defensive strategy in decentralized FL, we leverage the corresponding strategy for each client. Note that the defense mechanism does reduce ASR. However, decentralized FL mitigates backdoor attacks because each client only has a few neighbors (e.g., 2 on a ring topology). Compared with DBA and centralized attack, our method can further pose a challenge to the effectiveness of these defense mechanisms.

Table 1. Attacking DFL with defensive mechanism.

Method	**DBA**	**Centralized**	**Ours**
Swift	0.656	0.782	0.801
Swift+FLIP	0.431	0.699	0.783
Swift+FedGame	0.587	0.728	0.779
DSGD	0.712	0.764	0.831
DSGD+FLIP	0.679	0.688	0.787
DSGD+FedGame	0.646	0.647	0.805

Case study. In Figure 8, we use the Grad-CAM visualization method (Gildenblat & contributors, 2021) to explore a sample image attacked by DBA and centralized attack with the global trigger. The two columns show the difference



(a) No attack (predict as 4) (b) Global trigger (predict as 2)

(c) Local trigger #1 (predict as 4) (d) Local trigger #2 (predict as 4) (e) Local trigger #3 (predict as 4)

Figure 8. Case Study

Table 2. Number of clusters.

Clusters	Swift-Ring	DSGD-Ring	Swift-Clique
2 clusters	0.789	0.812	0.872
3 clusters	0.801	0.831	0.893
4 clusters	0.818	0.823	0.876
5 clusters	0.752	0.788	0.862

between two heat maps of activation (e.g., the importance for prediction) for predicting a hand-written digit '4' as '4' and '2', respectively. Same as the conclusion in DBA (Xie et al., 2020), each local triggered image alone is a weak attack as none of them can change the prediction. However, with a global trigger, the poisoned image is classified as '2' (the target label), and we can see the activation area is transformed to the trigger location. It suggests that each small local trigger is difficult to detect for defenders because most locally triggered images are similar to the clean image.

6. Related works

Using more data for model training benefit the performance in general. However, it poses privacy risk concerns by collecting data from various institutions. Federated Learning (McMahan et al., 2017; Khaled & Jin, 2023; Cheng et al., 2024; Huang et al., 2021a; Zhong et al., 2023) has emerged as a powerful distributed learning framework by sharing a global model without sharing their data. FL frameworks can be classified into two categories: centralized FL and decentralized FL (Li et al., 2023b). Centralized FL enables clients to perform limited training on local datasets while the centralized server aggregates the client parameters using different aggregation methods. (McMahan et al., 2017; Li et al., 2020b; Wang et al., 2024; Hamer et al., 2020). In decentralized FL, the communications are performed among the parties and every party can update the global parameters directly (Bornstein et al., 2023; Li et al., 2020a; Marfoq et al., 2020; Shi et al., 2023; Dai et al., 2022) to keep each client's data private.

The nature of federated learning provides a way for adversarial parties to attack the model. Since any client has access to the global model, the attacker can perform membership attacks on the model (Li et al., 2023a), data stealing (Garov et al., 2024) or model poisoning attack (Yan et al., 2023; Jia et al., 2023; Zhang et al., 2023a; Li et al., 2022; Huang et al., 2021b). Some defensive methods have also been studies (Xie et al., 2024; 2021; Zhang et al., 2023b; Fang & Chen, 2023) based on model updates. However, the attack and defense on the decentralized FL have not been studied. To the best of our knowledge, our paper is the first work to investigate DBA on decentralized FL.

7. Conclusion

In this paper, we apply DBA to decentralized FL. We experimentally demonstrate that the attack success rate of DBA depends on the distribution of attackers in the network architecture. Considering that the attackers can not decide their location, we propose a two-step attacking strategy to improve the ASR of DBA in decentralized FL: (1) detecting the network and (2) an improved DBA.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, such as safety and privacy in federated learning.

References

Amiri, M. M. and Gündüz, D. Federated learning over wireless fading channels. *IEEE Trans. Wirel. Commun.*, 19(5):3546–3557, 2020. doi: 10.1109/TWC.2020.2974748. URL https://doi.org/10.1109/TWC.2020.2974748.

Andrew, G., Kairouz, P., Oh, S., Oprea, A., McMahan, H. B., and Suriyakumar, V. M. One-shot empirical privacy estimation for federated learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=0BqyZSWfzo.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948. PMLR, 2020. URL http://proceedings.mlr.press/v108/bagdasaryan20a.html.

Bai, R., Bagchi, S., and Inouye, D. I. Benchmarking algorithms for federated domain generalization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=wprSv7ichW.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. B. Analyzing federated learning through an adversarial lens. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 634–643. PMLR, 2019. URL http://proceedings.mlr.press/v97/bhagoji19a.html.

Blanchard, P., Mhamdi, E. M. E., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 119–129, 2017.

Bornstein, M., Rabbani, T., Wang, E., Bedi, A. S., and Huang, F. SWIFT: rapid decentralized federated learning via wait-free model communication. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=jh1nCir1R3d.

Cheng, Z., Huang, X., Wu, P., and Yuan, K. Momentum benefits non-iid federated learning simply and provably. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=TdhkAcXkRi.

Dai, R., Shen, L., He, F., Tian, X., and Tao, D. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4587–4604. PMLR, 2022. URL https://proceedings.mlr.press/v162/dai22b.html.

Dai, Y. and Li, S. Chameleon: Adapting to peer images for planting durable backdoors in federated learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6712–6725. PMLR, 2023. URL https://proceedings.mlr.press/v202/dai23a.html.

Fang, P. and Chen, J. On the vulnerability of backdoor defenses for federated learning. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 11800–11808. AAAI Press, 2023. doi: 10.1609/AAAI.V37I10.26393. URL https://doi.org/10.1609/aaai.v37i10.26393.

Garov, K., Dimitrov, D. I., Jovanovic, N., and Vechev, M. T. Hiding in plain sight: Disguising data stealing attacks in federated learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=krx55l2A6G.

Gildenblat, J. and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

Hamer, J., Mohri, M., and Suresh, A. T. Fedboost: A communication-efficient algorithm for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3973–3983. PMLR, 2020. URL http://proceedings.mlr.press/v119/hamer20a.html.

Huang, Y., Chu, L., Zhou, Z., Wang, L., Liu, J., Pei, J., and Zhang, Y. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7865–7873, 2021a.

Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating gradient inversion attacks and defenses in federated learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 7232–7241, 2021b.

Jia, J., Yuan, Z., Sahabandu, D., Niu, L., Rajabi, A., Ramasubramanian, B., Li, B., and Poovendran, R. Fedgame: A game-theoretic defense against backdoor attacks in federated learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083.

Khaled, A. and Jin, C. Faster federated optimization under second-order similarity. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ElC6LYO4MfD.

Li, H., Sun, X., and Zheng, Z. Learning to attack federated learning: A model-based reinforcement learning attack framework. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Li, J., Li, N., and Ribeiro, B. Effective passive membership inference attacks in federated learning against overparameterized models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023a. URL https://openreview.net/forum?id=QsCSLPP55Ku.

Li, Q., Wen, Z., and He, B. Practical federated gradient boosting decision trees. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4642–4649. AAAI Press, 2020a. doi: 10.1609/AAAI.V34I04.5895. URL https://doi.org/10.1609/aaai.v34i04.5895.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.*, 35(4):3347–3366, 2023b. doi: 10.1109/TKDE.2021.3124599. URL https://doi.org/10.1109/TKDE.2021.3124599.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=ByexElSYDr.

Lian, X., Zhang, C., Zhang, H., Hsieh, C., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5330–5340, 2017.

Marfoq, O., Xu, C., Neglia, G., and Vidal, R. Throughput-optimal topology design for cross-silo federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information*

Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In Singh, A. and Zhu, X. J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

Shi, Y., Shen, L., Wei, K., Sun, Y., Yuan, B., Wang, X., and Tao, D. Improving the model consistency of decentralized federated learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31269–31291. PMLR, 2023. URL https://proceedings.mlr.press/v202/shi23d.html.

Wang, H., Xu, H., Li, Y., Xu, Y., Li, R., and Zhang, T. Fedcda: Federated learning with cross-rounds divergence-aware aggregation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=nbPGqeH3lt.

Xie, C., Huang, K., Chen, P., and Li, B. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgyS0VFvr.

Xie, C., Chen, M., Chen, P., and Li, B. CRFL: certifiably robust federated learning against backdoor attacks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11372–11382. PMLR, 2021. URL http://proceedings.mlr.press/v139/xie21a.html.

Xie, Y., Fang, M., and Gong, N. Z. Fedredefense: Defending against model poisoning attacks for federated learning using model update reconstruction error. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Wjq2bS7fTK.

Yan, H., Zhang, W., Chen, Q., Li, X., Sun, W., Li, H., and Lin, X. RECESS vaccine for federated learning: Proactive defense against model poisoning attacks. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Zhang, H., Jia, J., Chen, J., Lin, L., and Wu, D. A3FL: adversarially adaptive backdoor attacks to federated learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a.

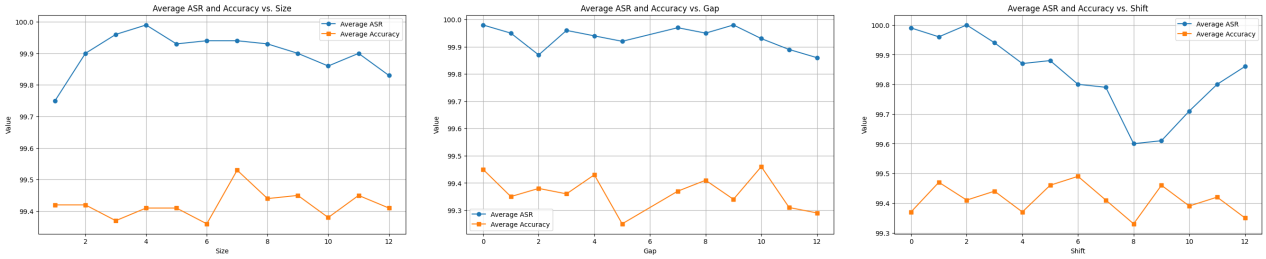
Zhang, K., Tao, G., Xu, Q., Cheng, S., An, S., Liu, Y., Feng, S., Shen, G., Chen, P., Ma, S., and Zhang, X. FLIP: A provable defense framework for backdoor mitigation in federated learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/forum?id=Xo2E217_M4n.

Zhong, A., He, H., Ren, Z., Li, N., and Li, Q. Feddar: Federated domain-aware representation learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=6P9Y25Pljl6.

Zhuang, H., Yu, M., Wang, H., Hua, Y., Li, J., and Yuan, X. Backdoor federated learning by poisoning backdoor-critical layers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=AJBGSVSTT2.

A. Appendix

Discussion of parameters. We conduct experiments to vary the number of clusters. We use K-means as the clustering algorithm. So the cluster size will automatically decided by the value of K. The results in Table 2 suggest that there is a tradeoff in choosing the value of K. When K is too small, it tends to be similar to DBA. When K is too large, it is similar to centralized attacks. We investigate the impact of the error of distance prediction on the method's effectiveness in Figure 10. Since we can directly control inaccuracies in distance prediction, we vary the topology of DFL and the number of clients. With more clients and random structures, we have observed larger errors. The following table shows



(a) Vary size (b) Vary gap (c) Vary shift

Figure 9. Effects of Local Triggers

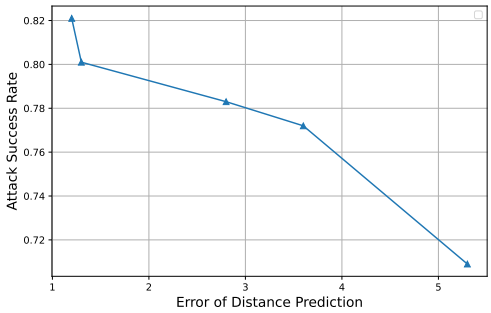


Figure 10. The Effects of Distance Prediction Error

ASR when the error is varying. We have not observed an error larger than 5.3. Even in the worst case, ASR with our method is still higher than DBA. We remark that it is unnecessary to set a threshold because our prediction will never be worse than random distribution in DBA. We also follow DBA to investigate the effects of trigger factors in the process of decomposing a global trigger. We only change one factor in each experiment shown in Figure 9. When we increase the size of the local trigger from 1 to 4, the attack success ratio will increase. At the same time, the accuracy varies slightly. However, while increasing the size from 4 to 12, the attack success ratio will drop. The value of the gap has little impact on both ASR and accuracy. This is because the relation between different local triggers has been removed by distributing the local triggers to different clients. We also note a U-shape curve of ASR when the shift increases. This is because when the trigger overlaps with some pattern in the clear image, the impact can be ignored due to overlap. However, when we further shift the trigger to the right bottom corner, the ASR will recover to a high ratio because most objects are located in the middle of the images in the dataset.

Table 3. Computational cost of our method.

Topology	Extra cost	2000 epochs of training
40 nodes	9 minutes	3 hours
80 nodes	25 minutes	9 hours
100 nodes	32 minutes	11 hours

Computational cost. In table 3, we report the running time of our algorithm and the training time of FL. We remark that the majority of the computational overhead is still the cost of training on the decentralized FL. For Cifar-100, it usually takes at least 3000 epochs to reach the convergent performance with FL. The cost of clustering can be ignored compared with training. Also, trigger distribution can be done in a few seconds. Therefore, our method can handle more complex real-world topologies and the extra computational overhead can be ignored.