SADBA: Self-Adaptive Distributed Backdoor Attack Against Federated Learning

Jun Feng[*1], **Yuzhe Lai**[*1], **Hong Sun**[†2], **Bocheng Ren**[3]

[1]Hubei Key Laboratory of Distributed System Security, Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology
[2]School of Economics, Wuhan Textile University
[3]School of Computer Science and Technology, Hainan University
junfeng@hust.edu.cn, laiyuzhe@hust.edu.cn, hsun@wtu.edu.cn, bc.revincent@gmail.com

Abstract

Backdoor attacks in federated learning (FL) face challenges such as lower attack success rates and compromised main task accuracy (MA) compared to local training. Existing methods like distributed backdoor attack (DBA) mitigate these issues by modifying malicious clients' updates and partitioning global triggers to enhance backdoor persistence and stealth. The recent full combination backdoor attack (FCBA) further improves backdoor efficiency with a full combination strategy. However, these methods are mainly applicable in small-scale FL. In large-scale FL, small trigger patterns weaken impact, and scaling them requires controlling exponentially more clients, which poses significant challenges, while simply reverting to DBA may decrease backdoor performance. To overcome these challenges, we propose the self-adaptive distributed backdoor attack (SADBA), which achieves similar performance to FCBA with a lower percentage of malicious clients (PMC). It also adapts more flexibly through an optimized model poisoning strategy and a self-adaptive data poisoning strategy. Experiments demonstrate SADBA outperforms state-of-the-art methods, achieving higher or comparable backdoor performance and MA across various datasets with limited PMC.

Introduction

Federated Learning (FL) is a distributed machine learning framework that allows multiple clients to train collaboratively without revealing their local data to protect privacy (Smith et al. 2017; Wang et al. 2024b; Wei et al. 2023). However, recent study has shown that it is vulnerable to backdoor attacks (Zhang et al. 2023; Wang et al. 2024a; Nasr et al. 2021), which aim to manipulate a subset of local training data by injecting a unique *trigger* (Wang et al. 2022) so that the model trained with poisoning data will predict target label for the data injected with *trigger* in the test period while predicting normally for the benign data (Ning et al. 2022). This dual nature makes backdoor attacks particularly insidious and challenging to detect (Dong et al. 2021; Sun et al. 2019) and mitigate (Wang et al. 2019).

Although FL is capable of aggregating dispersed and often restricted information provided by different parties to train a better model, its distributed learning methodology and inherently heterogeneous (i.e., non-i.i.d.) data distribution (Huang, Ye, and Du 2022; Cai et al. 2023) across different parties may unintentionally provide a venue for new attacks. The limitation of access to individual parties' data due to privacy concerns or regulatory constraints can facilitate backdoor attacks on the shared model trained with FL or allow modification of the local dataset to a malicious type (Feng et al. 2024; Liu et al. 2024a; Jia, Fang, and Gong 2023). For example, attackers can change the model's behavior only on specific attacker-chosen inputs via data poisoning (Gu, Dolan-Gavitt, and Garg 2017; Chen et al. 2017). However, methods that purely insert backdoors through data poisoning are inefficient. Subsequently, the model replacement-based (MR) approach (Bagdasaryan et al. 2020) was introduced, which improves backdoor performance by scaling up the malicious clients' model updates in FL, achieving the same attack success rate (ASR) with fewer malicious clients. For instance, in a picture classification task with the CIFAR dataset, an attacker who controls 1% of the participants achieves the same ASR as a data-poisoning attacker who controls 20% (Ji et al. 2018). Nonetheless, the MR method's trigger insertion strategy is a centralized static insertion, which does not fully exploit

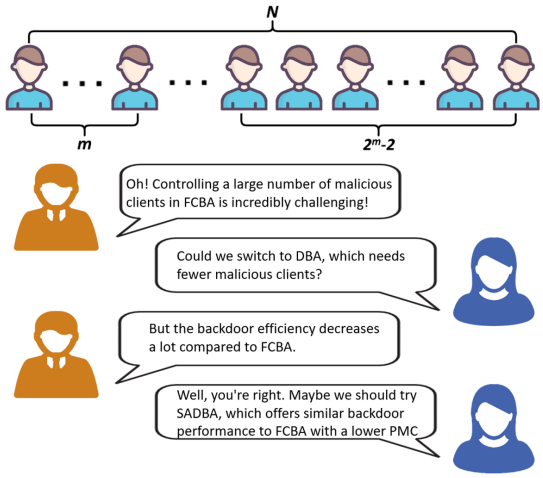

Figure 1: Background of SADBA.

[*]These authors contributed equally.
[†]Corresponding author.

Backdoor Strategy	Reducing PMC	Improving Backdoor Persistent	Improving Adaptability
MR (Bagdasaryan et al. 2020)	—	✗	✓
DBA (Xie et al. 2019)	✓	✗	✗
FCBA (Liu et al. 2024b)	✗	✓	✗
SADBA (Ours)	✓	✓	✓

Table 1: Comparison with other strategies.

the distributed learning methodology of FL, as it embeds the same global trigger pattern to all adversarial parties. To further leverage the power of FL in aggregating dispersed information from local parties to train a shared model, the distributed backdoor attack (DBA) (Xie et al. 2019) was proposed. DBA decomposes the global trigger into local patterns and embeds them in different adversarial parties, resulting in a higher ASR. However, these methods boost ASR only briefly post-poison injection, raising concerns about their long-term efficacy. To enhance the durability of backdoor attacks against dilution by benign model updates in FL. Liu et al. (2024b) propose the Full Combination Backdoor Attack (FCBA) by employing a full combination strategy to distribute the trigger partitions based on DBA to more malicious clients, thereby improving backdoor persistence. However, this also results in a substantial increase in the number of clients that the attacker needs to control. Therefore, further improvements in backdoor performance with a lower percentage of malicious clients (PMC) are necessary. Fig. 1 shows the background of our method. Here, PMC represents the proportion of malicious clients required to carry out an attack, with a lower PMC indicating a more practical and less restrictive attack setup. Backdoor persistence reflects the stability of the ASR over time, even as benign clients continue to update the model. Moreover, adaptability evaluates the attack's effectiveness across diverse federated learning environments.

To address these challenges, we propose a novel approach called Self-Adaptive Distributed Backdoor Attack (SADBA). Our method comprises two components: ❶ a novel data poisoning strategy that enhances each malicious client's backdoor contribution to the global model while maintaining a limited PMC. ❷ a model poisoning optimization strategy that addresses the issue of asynchronous local model training. In FL, some malicious local models are rarely chosen for training due to the random selection strategy, which often results in low backdoor contributions from these models, while others may have already converged and offer limited backdoor impact. By dynamically adjusting the malicious clients' local training parameters, SADBA ensures a more consistent and effective backdoor insertion across all malicious clients. By integrating these strategies, SADBA not only mitigates the exponential client control problem but also provides a robust and adaptable solution for backdoor attacks in FL, achieving higher ASR and main task accuracy (MA) across diverse datasets. Tab. 1 highlights the differences between our method and the state-of-the-art approaches (SOTAs).

Contributions. Our main contributions can be summarized as follows:

- We propose the Self-Adaptive Distributed Backdoor Attack (SADBA), an innovative approach for distributed backdoor attacks that enhances both model poisoning and data poisoning strategies. Evaluations across three image classification tasks demonstrate that SADBA achieves an average 29.7% increase in ASR while reducing PMC by 14.3% SOTAs. Moreover, in terms of scenario adaptability, SADBA, with a trigger pattern of $m = 9$, can reduce the min-PMC requirement by up to 98.2% compared to SOTAs. This advancement highlights the effectiveness of SADBA in optimizing attack efficiency and success in distributed learning environments.
- For model poisoning, we introduce a dynamic parameter-setting method to address the issue of asynchronous problem from malicious clients. For data poisoning, we employ Latin Hypercube Sampling to self-adaptively distribute triggers among malicious clients. Additionally, we leverage a backdoor gradient-based strategy for sample selection and trigger insertion to improve backdoor attack performance.
- Further experimental research shows that SADBA remains robust across various environments. Ablation studies indicate that most factors have a limited effect on the attack, and current defense mechanisms(Zhang et al. 2024a; Yao et al. 2024) are ineffective at countering it.

Related Work

Federated Learning

FL is increasingly used as a privacy-enhancing technique for distributed machine learning in a variety of applications, ranging from vision to fraud detection (Feng et al. 2020; Yan et al. 2023; Zhang et al. 2024b). The main idea behind FL is to train local models on multiple datasets hosted separately by different participants. FL can be classified into Horizontal FL (HFL) and Vertical FL (VFL) (Wang et al. 2024b). HFL involves training models collaboratively across different organizations with similar data features but different samples, while VFL involves training models across organizations with different data features but the same samples. In our study, we use the widely adopted setting of HFL with random client selection:

$$G^{t+1} = G^t + \frac{\eta}{n} \sum_{i=1}^{m} (L_i^{t+1} - G^t), \tag{1}$$

where η denotes the global learning rate and n represents the total number of clients.

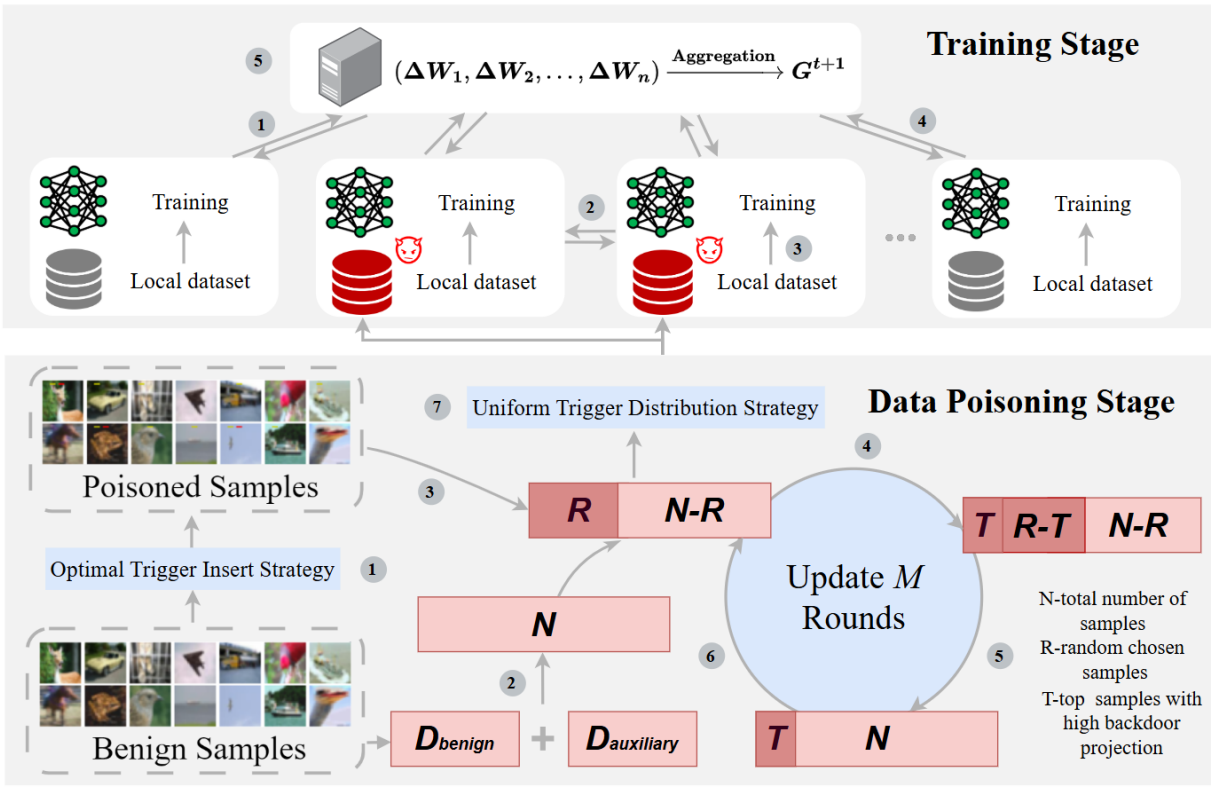

Figure 2: Overview of self-adaptive distributed backdoor attack (SADBA) in FL.

Backdoor Attack on Federated Learning

Backdoor attacks in FL can be broadly classified into model-poisoning and data-poisoning strategies. Model-Poisoning Attacks involve malicious clients manipulating their local model updates to influence the global model's behavior. Jia, Fang, and Gong (2023) introduced the SelfishAttack, which enables selfish clients to optimize their local models to bypass Byzantine-robust aggregation rules (Yin et al. 2018; Guerraoui, Rouault et al. 2018). This approach, however, relies on accessing benign model updates, limiting its effectiveness in decentralized FL scenarios. Bagdasaryan et al. (2020) proposed a model replacement (MR) strategy to enhance adversarial updates, but its success is compromised by subsequent benign updates and inconsistent contributions from malicious clients. Data-Poisoning Attacks focus on compromising local datasets to disrupt model behavior during testing. Ji et al. (2018) introduced a backdoor component insertion method, which proves less effective in FL due to the aggregation of numerous benign models. Bagdasaryan et al. (2020) used semantic backdoors in centralized models, but this approach fails to leverage FL's distributed nature. Xie et al. (2019) presented a pixel-based backdoor technique that decomposes global triggers into local partitions, yet its performance in single-shot scenarios is inadequate. Liu et al. (2024b) proposed a full combination strategy, but their approach relies on random trigger insertion and does not account for sample-specific backdoor contributions. We propose an optimal gradient-based sample selection strategy to overcome these challenges and enhance backdoor efficiency.

Strategies for Improving Backdoor Attack Robustness

In MR, a generic approach is proposed that enables the adversary to produce a model with high accuracy on both the main and backdoor tasks while evading the aggregator's anomaly detector. Following Kerckhoffs's Principle, we assume that the anomaly detection algorithm is known to the attacker. Consequently, the attacker can add an anomaly detection term L_{ano} into the objective function, as shown in Eq. (4). This term accounts for various types of anomaly detection, such as isolation forest (Xu et al. 2023), Bayesian networks, and k-means clustering (Chang et al. 2020). Shokri et al. (2020) demonstrate the effectiveness of evading defenses like activation clustering (Chen et al. 2018) and (Wang et al. 2019) by using a defense-aware term L_{ano} to generate L_{model}:

$$L_{model} = \alpha L_{ano} + (1 - \alpha) L_{class}. \tag{2}$$

The hyperparameter α is used to control the tradeoff between ASR and the anomalousness of the backdoored model for various anomaly detectors.

Self-Adaptive Distributed Backdoor Attack on Federated Learning

Threat Model

Adversary's Goal. We consider an adversary with goals similar to those of traditional backdoor attacks against machine learning models. Our focus is on the more complex Multi-Attacker Mode within a Single-shot FL setting (Ren et al. 2023, 2024). In this context, multiple malicious clients exist, and once an attacker is chosen to train in one epoch, it does not continue training for several epochs as in the Multiple-shot setting.

Adversary's capabilities. We assume that the adversary can access the training data hosted by the M ($M < N$) compromised participants based on the Kerckhoffs's theory. They can manipulate the local datasets hosted by the compromised participants, e.g., by inserting backdoor triggers or sub-trigger patterns into the local datasets of each compromised participant j. However, the adversary does not control the server; thus, they cannot directly access the global model aggregation process or the labels of the training data. Moreover, the adversary cannot access or manipulate the local models or datasets owned by the non-compromised participants.

Adversary's knowledge. The adversary can access the feature embeddings Ej generated by the local models of each compromised participant j. They can also access the global model Gt sent by server before local training in epoch t+1. Before the training stage, we assume the adversary may collect a set of auxiliary data with the similar distribution and label space to imitate the benign clients' dataset. This assumption is realistic, as, for example, the adversary can obtain additional images with the similar labels as the auxiliary training data. This approach is also utilized in label inference attacks (Fu et al. 2022).

Self-Adaptive Distributed Backdoor Attack

General Framework. Fig. 2 provides an overview of our method. During the data poisoning stage, we aim to distribute the trigger pattern m uniformly among malicious clients. Besides, we employ a self-adaptive trigger positioning and sample-specific poisoning strategy to enhance each malicious client's backdoor contribution. In the training stage, we utilize a dynamic hyper-parameter adjustment strategy to address the asynchronous problem among malicious clients. Specifically, before training, ① the attacker will execute an optimal trigger insertion strategy for each sample during the offline stage to generate an auxiliary poisoning dataset. To mimic the benign clients' dataset in real FL, ② we append an auxiliary dataset with similar sample labels. After preparation, ③ we randomly replace R clean samples with corresponding poisoned samples from the poisoning dataset. Then, ④ we select T poisoned samples with the highest backdoor projection and ⑤⑥ revert the remaining R-T samples to their benign state for M rounds. Finally, ⑦ we use a Latin Hypercube Sampling based trigger distribution strategy to ensure that each type of trigger is uniformly distributed among malicious clients. In the training stage, at round $t+1$, ❶ the server first distributes the previous round's global model to the local clients. If there are K malicious clients chosen for training, ❷❸ they will communicate with each other to determine their local training speed for local training. ❹❺ Then the K malicious clients submit the scaled local model updates to the server for aggregation.

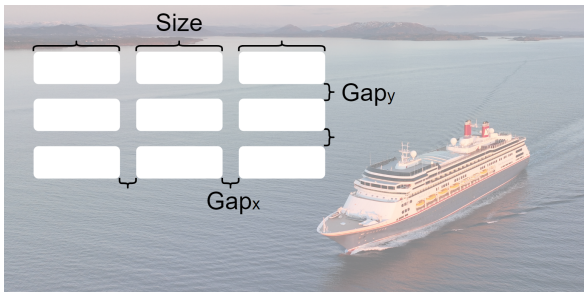

Figure 3: Trigger shape factors (size and gap) in backdoored images.

Data Poisoning Strategy. For the trigger type in our backdoor attack, we also use pixel blocks as DBA and FCBA do. However, unlike DBA and FCBA, which employ a static trigger insertion strategy by placing the trigger in the upper left corner and using a random sample selection method, we implement an optimal trigger insertion strategy and consider the differences of each sample's contribution towards backdoor attack to improve ASR. Finally, we use a uniform trigger distribution method to overcome the constraints of PMC. The main process of data poisoning is depicted in Fig. 2.

- **Optimal Trigger Insert Strategy.** Different from the static trigger insertion strategies used by DBA and FCBA, SADBA employs a self-adaptive trigger insertion strategy. This approach determines the optimal trigger position that enhances the similarities between poisoned feature embedding and that of the target data. By making these features as close as possible, the classifier is misled into misclassifying the backdoor data as the target data, thereby creating a false association between the trigger and the target label. The objective of finding the optimal trigger insertion position is given as follows:

$$B^* = \arg\min_{B_i} \sum_j \left\| B_i(x_i, \phi) - x_j^t \right\|_{fro}^2. \quad (3)$$

  For sample x, with the global trigger ϕ, x^t represents the target sample. The shape of global trigger is determined by two factors, Trigger Size and Trigger Gap as illustrated in Fig. 3. We aim to find the best trigger insert position for x by computing its distance to the target sample. We include $\|\cdot\|_{fro}$ the Frobenius norm, in our calculations. Finally, every optimally trigger inserted sample is stored in D_{adv}^* to prepare for the sample selection strategy.
- **Sample Selection Strategy.** Unlike FCBA, which randomly chooses samples for trigger insertion, we consider each sample's backdoor contribution by comparing its backdoor gradient's projection with the average

backdoor gradient. Traditional backdoor sample selection often uses the Euclidean distance of each sample's backdoor gradient. However, this approach does not consider the direction of the gradients. Specifically, in Fig. 4, there are two samples with the same L2 norm but inconsistent contributions to the backdoor gradient direction. Therefore, we use the projection of each sample's backdoor gradient with the average backdoor gradient to measure each sample's contribution. In real FL, many benign clients participate in training, so we use auxiliary data with similar labels to mimic the benign clients' dataset. Our sample selection strategy process is detailed in Fig. 2. Eq. (4) shows the projection of a sample's backdoor gradient onto the average backdoor gradient of the R samples.

$$\begin{aligned} p_i(B^*(x_i,\phi)) &= \mathrm{E}_\theta\left[\frac{g_i(B^*(x_i,\phi))\cdot\overrightarrow{OS}}{\left\|\overrightarrow{OS}\right\|_2}\right] \\ \text{s.t.}\overrightarrow{OS} &= \frac{1}{R}\sum_{j=1}^{R} g_j(B^*(x_j,\phi)). \end{aligned} \tag{4}$$

- **Trigger Distribution Strategy.** To overcome the limitations of the PMC constraint, we adopt a uniform trigger distribution strategy. Specifically, for a given trigger pattern m, we no longer maintain the fixed relationship between malicious clients and the number of local triggers generated by combining the trigger pattern. Instead, we use a self-adaptive approach that adjusts to a given PMC ($PMC_{DBA} \leq PMC \leq PMC_{FCBA}$) by employing Latin Hypercube Sampling to uniformly sample the trigger combinations. This method eliminates the need for a fixed PMC to match a specific trigger pattern, allowing for self-adaptive adjustment of the PMC based on your attack capabilities. Thus reducing the constraints imposed on PMC in SADBA. The *min*-PMC for each attack is illustrated in Tab. 3.

Model Poisoning Strategy. In FCBA and DBA, the model replacement(MR) method is used to enhance the influence of backdoor updates on the global model. Therefore, we primarily focus on optimizing MR. In MR, the scale parameter is set statically based on the server learning rate η and the total number n of clients for a task. By setting the scale parameter as the estimated n/η, at round t+1, it leads to the final aggregated global model being almost replaced by the average adversarial model $\overline{X}$ when the global model has converged, as shown in Eq. (5).

$$\begin{aligned} G^{t+1} &= G^t + \frac{n}{\eta}\left[\sum_{i=1}^{m-k}(L_i^{t+1}-G^t) + \frac{1}{k}\sum_{j=1}^{k}\gamma(X_j-G^t)\right] \\ &\approx G^t + \frac{n}{\eta}\cdot\frac{1}{k}\sum_{j=1}^{k}\gamma(X_j-G^t) \\ &\approx G^t + \frac{1}{k}\cdot\sum_{j=1}^{k}X_j - G^t \\ &= \overline{X}, \end{aligned} \tag{5}$$

where X is the malicious model, L is benign client's updated model, k is the number of malicious clients chosen to train in epoch t+1, m is the number of clients chosen to train in each epoch, and γ is the scale parameter. However,in MR, each malicious clients' local learning rate lr and the number of local training rounds E are the same. When several malicious clients collaboratively train in FL, due to the random selection setting, some malicious clients' models may hardly converged for several rounds, leading to a lower contribution towards the backdoor, while some malicious models may have already converged with limited backdoor contribution. In order to mitigate the asynchronous problem among malicious clients, we propose a dynamic training synchronization method to improve each malicious client's backdoor contribution.

- **Training Synchronization Optimization.** We use a self-adaptive strategy to make each malicious client's training situation as consistent as possible by dynamically adjusting the local learning rate (lr) and learning rounds (E) according to its training status compared with other malicious clients. If its local backdoor loss with the local dataset is less than the average, we will increase its local learning rate lr and learning rounds E to let it catch up with the average level. However, blindly improving r and E is not applicable as we use the single-shot strategy. DBA's experiments show that too much backdoor effect

Algorithm 1: Optimized Local Training Process

Initialize malicious model X and loss function l
$X \leftarrow G^t$
$l = \alpha L_{ano} + (1-\alpha)L_{class}$
Malicious clients exchange their latest backdoor average loss $L_{class}(X, D_{adv})$.
if $L_{class}(X, D_{adv}) < \frac{1}{M}\sum_j L_{class}(X_j, D_{adv})$ and $L_{class}(G^t, D) < \varepsilon$ **then**
  // properly increase E_{adv} and lr_{adv}
  $E_{adv} \leftarrow \mu_1 \cdot E_{adv}$
  $lr_{adv} \leftarrow \mu_2 \cdot lr_{adv}$
  $r_{step} \leftarrow 1$
end if
for epoch $e \in E_{adv}$ **do**
  if $L_{class}(X, D_{adv}) < \varepsilon'$ **then**
    // if model converges, then stop
    break
  end if
  for batch $b \in D_{c\ln}$ **do**
    // replace batch b_{cln} with b_{adv} malicious samples
    $b \leftarrow replace(b_{adv}, b, D_{adv})$
    $X \leftarrow X - lr_{adv} \cdot \nabla l(X, b)$
  end for
  if *epoch* $e \in step\_sched$ **then**
    // decrease learning rate
    $lr_{adv} \leftarrow lr_{adv}/r_{step}$
  end if
end for
// scale up the model by γ.
$L_{adv}^{t+1} \leftarrow \frac{1}{k}\cdot\gamma(X - G^t) + G^t$
return L_{adv}^{t+1}

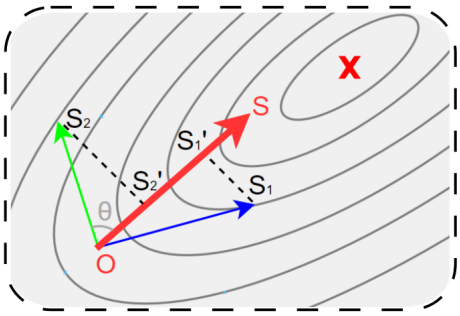

Figure 4: Comparison of two backdoor gradients $\overrightarrow{OS_1}$, $\overrightarrow{OS_2}$ has the same L2 norm but different directional contributions relative to the average backdoor gradient $\overrightarrow{OS}$.

Dataset	Labels	Image Size	Training/Test Images	Model
MNIST	10	1*28*28	60000/10000	2Conc + 2fc
Fashion-MNIST	10	1*28*28	60000/10000	2Conc + 2fc
CIFAR-10	10	3*32*32	50000/10000	Resnet-18

Table 2: Dataset and model architecture.

in the early stage will cause the global model to fail in MA, which may trigger alertness from anomaly detection (Liu et al. 2023), decreasing attack's stealthiness. Therefore, we mainly increase the lr and E with slow convergence speed after the global model has converged. Besides, to evade inspection from anomaly detection, we also add an anomaly detection term L_{ano} into each malicious client's local model to improve our attack's robustness. The detailed process of our method is shown in Algorithm 1.

Experiments

This section details the implementation and evaluation of our method. We outline the experimental setup and describe the main evaluation metrics. Then we compare the backdoor performance of SADBA with SOTAs across three tasks. Finally, our analysis and experiments demonstrate that SADBA is effective in enhancing both backdoor performance and adaptability.

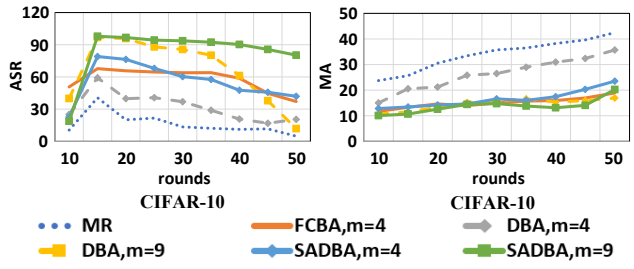

Figure 5: ASR and MA in large-scale FL.

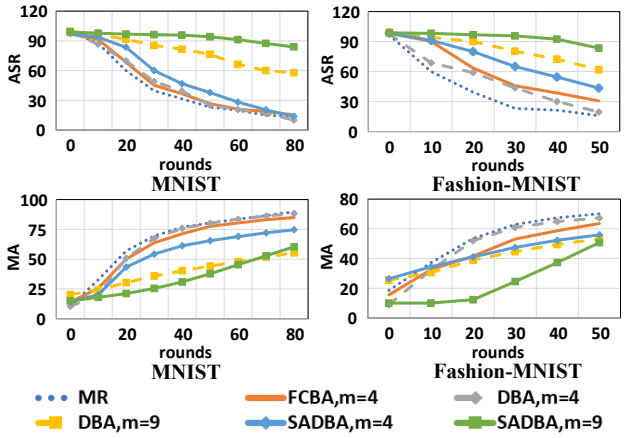

Figure 6: ASR and MA in small-scale FL.

Experiment Setup

Datasets & Model Architecture. Our experiments are conducted on three classification datasets: MNIST, Fashion MNIST and CIFAR-10. The dataset details and the model architectures utilized are summarized in Tab. 2.

Training Setting. For the three image classification tasks, we assess backdoor performance under a small scale FL setting with 100 clients and a large scale FL setting with 500 clients. Continuing the examination of FCBA's backdoor performance in more realistic scenarios, our experiment was conducted using single-shot attacks. This approach implies that once selected, malicious clients, akin to benign clients, partake in only one round of training and are not repeatedly chosen in successive rounds, as is the case with multiple-shot attacks. The global learning rate η is set to 0.1 for all tasks. We utilize *Stochastic Gradient Descent* (SGD) as the optimizer. During training, each client trains for E epochs with a specific local learning rate lr and a batch size of 64. In each round, we select 10 clients out of 100 clients, to submit their local model updates for aggregation. We consider a more realistic scenario where the selection of clients in each round is randomized. The target labels for the backdoor attacks are "7" in MNIST, "Bird" in CIFAR-10, and "T-shirt/top" in Fashion-MNIST.

Main Evaluation Metrics

Percentage of Malicious Clients (PMC). PMC is the proportion of malicious clients in federated learning. We use *min*-PMC to represent the minimum proportion of malicious clients required to execute the backdoor attack. If the PMC is less than the *min*-PMC, the attack cannot be deployed in that environment. A smaller *min*-PMC indicates stronger adaptability and robustness of the attack. The *min*-PMC for each attack is presented in Tab. 3.

Attack Success Rate (ASR). ASR indicates the percentage of adversarial examples that successfully cause the model to make incorrect predictions.

Attack	Trigger Parttern (m)	Number of Clients (NC)					
		100	200	400	600	800	1000
Baseline (MR)	1	1	0.5	0.25	0.167	0.125	0.1
	4	–	–	–	–	–	–
	9	–	–	–	–	–	–
DBA	1	–	–	–	–	–	–
	4	4	2	1	0.67	0.5	0.4
	9	9	4.5	2.25	1.5	1.125	0.9
FCBA	1	–	–	–	–	–	–
	4	14	7	3.5	2.34	1.75	1.4
	9	**510**	**255**	**127.5**	**85**	**63.7**	**51**
SADBA (Ours)	1	–	–	–	–	–	–
	4	1	0.5	0.25	0.167	0.125	0.1
	9	1	0.5	0.25	0.167	0.125	0.1

Table 3: Comparison of the *min*-PMC (%) for each Attack.

Main Task Accuracy (MA). MA is the percentage of correctly classified samples in the primary classification task, focusing on the model's accuracy on benign, trigger-free samples.

Comparisons between SADBA and SOTAs

In this section, we evaluate the backdoor performance of these attacks under different FL setting. We use the ASR curve trend to illustrate attack persistence after 0 rounds post-poison injection.

Fig. 5 illustrates the attack's persistence in large-scale FL with 400 clients using the CIFAR-10 dataset under a non-iid setting. From the figure, we observe that **in large-scale FL, a more complex trigger pattern can enhance backdoor performance.** This enhancement can be attributed to the fact that a more intricate trigger pattern generally results in a larger PMC, leading to a greater impact of the backdoor impact in FL.

However, blindly increase trigger patterns is not accessible. It often requires more malicious clients to cooperate in the attack, particularly in FCBA, In Tab. 3, the trigger pattern m denotes the number of sub-triggers derived from the global trigger, with $m = 1$ indicating a single pixel block trigger. In MR, only a single centralized trigger is used, while other distributed attacks require multiple patterns to enhance their backdoor effectiveness. Consequently, using a centralized pattern is infeasible for DBA, FCBA, and SADBA. The table also reveals that when $m = 9$, due to the exponential nature of the binomial theorem used to determine full complete combinations, the *min*-PMC expands to 510% when NC is 100, 255% when NC is 200 and 127.5% when NC is 400. This indicates that the number of malicious clients required by FCBA would exceeds the total number of clients in such scenarios. In large-scale FL with an NC of 1000, the *min*-PMC for FCBA remains over 50%, making FCBA impractical to achieve such an attack in real-world scenarios. Even though FCBA could potentially reduce the trigger pattern to 4, the backdoor performance, as shown in Fig. 5, is less effective.

Fig. 6 shows the attack's performance in terms of ASR in small-scale FL. It indicates that **SADBA with a smaller trigger pattern still has a similar performance with FCBA.** In the case of Fashion-MNIST, SADBA with a trigger pattern of 4 even surpasses FCBA in terms of attack persistent, achieving an ASR of 49.08% after 50 rounds, compared to FCBA's 30.83%. For MA, Fig. 6 illustrates that although MA dips during backdoor injection, it rebounds with additional rounds.

Combined with the attack performance in both smallsclae FL and large-scale FL, **SADBA can achieve a higher ASR than DBA in large scale while achieve a similar performance of FCBA in small-scale FL with a stable MA.**

Conclusions and Future Work

In this paper, we propose SADBA, the first adaptive distributed backdoor attack. Extensive experiments on three image classification tasks demonstrate that SADBA effectively adapts to diverse environment settings, particularly in large-scale FL scenarios. Our results indicate that SADBA surpasses most SOTAs in terms of attack success rate and backdoor persistence while requiring a lower PMC. Furthermore, our ablation analysis on the crucial factors influencing SADBA indicates that it maintains stability across various conditions. SADBA's robustness extends to non-i.i.d. data distributions and demonstrates resistance to existing backdoor defense mechanisms in practice. These findings provide valuable insights for threat assessment and contribute to the evaluation of adversarial robustness in FL.

While SADBA offers enhanced adaptability and stable backdoor performance, certain limitations remain. For example, its adaptability in some specific non-i.i.d. FL settings, such as the Dirichlet distribution with α=0.1 is not fully optimized, which compromises its generalizability in real-world environments. Consequently, developing a robust, distributed backdoor attack for FL is an important area for future research.

Acknowledgments

This research was supported by the National Key R&D Program of China (No. 2024YFB3108700), the National Natural Science Foundation of China (No. 62372195, 62462054, and 62166047), the Natural Science Foundation Project of Jiangxi Province, China (No. 20232BAB202007), and CCF-Huawei Populus Grove Fund.

References

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS'20)*, 2938–2948.

Cai, L.; Chen, N.; Cao, Y.; He, J.; and Li, Y. 2023. FedCE: personalized federated learning method based on clustering ensembles. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM'23)*, 1625–1633.

Chang, Y.; Tu, Z.; Xie, W.; and Yuan, J. 2020. Clustering driven deep autoencoder for video anomaly detection. In *European Conference on Computer Vision (ECCV'20)*, 329–345. Springer.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Dong, Y.; Yang, X.; Deng, Z.; Pang, T.; Xiao, Z.; Su, H.; and Zhu, J. 2021. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR'21)*, 16482–16491.

Feng, J.; Yang, L. T.; Ren, B.; Zou, D.; Dong, M.; and Zhang, S. 2024. Tensor recurrent neural network with differential privacy. *IEEE Trans. Computers*, 73(3): 683–693.

Feng, J.; Yang, L. T.; Zhu, Q.; and Choo, K.-K. R. 2020. Privacy-preserving tensor decomposition over encrypted data in a federated cloud environment. *IEEE Trans. Dependable and Secure Computing*, 17(4): 857–868.

Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Zhou, J.; Liu, A. X.; and Wang, T. 2022. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium (USENIX Security'22)*, 1397–1414.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Guerraoui, R.; Rouault, S.; et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning (ICML'18)*, 3521–3530. PMLR.

Huang, W.; Ye, M.; and Du, B. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 10143–10153.

Ji, Y.; Zhang, X.; Ji, S.; Luo, X.; and Wang, T. 2018. Model-reuse attacks on deep learning systems. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*, 349–363.

Jia, Y.; Fang, M.; and Gong, N. Z. 2023. Competitive advantage attacks to decentralized federated learning. *arXiv preprint arXiv:2310.13862*.

Liu, H.; Ming, Y.; Wang, C.; and Zhao, Y. 2024a. Flexible selective data sharing with fine-grained erasure in VANETs. *IEEE Trans. Information Forensics and Security*, 19: 9582–9597.

Liu, T.; Zhang, Y.; Feng, Z.; Yang, Z.; Xu, C.; Man, D.; and Yang, W. 2024b. Beyond traditional threats: A persistent backdoor attack on federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'24)*, volume 38, 21359–21367.

Liu, Z.; Zhou, Y.; Xu, Y.; and Wang, Z. 2023. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, 20402–20411.

Nasr, M.; Songi, S.; Thakurta, A.; Papernot, N.; and Carlin, N. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on Security and Privacy (SP'21)*, 866–882.

Ning, R.; Li, J.; Xin, C.; Wu, H.; and Wang, C. 2022. Hibernated backdoor: A mutual information empowered backdoor attack to deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'22)*, volume 36, 10309–10318.

Ren, B.; Yang, L. T.; Nie, X.; Feng, J.; Deng, X.; and Zhu, C. 2024. Zero-shot fault diagnosis for smart process manufacturing via tensor prototype alignment. *IEEE Trans. Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2024.3350715*.

Ren, B.; Yang, L. T.; Zhang, Q.; Feng, J.; and Nie, X. 2023. Tensor-empowered adaptive learning for few-shot streaming tasks. *IEEE Trans. Neural Networks and Learning Systems*, 34(10): 6861–6871.

Shokri, R.; et al. 2020. Bypassing backdoor detection algorithms in deep learning. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P'20)*, 175–183.

Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. S. 2017. Federated multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS'17)*, 30.

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

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP'19)*, 707–723.

Wang, X.; Pan, H.; Zhang, H.; Li, M.; Hu, S.; Zhou, Z.; Xue, L.; Guo, P.; Wang, Y.; Wan, W.; et al. 2024a. TrojanRobot: Backdoor attacks against robotic manipulation in the physical world. *arXiv preprint arXiv:2411.11683*.

Wang, X.; Wang, S.; Li, Y.; Fan, F.; Li, S.; and Lin, X. 2024b. Differentially private and heterogeneity-robust federated learning with theoretical guarantee. *IEEE Trans. Artificial Intelligence*, 5(12): 6369–6384.

Wang, Y.; Zhao, M.; Li, S.; Yuan, X.; and Ni, W. 2022. Dispersed pixel perturbation-based imperceptible backdoor trigger for image classifier models. *IEEE Trans. Information Forensics and Security*, 17: 3091–3106.

Wei, K.; Li, J.; Ma, C.; Ding, M.; Chen, W.; Wu, J.; Tao, M.; and Poor, H. V. 2023. Personalized federated learning with differential privacy and convergence guarantee. *IEEE Trans. Information Forensics and Security*.

Xie, C.; Huang, K.; Chen, P.-Y.; and Li, B. 2019. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations (ICLR'19)*.

Xu, H.; Pang, G.; Wang, Y.; and Wang, Y. 2023. Deep isolation forest for anomaly detection. *IEEE Trans. Knowledge and Data Engineering*, 35(12): 12591–12604.

Yan, X.; Miao, Y.; Li, X.; Choo, K.-K. R.; Meng, X.; and Deng, R. H. 2023. Privacy-preserving asynchronous federated learning framework in distributed IOT. *IEEE Internet of Things J.*, 10(15): 13281–13291.

Yao, Z.; Zhang, H.; Guo, Y.; Tian, X.; Peng, W.; Zou, Y.; Zhang, L. Y.; and Chen, C. 2024. Reverse backdoor distillation: Towards online backdoor attack detection for deep neural network models. *IEEE Trans. Dependable and Secure Computing*.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning (ICML'18)*, 5650–5659.

Zhang, H.; Hu, S.; Wang, Y.; Zhang, L. Y.; Zhou, Z.; Wang, X.; Zhang, Y.; and Chen, C. 2024a. Detector collapse: Backdooring object detection to catastrophic overload or blindness. *arXiv preprint arXiv:2404.11357*.

Zhang, H.; Yao, Z.; Zhang, L. Y.; Hu, S.; Chen, C.; Liew, A.; and Li, Z. 2023. Denial-of-service or fine-grained control: Towards flexible model poisoning attacks on federated learning. *arXiv preprint arXiv:2304.10783*.

Zhang, Y.; Miao, Y.; Li, X.; Wei, L.; Liu, Z.; Choo, K.-K. R.; and Deng, R. H. 2024b. Efficient privacy-preserving federated learning with improved compressed sensing. *IEEE Trans. Industrial Informatics*, 20(3): 3316–3326.