Model Backdoor Attack on Federated Learning Based on Parameter Analysis

Bo Wang, *Member, IEEE*, Yiming Yan, Maozhen Zhang, Wei Wang, *Member, IEEE*, and Hongwei Yao

***Abstract*—With the increasingly widespread application of federated learning (FL) in various fields, the issue of backdoor attacks against FL has garnered significant attention from both academia and industry. While there has been some progress in researching backdoor attacks against FL, data backdoor attacks are easily mitigated in federated environments, and model backdoor attacks are susceptible to detection by defense mechanisms. Therefore, we propose a refined backdoor attack method tailored for FL under the image classification task. Our method involves the collaborative operation of three key modules. Firstly, the parameter importance analysis module identifies parameters with minimal impact on model performance, and creates parameter importance masks to provide precise targets for subsequent operations. Subsequently, the activation difference computation module calculates the activation differences between backdoor samples and benign samples to locate trigger-sensitive parameters. Our method implants the backdoor by flipping and zeroing the precisely located layer parameters, while maintaining the model's classification performance on benign samples. Experimental results show that our method is feasible to achieve an average attack success rate more than 99% across the three victim models. This demonstrates the effectiveness of our method in FL environments and its robustness against various FL defense mechanisms.**

***Index Terms*—Federated learning, backdoor attacks, parameter analysis, parameter manipulation.**

I. Introduction

DEEP neural networks have achieved great success in various fields such as computer vision [1], speech recognition [2] and natural language processing [3]. However, it also raises privacy concerns as the models may reveal sensitive information about users. In recent years, federated learning (FL) [4] has attracted much attention for training machine learning models without accessing private data. However, the distributed learning framework of FL may also be subject to malicious attacks when aggregating gradient information from different sources to generate global models [5], [6]. Especially, privacy concerns [7] or regulatory restrictions may facilitate backdoor attacks on shared models trained using FL.

Currently, research on backdoor attacks in FL have made some progress, but still face numerous challenges in practical scenarios. In data backdoor attacks within FL, an attacker aims to implant a backdoor into the global model by poisoning the local data of malicious clients, without manipulating their local training processes. However, these backdoors are often mitigated by the normal updates contributed by other clients during the global model aggregation process, resulting in a reduced success rate of such attacks [8]. In federated learning model backdoor attacks, the attackers not only poison the local data of malicious clients but also manipulate their local training processes or directly modify local model parameters. Existing methods for model backdoor attacks in FL often rely on indiscriminate amplification of malicious updates to ensure dominance in the global model [9], [10], [11]. However, this strategy may cause significant deviations in model updates from the normal range. This increases the risk of detection by defense mechanisms such as pruning [12], [13] or anomaly detection [14]. Additionally, it can impair the predictive performance of the model on the main task, thereby reducing the stealthiness and practicality of the malicious model.

Based on above observation, we pay attention to the fine-grained manipulation of malicious updates with the aim of consistently maintaining backdoor properties within the global model. Addressing the weaknesses of existing model backdoor attacks against defense mechanisms, we propose a federated learning model backdoor attack method based on parameter analysis and activation difference. Specifically, the method comprises three modules: parameter importance analysis, activation difference computation, and parameter manipulation flipping. Firstly, the parameter importance analysis module aims to identify a set of candidate parameters with minimal impact on task performance from model parameters. These parameters are ideal manipulation targets as their modification minimally affects the normal operation of the model, thus significantly reducing the risk of detection by defense mechanisms. Subsequently, the activation difference computation module computes the activation features of backdoor images and clean images separately during model training, obtaining the activation difference

Received 2 December 2024; revised 3 January 2026; accepted 11 February 2026. Date of publication 16 February 2026; date of current version 12 May 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62372452, in part by the Application Fundamental Research Project of Liaoning Province under Grant 2025JH2/101330123, and in part by the Open Research Fund of the National Key Laboratory of Cyber Space Behavior Cognition Technology under Grant CBCT2024KF02. *(Corresponding author: Wei Wang.)*

Bo Wang, Yiming Yan, and Maozhen Zhang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: bowang@dlut.edu.cn; yiming@mail.dlut.edu.cn; maozhenzhang@mail.dlut.edu.cn).

Wei Wang is with the State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS) and New Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China (e-mail: wwang@nlpr.ia.ac.cn).

Hongwei Yao is with the State Key Laboratory of Blockchain and Data Security, Zhejiang University, Zhejiang 310058, China (e-mail: yhongwei@zju.edu.cn).

Digital Object Identifier 10.1109/TDSC.2026.3664908

matrix introduced by the backdoor trigger. Finally, the parameter manipulation flipping module flips or zeros the corresponding parameters based on the mapping relationship between activation difference and candidate parameters to ensure effective triggering of the expected backdoor behavior without compromising the main task performance. This method successfully enhances the stealthiness of attacks without impairing main task performance, posing a challenge to the security of FL systems and driving further research into related defense strategies.

The main contributions of this paper are as follows:

- In this paper, we realize a federated learning model backdoor attack for image classification tasks by manipulating the model parameters. Our method achieves the stable existence of the backdoor in the global model without causing the model update to deviate from the normal range significantly.
- We propose a federated learning model backdoor attack method based on parameter analysis and activation difference. It achieves fine-grained manipulation of model parameters through the coordinated cooperation of three modules: parameter importance analysis, activation difference computation, and parameter manipulation flipping, and completes the covert poisoning of the model.
- Extensive experiments conducted on three datasets and seven federated learning backdoor defense methods validate the effectiveness of our approach and its robustness against defense methods. In the full range of defense scenarios, our method achieves an attack success rate that is up to 80% higher than the baseline methods.

II. Related Works

A. Federated Learning Architecture

Federated learning [4], as an emerging paradigm of distributed learning, aims to collaboratively train a high-performance global model across multiple devices or organizations while preserving the privacy of local data. Federated learning comprises three core components: clients, a central server, and a communication system. Each client, whether a smartphone, medical device, or bank branch, retains its unique dataset and independently trains a local model. Subsequently, updates of these local models are sent to the central server, which coordinates the entire federated learning process by aggregating model updates [15], [16], [17] from various clients to optimize the global model. A highly encrypted and secure communication system during this process ensures the secure transmission of model updates between clients and the central server, thereby safeguarding the privacy of data and models. Based on the differences between the data samples of participating users and their characteristics, federated learning can be categorized into horizontal federated learning [18], vertical federated learning [19] and federated transfer learning [20]. Among them, this paper focuses on the implementation of backdoor attacks and their impact on model security in a federated learning scenario, and the mathematical process of federated learning is described in Subsection III-B.

B. Backdoor Attacks in Federated Learning

Although federated learning effectively addresses the data silos and data privacy issues in traditional centralized learning [21], its inherent openness and distributed nature introduce new security challenges, among which backdoor attacks are particularly prominent. Specifically, backdoor attacks against federated learning can be equally categorized into data backdoor attacks and model backdoor attacks.

1) Federated Learning Data Backdoor Attacks: In data backdoor attacks, attackers implant data samples containing backdoors in the local training data. These samples are carefully designed to survive the global model aggregation phase of federated learning and subsequently trigger predefined backdoor behaviors in the aggregated global model. For example, Tolpegin et al. [22] demonstrated a method for targeted attacks on federated learning systems through label flipping. In this attack, instead of changing the features of the local samples, attackers modify the labels of samples to other incorrect labels. This approach can mislead the global model without directly manipulating features and cause the model to behave abnormally under specific inputs. Bhagoji et al. [10] not only implant backdoors into the local training data but also cleverly adjust the gradient to amplify the backdoor effect, making the global model more susceptible to misclassification when encountering well-designed poisoned samples. Xie et al. [23] employed a more sophisticated distributed backdoor attack strategy by dispersing backdoors among multiple malicious participants, enhancing the stealthiness of the attack and demonstrating its persistence in the global model. Wang et al. [24] effectively overcame the sample label consistency problem ignored by existing methods in the pursuit of trigger concealment by combining re-parameterized noise triggers and deep learning image steganography techniques. However, data backdoor attacks only inject backdoors into the training dataset, while models trained using normal gradient descent algorithms struggle to maintain long-term attack effects after global aggregation.

2) Federated Learning Model Backdoor Attacks: Federated learning model backdoor attacks rely mainly on the dynamic changes of model parameters during updating, rather than solely on the static characteristics of a single dataset. By carefully selecting model parameters that are difficult to be overwritten or updated by other clients, attackers are able to ensure that the backdoor lurks in the model for a long period of time. Even if the data distribution changes, the backdoor attack can continue to work. Compared with the data poisoning backdoor attack during the model training phase, the model backdoor attack in the aggregation phase exhibits the greater attack strength and optimal stealth.

Bagdasaryan et al. [25] meticulously designed a method that involves special scaling of locally trained models containing backdoors, ensuring that these malicious updates dominate during global model aggregation, thereby manipulating the global model to align with the backdoor model. Fung et al. [14] introduced a novel model poisoning strategy, where attackers collaboratively upload model parameters containing backdoors through collusion with multiple malicious participants. Wang et al. [8] proposed a tail-sample-based backdoor attack strategy

focused on selecting and manipulating samples which appear infrequently in the dataset. By coordinating with multiple malicious clients, these tail backdoor samples are cleverly integrated during the model update process, effectively controlling the aggregated model. The Neurotoxin proposed by Zhang et al. [26] projects malicious gradients onto the unused parameter space of benign clients. This enables the attack to remain in the global model even after the poisoned data upload is stopped, significantly enhancing the stability of the attack. Fang et al. [27] proposed a new method called Focused Flip Backdoor Attack (F3BA). The attacker flips small parameters in the model and optimizes triggers to prevent excessive model updates. Then, the attacker uses the optimized triggers to poison the local data and retrain the model to maintain its normal performance. Lyu et al. [28] proposed CerP, which assumes that a regular adversary can compromise multiple clients and launch coordinated attacks. CerP achieves backdoor attacks by jointly optimizing three objectives. Firstly, optimize the trigger to maximize the accuracy of the model on backdoor tasks, while limiting its size to avoid detection. Secondly, minimize the difference between the poisoned model update and the benign model update. Finally, the alliance among malicious clients is utilized to suppress the high similarity between the updates of the poisoned model.

However, existing federated learning backdoor attacks often rely on undifferentiated amplification of malicious updates. These strategies may cause model updates to deviate significantly from the normal range and impair the predictive performance of the model on the main task.

C. Defense Methods Against Backdoors in Federated Learning

In this section, we elaborate on the defense strategies and specific implementation methods against backdoors in federated learning, focusing on both data and model aspects. At the data level, the defense strategies emphasize the development of effective data filtering mechanisms to identify and exclude malicious samples, while ensuring data diversity and model performance remain unaffected. Model defenses aim to enhance the robustness of the aggregation process by introducing advanced anomaly detection methods and improving the aggregation protocol to prevent malicious model updates from harming the global model.

1) Data Defense Methods Against Backdoors: Data defense methods focus on filtering input datasets to remove potential backdoor patterns. Various input filters are utilized to identify and clear possible malicious samples while preserving data diversity and model performance.

For example, Nguyen et al. [29] proposed a defense strategy based on the Dropout mechanism during the SGD process to identify and suppress the impact of backdoor attacks by monitoring the model's performance on both normal and backdoor inputs. When the model performance approaches a predefined turning point, the weight of the gradient which may be affected by the backdoor attack is adjusted and set to zero to block the backdoor attack propagation path. Hou et al. [30] developed a defense algorithm based on federated learning filters and fuzzy label flipping strategy. This algorithm constructs and maintains a filter library on the central server to store and identify common backdoor types.

2) Model Defense Methods Against Backdoors: Taking measures to weaken or eliminate malicious activities before model update aggregation is crucial in defending against backdoor attacks. Limiting the size of updates uploaded by clients or applying differential privacy techniques can effectively reduce the impact of malicious updates on the global model. Reference [31] distinguished benign and malicious updates by projecting local model updates into a low-dimensional space and using an encoder-decoder model. This method highlights the differences between benign and malicious updates, enhancing the accuracy of malicious update detection. Gao et al. [32] achieved selective zeroing of local features in uploaded updates by adjusting aggregation protocols, such as the PartFedAvg protocol, to reduce the impact of data poisoning attacks.

During the model aggregation phase, a series of defense strategies can also be adopted. These strategies primarily focus on strengthening the robustness of aggregation algorithms to resist the potential impacts of abnormal updates on the global model. Reference [33] adaptively adjusted learning rates by analyzing the sign function distribution of each model feature and setting hyperparameters to determine learning rate thresholds, enhancing the robustness of the federated learning framework. Wan and Chen [34] trained an attention-based neural network to collect model updates under different types of backdoor attacks in a simulated federated learning task, learning the potential vulnerabilities of the global model to backdoor attacks in a self-supervised manner to enhance defense effectiveness.

Lin et al. [35] proposed a distillation framework for robust federated model fusion. During the server update process, the global model is used as the student model for distillation, and all client models are used as teacher models to guide the next round of global model updates. Sturluson et al. [36] assigned a median score to each client based on FedDF to measure the frequency with which each client's output logits became the median of class predictions. Wu et al. [37] proposed the FedMV method, which calculates and sorts the average activation values of all filters in the last convolutional layer of each local model on the test samples. After the server collects the local rankings of all clients, it determines the global pruning standard by calculating the average ranking and finally removes the filters with higher average rankings in this convolutional layer. Rieger et al. [38] proposed DeepSight, which clusters clients by using two different metrics and removes clusters identified as malicious clients that exceed the threshold. Yang et al. [39] proposed a defense method called RoseAgg. By aggregating multiple updates that exhibit high similarity into a single update, principal component analysis is used to extract benign principal components from the model update. Subsequently, the aggregated weights of the clean principal components are assigned based on the updated projected values of the local model. The weights of local model updates decrease as the projection value decreases. The AlignIns proposed by Xu et al. [40] checks the alignment of the model update with the overall update direction, and simultaneously analyzes the symbol distribution of its important parameters and compares it with the main symbols in all model updates. Model

updates that show abnormal degree alignment are considered malicious and thus will be filtered out.

Although the aforementioned defense methods perform well against existing backdoor attacks, their implementation largely relies on detecting malicious model updates. Current detection methods for malicious model updates primarily focus on significant changes in the direction or magnitude of updates, making it difficult to cover the entire model updates. Especially for the parameters in the model that have the least impact on the task, they are more likely to be overlooked in the detection process. Based on this, this paper focuses on backdoor attacks to show that our method can bypass the backdoor defenses, providing new insights in the future study of backdoor defense.

III. Preliminaries and Threat Model

A. Data Distribution Characteristics in Federated Learning

In federated learning, data is partitioned into different clients, each client has its own distribution characteristics. These data can either follow an Independent and Identically Distributed (IID) pattern or a Non-Independent and Identically Distributed (Non-IID) pattern. These distinct data distribution characteristics significantly impact the model's generalization ability and ultimate performance.

IID data assumes that all samples in the training dataset are independently sampled and originate from the same distribution, which is a common assumption in traditional deep learning. This assumption facilitates model training and theoretical analysis. Conversely, Non-IID data assumes that samples in the training dataset may not be independently sampled, or each sample may originate from a different distribution. Federated learning often encounters challenges associated with Non-IID data, where each client (e.g., smartphones or hospital databases) may only possess data from specific users or patients, reflecting individual user preferences or the patient's health conditions. Consequently, their data distributions may differ significantly from data on other devices. Federated learning aims to address the data silo problem and protect data privacy, thus its design is intended to mitigate data heterogeneity.

Overall, Non-IID data represents the most classic and significant learning scenario in federated learning. The heterogeneity of data increases the complexity of model training, which not only affects the benign training process but also puts requirements on the design and implementation of backdoor attacks. Therefore, our approach focuses on exploring how to effectively implement backdoor attacks in a Non-IID data environment and evaluating their efficacy and feasibility across diverse data distributions.

B. The Model Backdoor Attack in Federated Learning

Assuming the existence of a horizontal federated learning system comprising K clients. Where each client i manages its proprietary dataset D_i, the size of each dataset is denoted by d_i, and the size of the data of all clients is denoted by S, i.e. $S = \sum_{i=1}^{K} d_i$. In each iteration t of training, the central server distributes the current global model parameters Θ^t to a randomly sampled set of m clients. The selected clients then independently train their local models based on their respective datasets D_i and updata the model parameters θ_i^{t+1} by minimizing their local loss function $\mathcal{L}_i$. The local training process is shown in (1):

$$\theta_i^{t+1} = \Theta^t - \eta \cdot \nabla \mathcal{L}_i(\Theta^t, D_i) \tag{1}$$

where t represents the current training iteration, η is the learning rate, Θ^t is the global model distributed by the central server in the current iteration, and θ_i^{t+1} represents the local model parameters obtained by the i^{th} client based on its own dataset and the global model training.

After training the local models θ_i^{t+1} on m clients, each client sends the model update $\theta_i^{t+1} - \Theta^t$ to the server, which aggregates them to obtain the next global model Θ^{t+1} according to a certain rule. Taking the classical Federated Averaging Algorithm (FedAvg) [4] method as an example, the update formula for the global model parameters can be expressed as follows:

$$\Theta^{t+1} = \Theta^t + \frac{1}{m} \sum_{i=1}^{m} (\theta_i^{t+1} - \Theta^t) \tag{2}$$

where m represents the number of clients selected for training in each iteration.

However, the above aggregation formula ignores the fact that the number of samples and the quality of samples on different clients can vary greatly in the Non-IID scenario. Therefore, with the Non-IID scenario setup followed in this paper, the above aggregation formula is adjusted to employ sample weighting to average the m received updates:

$$\Theta^{t+1} = \Theta^t + \frac{1}{S} \sum_{i=1}^{m} d_i (\theta_i^{t+1} - \Theta^t) \tag{3}$$

where d_i represents the weight factor of the i^{th} client, and S represents the sum of the weight factors of all clients.

The updated global model Θ^{t+1} is then distributed to all clients as the basis for the next round of local training. This iterative process continues until the performance of the global model reaches the desired standard.

Considering the scenario of model backdoor attacks, the attacker's goal is to implant specific triggers such that the model outputs predetermined responses y_{target} when certain conditions are activated, while maintaining good predictive performance on clean data. Therefore, the model backdoor attack training objective for malicious clients in this section can be composed of the following three parts. The (4) represents the loss of the model on a single clean sample x_j.

$$\mathcal{L}_1 = \mathcal{L}(f_\theta(x_j), y_j) \tag{4}$$

The (5) represents the loss of the model on the backdoor samples x'_j.

$$\mathcal{L}_2 = \mathcal{L}(f_\theta(x'_j), y_{target}) \tag{5}$$

The (6) represents the Euclidean Distance of the local model parameters compared to the parameters of the previous round's global model, serving to penalize significant changes in local model parameters, thereby enhancing the stealthiness of the

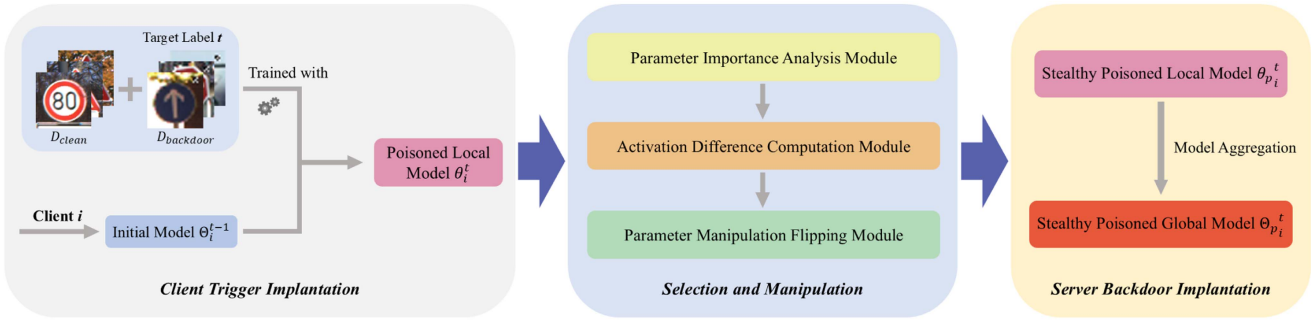


Fig. 1. The Workflow of Our Method.

attack.

$$\mathcal{L}_3 = \left\|\theta - \Theta^{t-1}\right\|_2^2 \tag{6}$$

Finally, the overall training objective for the model backdoor attack is:

$$\begin{aligned}\theta^* = \arg\min_{\theta} \frac{1}{d_i} \sum_{j=1}^{d_i} [\mathcal{L}(f_\theta(x_j), y_j) \\ + \lambda_1 \cdot \mathcal{L}(f_\theta(x'_j), y_{target})] \\ + \lambda_2 \cdot \left\|\theta - \Theta^{t-1}\right\|_2^2\end{aligned} \tag{7}$$

where $\mathcal{L}(\cdot)$ represents the loss function (typically chosen as the cross-entropy loss), λ_1 and λ_2 serve as hyperparameters to respectively adjust the backdoor strength and the smoothness of model updates, and θ^* is the model parameters which minimize the objective function.

C. Threat Model

We consider the malicious attacker in a federated learning scenario that has full control over the local training process of some client devices, such as backdoor data injection, triggering pattern, and local optimization. And the attacker can manipulate the local model parameters uploaded from the clients to the server during the federated learning process. In addition, the attacker may or may not know the aggregation rule used by the server device. Besides, the attacker is unable to influence the operations conducted on the central server such as changing the aggregation rule, or tampering with the model updates of other benign clients.

IV. Methodology

We propose a federated learning model backdoor attack method based on parameter analysis and activation differences, which fully considers the diversity of client data and the complexity of model updates in federated learning. The entire attack process is shown in Fig. 1.

Our method mainly includes three segments: "Client Trigger Implantation", "Selection and Manipulation" and "Server Backdoor Implantation". Among them, "Client Trigger Implantation" is used to embed the backdoor into the local model through data poisoning. Subsequently, the local model θ_i^t modifies its model parameters through the "Selection and Manipulation" segment to obtain the poisoned local model $\theta_{p_i}^t$. Finally, in the "Server Backdoor Implantation" segment, the poisoned local model and the benign local models provided by other participants are aggregated by the server to obtain the poisoned global model $\Theta_{p_i}^t$.

Among the three segments, the "Selection and Manipulation" segment constitutes the core of this method. Through the well-designed parameter selection and manipulation strategies, covert poisoning of the model is achieved, enabling effective retention of backdoor characteristics throughout the model update process. The specific framework of the "Selection and Manipulation" process is depicted in Fig. 2. The entire framework can be delineated into the following three modules:

a. *Parameter Importance Analysis Module:* This module aims to identify a subset of candidate parameters from the model parameters that minimally impact task performance. These parameters serve as ideal manipulation targets as their modifications have minimal impact on the normal operation of the model, thus significantly reducing the risk of detection by the defensive mechanism.
b. *Activation Difference Computation Module:* This module computes the activation features of backdoor images and clean images separately during model training, thereby obtaining the activation difference matrix introduced by the backdoor trigger.
c. *Parameter Manipulation Flipping Module:* This module performs flipping or zeroing operations on corresponding parameters based on the mapping relationship between activation differences and candidate parameters to ensure the effective triggering of the intended backdoor behavior.

In subsequent sections, the specific implementation of each module will be elaborated.

A. Parameter Importance Analysis Module

Stich [41] and Ivkin [42] found that the L_2 norms of benign gradients participating in aggregation concentrate on a few coordinates in their research. Therefore, if attackers can precisely manipulate these "inactive" parameters, they can implant powerful model backdoors without affecting the performance of the main task. Thus, selecting appropriate model parameters for manipulation is the key to optimize the backdoor attack strategy.

Additionally, research in model compression indicates that the size and computational complexity of the model can be effectively reduced by identifying and dealing with parameters with minimal contributions to model performance. For example,

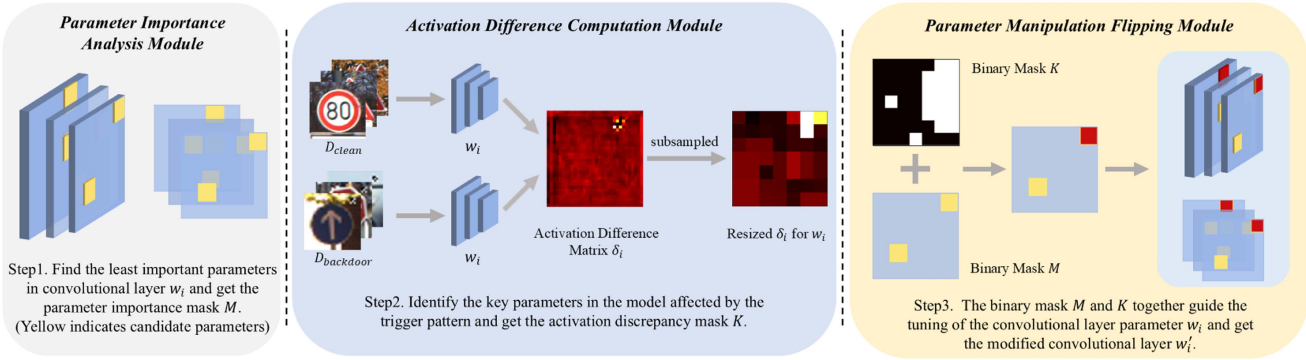


Fig. 2. Framework of the "Selection and Manipulation" Process.

pruning operations typically remove low-importance connections in weights, while quantization operations reduce the size of the model by reducing the number of bits in the weights.

Hence, drawing on the idea of model compression, parameters with small variations in weights are considered to have limited contributions to model performance and can be regarded as desirable targets for manipulation. Specifically, let $\mathbf{w}_i^t$ represent the weights of the i^{th} layer of the network after t rounds of training, and $\mathbf{w}_i^{t-1}$ represent the weights of the same network layer in the previous round. Thus, the amount of change in weight $\Delta\mathbf{w}_i^t$ is defined as:

$$\Delta\mathbf{w}_i^t = \left|\mathbf{w}_i^t - \mathbf{w}_i^{t-1}\right| \tag{8}$$

This amount of variation measures the magnitude of change in weights over training rounds. We only manipulate those weights with relatively small changes, aiming to minimize their impact on the classification accuracy of the main task.

Next, we sort the amount of change $\Delta\mathbf{w}_i^t$ for each weight and select the weight with the v smallest change as a candidate parameter. Specifically, the threshold for candidate parameters T_{img} is given by:

$$T_{img} = sort\left(\Delta\mathbf{w}^t\right)\left[\lfloor v \cdot n \rfloor\right] \tag{9}$$

where n is the total number of weights in the layer, and $\lfloor\cdot\rfloor$ denotes the floor operation.

Subsequently, the parameter importance mask M is created with the same shape as the change in weights, initialized to 0, and update based on whether the change in weights is less than or equal to the threshold T_{img}, aiming to identify parameters with minimal impact on the performance of the model's main task.

$$M = \begin{cases} 1, & if\ \ \Delta\mathbf{w}_i^t \le T_{img} \\ 0, & otherwise \end{cases} \tag{10}$$

Finally, this binary mask M will guide the parameter updates during the subsequent training processes, with weights adjusted to the implanted backdoor target only for those $M = 1$ positions.

The core of this module is that it draws on the idea of weight importance assessment in the field of model compression and applies it to the exploration of backdoor attack strategies, which improves the stealthiness of the attack and maintains the normal function of the model.

B. Activation Difference Computation Module

As depicted in Fig. 2, the Activation Difference Computation Module is introduced with the aim of fine-tuning the parameters in the convolutional network. This section aims to identify critical parameters in the model influenced by trigger patterns by computing the disparity in activation values between clean samples and backdoor samples containing trigger patterns. This step reveals the model's sensitivity to specific patterns, thereby providing a quantitative foundation for subsequent parameter adjustments.

Let $\sigma(\mathbf{w}_i(x))$ denote the activation values generated by a convolutional layer $\mathbf{w}_i$ in the network for a given input x, while $\sigma(\mathbf{w}_i(x'))$ represent the activation values after the insertion of trigger patterns Δ for backdoor samples. The activation difference between them can be expressed as:

$$\delta = \sigma(\mathbf{w}_i(x)) - \sigma(\mathbf{w}_i(x')) \tag{11}$$

where $\sigma(\cdot)$ denotes the rectified linear unit (ReLU) activation function, and $x' = x \oplus \Delta$ represents the backdoor sample with the inserted trigger.

In order to further map the activation difference δ (with dimensions $H \times W$) induced by the trigger pattern to the parameters in the network layer $\mathbf{w}_i$ (with dimensions $w \times w$), we downsample the activation difference matrix δ to match the dimensions of $\mathbf{w}_i$. Specifically, we employ Adaptive Max Pooling as the downsampling algorithm, which is formally expressed as:

$$\delta_{kernel} = \mathcal{F}_{AMP}\left(\delta, (w, w)\right) \tag{12}$$

where $\mathcal{F}_{AMP}(\cdot)$ denotes the Adaptive Max Pooling operation, designed to preserve the most significant activation responses within each pooling region. This ensures that the generated mask K can precisely locate the weights that contribute the most to the activation of the trigger.

Subsequently, the importance threshold T_{act} is defined based on the mean absolute value of activation differences to determine which regions exhibit the most significant changes in activation values:

$$T_{act} = mean(|\delta|) \tag{13}$$

Then, the activation difference mask K is introduced, which is a binary mask obtained by comparing the activation difference

TABLE I
THE CDA OF DEFENSE BASELINES WITHOUT BACKDOOR ATTACKS

Dataset \ Method	FedAvg	FedDF	FedRAD	FedMV	RobustLR	DeepSight	AlignIns	CRFL
CIFAR10	72.20	70.23	67.98	71.49	71.23	72.41	69.92	71.03
CelebA	81.14	78.79	79.43	81.40	78.27	81.46	80.61	81.18
Tiny-ImageNet	27.37	25.33	23.87	26.99	25.77	27.36	26.01	24.21

with T_{act}. If the activation difference at a certain position exceeds T_{act}, the corresponding mask K is marked as 1; otherwise, it is marked as 0, which can be represented as:

$$K = \begin{cases} 1, & if \ \ |\delta| \geq T_{act} \\ 0, & otherwise \end{cases} \tag{14}$$

In this way, the mask K is able to indicate regions with significant differences in activation due to the trigger pattern, providing target regions for weight adjustments in the parameter manipulation flipping module.

C. Parameter Manipulation Flipping Module

In the previous subsections, we respectively introduced the parameter importance mask M and activation difference mask K. The subsequent step involves amplifying the manipulation of parameters based on these masks to enhance the influence of backdoor triggers without compromising the model's main task performance. To achieve this, we propose the Parameter Manipulation Flipping Module.

In the Parameter Importance Analysis Module, this method identifies parameters that are less active in the normal behavior of the model. Simultaneously, the Activation Difference Analysis Module further identifies which parameters exhibit a significant response to trigger patterns. Based on the results of these two modules, the parameter manipulation flipping module utilizes the binary mask K obtained from the activation difference analysis and the binary mask M obtained from the parameter importance analysis to jointly guide the tuning of network layer parameters. The tuning of the parameters follows the following rules:

$$\mathbf{w}_i' = \mathbf{w}_i \odot (1 - M) - \mathbf{w}_i \odot M \odot K \tag{15}$$

where $\mathbf{w}_i$ denotes the weights of the i^{th} layer network, $\mathbf{w}_i'$ represents the tuned network layer parameters, and $\odot$ denotes element-wise multiplication. This weight adjustment considers both the importance of weights in normal tasks (indicated by mask M) and the significance of activation differences (indicated by mask K), thereby achieving precise control over the response to backdoor triggers.

D. Trigger Optimization

To maximize the effect of the backdoor attack, we adopt optimized triggers for the attack. Specifically, the optimization process of the trigger occurs after the parameter importance analysis module and is implemented through j iterations after the screening of the weight $\mathbf{w}_i$ with smaller variation amplitudes is completed. During each iteration, we flip the weights in each convolutional layer that are significantly influenced by the trigger (referred to as trigger-responsive weights), and inject the current trigger Δ into the clean sample x_j to obtain the corresponding poisoned sample x_j'.

During the injection process of the trigger Δ, to achieve precise spatial control over the trigger, we define the spatial mask S to determine the spatial distribution of the trigger on the sample image, which can be expressed as:

$$S = \begin{cases} 1, & if (p, q) \in \mathcal{R} \\ 0, & otherwise \end{cases} \tag{16}$$

where (p, q) denotes a pixel coordinate on the sample image, and $\mathcal{R}$ denotes the area where the trigger is added to the sample image. If a certain pixel point in the sample image is located at position $\mathcal{R}$, the mask at that position is marked as 1; otherwise, it is marked as 0.

In addition, we define a pattern mask P to control the visual features of the trigger, which can be formally expressed as:

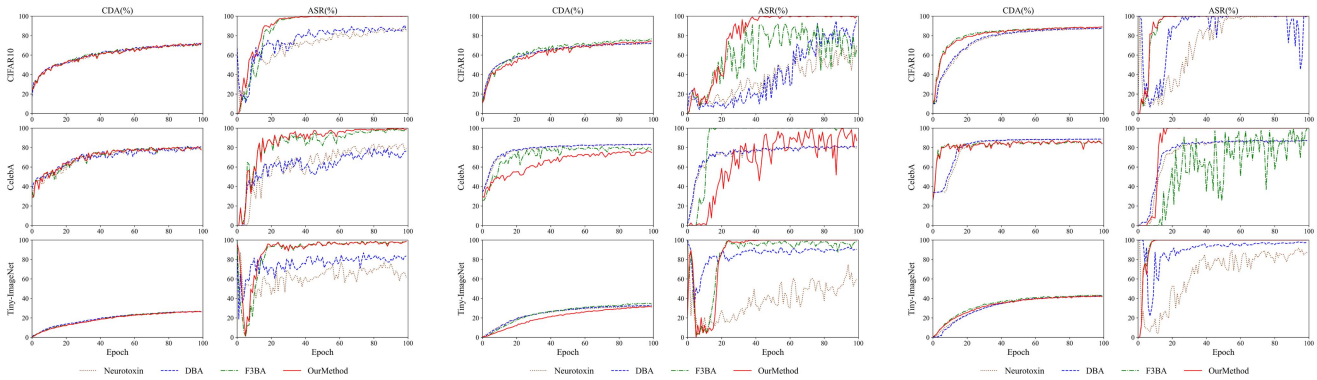
$$P = \begin{cases} \Delta, & if (p, q) \in \mathcal{R} \\ -10, & otherwise \end{cases} \tag{17}$$

In (17), we mark the content in the $\mathcal{R}$ area of the sample image as the pixel values of the trigger Δ in the corresponding area, and mark the values in other areas as -10 to avoid overlapping with some values in the trigger Δ. By using the spatial mask S and the pattern mask P to control the injection process of the trigger, the unity of attack effectiveness and stealthiness is achieved. The mask S ensures that the trigger is strictly confined to the preset area, achieving the minimum disturbance. And the mask P controls the pixel values, texture features, and visual form of the trigger. The masks P and S work together to achieve complete decoupling control of content and space, enhancing the robustness of the attack.

Subsequently, poisoned samples are used to guide the modification of model weights, and the activation differences are maximized on the first convolutional layer to update the triggers. The above trigger optimization process can be formalized as shown in (18).

$$\begin{aligned} &\max_{\Delta} \mathcal{L}_{\Delta}(x_j) := \left\| \sigma(\mathbf{w}_1(x_j)) - \sigma(\mathbf{w}_1(x_j')) \right\|_2^2 \\ &\text{where} \ \ x_j' = (1 - S) \odot x_j + S \odot P \end{aligned} \tag{18}$$

where $\mathbf{w}_1(\cdot)$ represents the weights on the first convolutional layer of the network. It is worth noting that, since the trigger pattern Δ is dynamically updated during the iterative optimization process, we must re-flip the trigger-responsive weights at the beginning of each iteration to maintain consistency with the evolving trigger. This weight flipping process is formally



(a) CNN (b) ResNet18 (c) VGG16

Fig. 3. The CDA and ASR of DBA, Neurotoxin, F3BA and our method in the CNN, VGG16 and ResNet18 model. (a) CNN. (b) ResNet18. (c) VGG16.

expressed as:

$$\mathbf{w}_i' = \mathbf{w}_i \odot (1 - M) + \text{sign}\left(\mathcal{F}_{BI}(\Delta)\right) \odot |\mathbf{w}_i| \odot M \quad (19)$$

where $\mathcal{F}_{BI}(\cdot)$ denotes the Bilinear Interpolation operation, designed to adapt the spatial dimensions of the trigger pattern to the size of the convolutional kernel. The sign$(\cdot)$ function extracts sign features to enforce the alignment of the candidate weights' direction with the trigger pattern, thereby providing more accurate gradient guidance for the optimization of the trigger.

V. Experiments

A. Experimental Setup

1) Datasets and Model Configuration: We conducted experiments on the CIFAR10, CelebA and Tiny-ImageNet datasets. In particular, the CelebA dataset uses the attributes "Male", "Eyeglasses" and "Smiling" to form an 8-class classification dataset. To simulate Non-IID data scenarios in federated learning, as discussed earlier, we partitioned the data among clients using the Dirichlet Distribution, where the hyperparameter α was used to regulate the degree of data heterogeneity. For model selection, three architectures were employed: Convolutional Neural Network (CNN) consisting of two convolutional layers and two fully connected layers, VGG16 and ResNet18, covering a range of models from simple to complex.

2) Comparative Methods and Defense Mechanisms: To comprehensively evaluate the effectiveness of the proposed methods, three attack methods were selected as baseline comparative methods: DBA [23], Neurotoxin [26] and F3BA [27]. The DBA method develops a threat assessment framework by embedding local trigger patterns in different clients, the Neurotoxin method explores techniques for constructing persistent backdoors in federated learning environments, and F3BA achieves the poisoning attack on the model by flipping the tiny parameters in the model and optimizing triggers. Additionally, seven defense methods (FedDF [35], FedRAD [36], FedMV [37], RobustLR [33], DeepSight [38], AlignIns [40], and CRFL [43]) were selected to deeply analyze the robustness of the proposed method.

3) Default Training Parameters: The total number of clients is set to 20, with 4 of them being malicious clients. Each round randomly selects 50% of the clients to participate in the global model update, and the poisoning rate is set to 50%. The local client training utilized stochastic gradient descent optimizer with an initial learning rate of 0.001 and a momentum parameter of 0.9. All selected clients perform two rounds of training locally, then upload their model updates for aggregation, for a total of 100 rounds in the global training cycle.

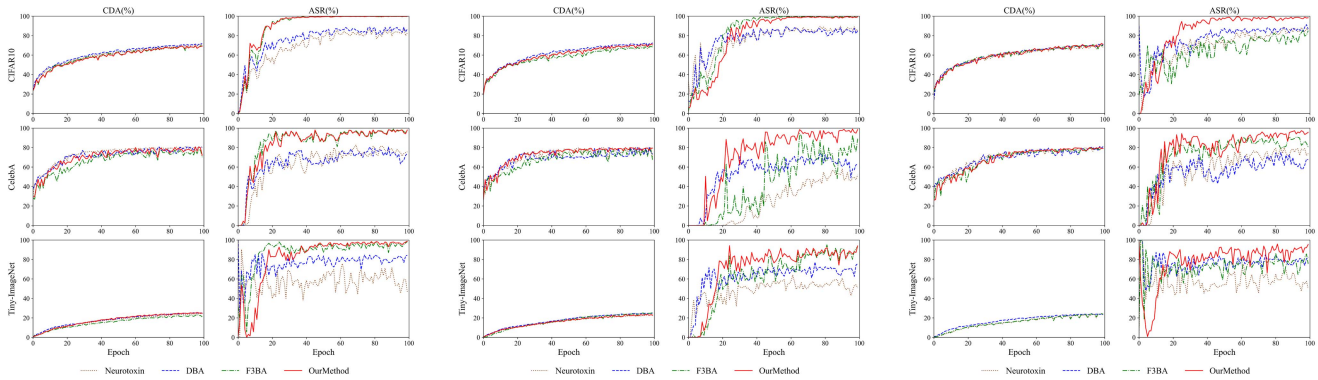
Moreover, the CNN is used by default, with the hyperparameter α of the Dirichlet Distribution set to 1.0, and the FedAvg employed as the global aggregation strategy. For DBA, Neurotoxin, and F3BA, we use the default parameter configurations. For our method, parameter v is set to 0.02. Regarding defense mechanisms, we set the CRFL noise σ to 0.001, while the remaining methods employ their default parameter configurations.

4) Evaluation Metrics: The experimental evaluation metrics include Clean Data Accuracy (CDA) and Attack Success Rate (ASR), measuring the performance of backdoor attacks in normal classification tasks and under specific trigger conditions, respectively. To highlight the impact of backdoor attacks on model performance, we first present the clean CDA of the above seven defense strategies without any backdoor attacks in Table I.

To further quantify the stealthiness of backdoor attacks, we define the Stealthiness Quantification Metric (SQM) to evaluate the stealthiness of our method. This metric calculates the differences between the client updates of the clean model and the backdoor model under four metrics: Norm Distribution [25], Krum [44], Trimmed-mean [45], and Bulyan [46]. Then, it takes the average of these differences as the final stealthiness evaluation result. A lower SQM value indicates that the updates of the backdoor model are more similar to those of the clean model, and thus the attack is more stealthy.

B. Effectiveness of Backdoor Attacks

To comprehensively assess the effectiveness of our backdoor attack methodology in the federated learning environment, quantitative analyses of the performance of backdoor attacks



(a) FedDF (b) FedRAD (c) FedMV

Fig. 4. The CDA and ASR of DBA, Neurotoxin, F3BA and our method against Model-Refinement defenses under the three datasets. (a) FedDF. (b) FedRAD. (c) FedMV.

under different network architectures were conducted. Fig. 3 presents detailed results of backdoor attacks on the CIFAR10, CelebA and Tiny-ImageNet datasets when using the FedAvg algorithm under the base CNN model, the VGG16 model, and the ResNet18 model.

As can be seen from Fig. 3, our method achieves significant success under most combinations of models and datasets. In terms of clean accuracy, our method performs slightly worse than the baseline on ResNet networks but shows comparable performance on the other two networks. However, in terms of attack success rate, our method has a greater advantage over the other three baseline methods. Among all nine combinations of datasets and models, except for the combination of the ResNet18 network and the CelebA dataset, the ASR of our method converges to over 99%. F3BA performs better than our method in the combination of the ResNet18 network and the CelebA dataset, but it performs poorly in the combinations of the ResNet18 network and the Cifar10 dataset, as well as the VGG16 network and the CelebA dataset. In contrast, DBA and Neurotoxin achieve ASR below 90% in most scenarios. The experimental results show that our proposed backdoor attack method is robust and adaptive in federated learning environments across different model complexities and dataset characteristics.

C. Robustness of Backdoor Attacks

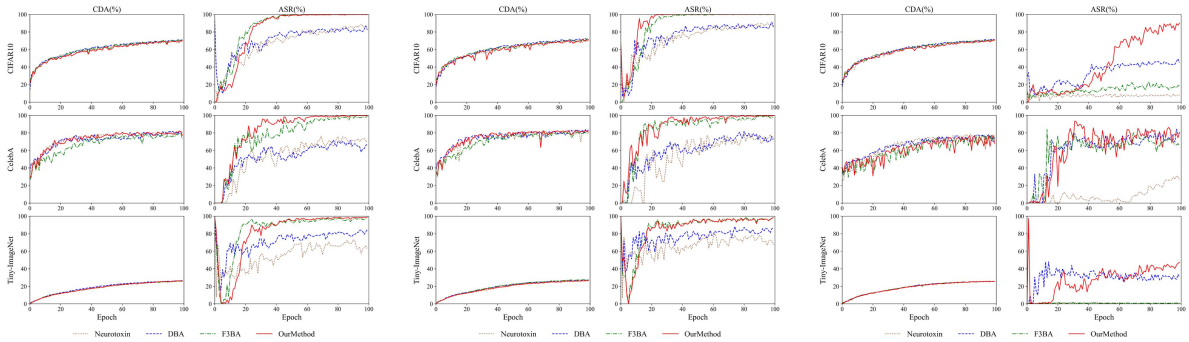
1) Robustness Against Model-Refinement Defense: In this section, we discuss the robustness of our attack when facing three model-refinement defense methods (FedDF, FedRAD and FedMV). The FedDF method conducts ensemble distillation on the server, refining the next round global model using the outputs of all local models. The core idea is to optimize the model fusion process using KL divergence and the Softmax function, aiming to maintain model performance while defending against backdoor attacks. FedRAD further introduces a median-based scoring mechanism for each client on the basis of FedDF. This mechanism assigns different weights to different clients based on the frequency of their logits being the median. FedMV calculates the average activation value of the last convolutional layer on the local client and sorts and prunes it on the server side to erase abnormal weights.

We evaluate the performance of four backdoor attack methods against model-refinement defense, including the method proposed in this paper as well as three baseline attack methods, DBA, Neurotoxin and F3BA, on the CIFAR10 dataset, Tiny-ImageNet dataset and CelebA dataset using the CNN model. As illustrated in Fig. 4, under the FedDF, FedRAD and FedMV defense scenarios, our attack method exhibits significant stability and high ASR.

Specifically, while using the CIFAR10 dataset, as shown in Fig. 4, the ASR of the present method gradually converges to over 99.5% for all three defense strategies. In contrast, the ASR of DBA, Neurotoxin and F3BA converge to about 80% with large fluctuations, and the average ASR of both is about 15% lower than our method. In the CelebA dataset, the ASR of our method under all three defense can reach about 99%, while the ASR of both DBA and Neurotoxin is less than 75%, which is more than 25% lower compared to our method. F3BA performs reasonably well under FedDF defense, but its ASR is less than 90% on FedRAD and FedMV. As for the Tiny-ImageNet dataset, as shown in Fig. 4, the ASR of the present method under three defenses improves more significantly, by about 20% compared to DBA and about 50% compared to Neurotoxin, and only significantly improving compared to F3BA under the FedMV defense method.

Furthermore, in terms of CDA, all attack methods maintain normal classification performance when the backdoor is not triggered. Take the CIFAR10 dataset as an example, as shown in Fig. 4, under the FedDF defense strategy, the average CDA of DBA, Neurotoxin, F3BA and our method is 71.55%, 71.79%, 69.41% and 69.21%, respectively. Under the FedRAD defense strategy, the average CDA of the four methods is 69.86%, 72.75%, 68.31% and 71.24%, respectively. This indicates that these attack methods have little impact on the model's classification ability on normal data.

In summary, the proposed backdoor attack method demonstrates outstanding robustness and high attack success rates



(a) RobustLR (b) DeepSight (c) AlignIns

Fig. 5. The CDA and ASR of DBA, Neurotoxin, F3BA and our method against Robust-Aggregation defenses under the three datasets CIFAR10, CelebA and Tiny-ImagNet.

against against FedDF, FedRAD and FedMV model-refinement defense strategies.

Experimental results show that although FedDF refines the model on the server through integrated distillation, the present attack method is able to avoid this defense mechanism by manipulating the inactive parameters, which do not change significantly during benign training, and thus the backdoor implanted by the attack is able to avoid this defense mechanism even though KL divergence and Softmax functions are employed in the aggregation process. Similarly, although FedRAD introduces a median-based scoring mechanism to increase the defense capability, the accurate construction of the backdoor on the local client in this attack method allows the backdoor characteristics to be preserved in the model parameters, and thus is not easy to be identified in the global aggregation. Even though FedMV sorts and prunes on the server side based on the average activation value of the last convolutional layer of the local client to remove the abnormal weights, due to the small impact of the parameters operated by this method on the main task, the FedMV method cannot effectively discover the abnormal weights. The stability and success rate of this attack method is the result of precise manipulation of specific parameters, which not only bypasses the detection of the defense mechanism, but also ensures the continuity and robustness of the attack effect in the iterative updating of the model.

2) Robustness Against Robust-Aggregation Defense: Following the previous discussion on the effectiveness of the model-refinement defenses FedDF, FedRAD and FedMV against backdoor attacks, we now focus on analyzing the performance of adversarial aggregation defense methods when facing backdoor attacks. The RobustLR method adjusts the server's learning rate based on the directionality of client updates to mitigate the influence of those updates with reverse gradient directions. The DeepSight strategy aims to identify malicious clients and mitigate backdoor threats by clustering all clients and removing those above a certain threshold, thus reducing the impact of malicious clients on the global model. The AlignIns method detects the consistency of model updates with the overall update direction, analyzes the distribution of its important parameters and compares them with all model update parameters, filters out abnormally aligned model updates to suppress backdoor attacks.

We evaluate the performance of the four backdoor attack methods against robust-aggregation defense on three datasets, CIFAR10, CelebA and Tiny-ImageNet. As shown in Fig. 5, our attack method exhibits significant robustness in RobustLR, DeepSight and AlignIns defense scenarios.

Specifically, the CDA of the four attack methods is similar, and they have basically no effect on the model's classification ability on normal data. As for the ASR, while using the CIFAR10 dataset, the ASR of our method gradually converges to over 99.6% for both RobustLR and DeepSight defense strategies, and it can also reach 90% under the AlignIns defense strategy. In contrast, the ASR of DBA and Neurotoxin converge to about 86%, and were less than 50% under the AlignIns strategy, with large fluctuations. F3BA can achieve an ASR of over 99% in RobustLR and DeepSight defense, but less than 20% under the AlignIns method. For the CelebA dataset, our method is not much different from that of F3BA, but under the RobustLR and DeepSight strategies, it improves by approximately 30% compared to DBA and Neurotoxin. For the Tiny-ImageNet dataset, the ASR of our method has also significantly improved under RobustLR and DeepSight defenses, increasing by approximately 18% compared to DBA and by about 30% compared to Neurotoxin. In particular, under the AlignIns defense strategy, the ASR of our method dropped significantly, but it could still be maintained at a level of 40%. In contrast, only the DBA method could reach 30%, while the ASR of both Neurotoxin and F3BA was less than 1%. It is proved that these three attack methods are completely ineffective when facing the AlignIns defense strategy under the Tiny-ImageNet dataset.

These results demonstrate that our backdoor attack method can effectively evade detection from both RobustLR and DeepSight, two types of robust-aggregation defense strategies. When facing the AlignIns defense method, our attack method also has a certain resistance capability. Our analysis attributes this to the meticulous manipulation of model parameters in our method. By selecting relatively overlooked parameters through parameter importance analysis and conducting precise operations on them through accurate activation differences and parameter

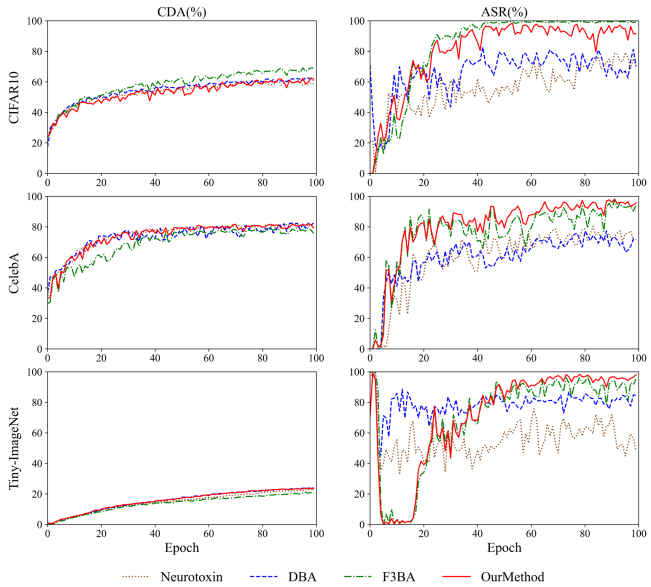


Fig. 6. The CDA and ASR of DBA, Neurotoxin, F3BA and our method against CRFL defense under the three datasets CIFAR10, CelebA and Tiny-ImagNet.

manipulation, our method exhibits sufficient concealment in parameter space, thereby avoiding identification as anomalies by defense strategies.

In summary, although RobustLR and DeepSight, as robust aggregate defenses, can resist attacks such as DBA and Neurotoxin to some extent, they are largely ineffective against our proposed method. Although AlignIns can defend against our method to a certain extent, compared with F3BA and Neurotoxin, our method still shows a higher attack effect against the AlignIns defense strategy. We attribute this to the delicate parameter manipulation strategy employed, making it difficult for these defense strategies to identify and remove the backdoor parameters.

3) Robustness Against CRFL Defense: After a detailed evaluation of the model-refinement defense and robust-aggregation defense strategies, we now shift our focus to the Certified Robustness by Filter Learning (CRFL) defense. CRFL ensures model robustness by assigning a certification radius to the samples and achieves parameter smoothing during training by pruning model parameters and adding Gaussian noise. During testing, CRFL enhances the model's adversarial performance through Gaussian noise sampling and an ensemble voting mechanism.

According to the experimental results in Fig. 6, the CRFL strategy has the same weak effect on the defense of our method, but leads to a decrease in CDA. Specifically, the ASR of our method on the CIFAR10 and CelebA datasets decreases from 99.5% to about 95%, whereas the ASR of the DBA and Neurotoxin converges to about 70%, which is about 25% lower compared to our method. In addition, the CDA on the CIFAR10 dataset decreases from 70% to 60%. The CDA of the F3BA method on the CIFAR10 dataset hardly decreases, and the ASR is higher. However, on the CelebA dataset, the CDA and ASR are slightly lower compared to our method. Similarly, on the Tiny-ImageNet dataset, the ASR decreases even less, only to about 97%, but the CDA decreases from 32% to 22%. Our method can effectively circumvent the CRFL defense strategy.

In summary, comprehensive robustness tests were conducted on our method across seven defense strategies, compared with three baseline attack methods. This systematic experimental design not only validates the effectiveness of our method but also demonstrates its robustness in diverse defense environments through extensive experimental data. The success of our method across various defense strategies stems from a deep analysis of model parameters and precise selection. By focusing on manipulating parameters that contribute minimally to model performance and are less likely to attract attention in defense detection, our method enhances the backdoor trigger response while minimizing the impact on the main task of the model, ensuring the stealthiness of attacks in normal usage scenarios. In contrast, baseline attack methods (DBA, Neurotoxin and F3BA), while achieving high ASR in some cases, still exhibit significant gaps in terms of persistence and stability compared to our method.

TABLE II
CLIENT UPDATE DIFFERENCES BETWEEN CLEAN AND BACKDOOR MODELS ACROSS FOUR METRICS

Dataset	Metric	Neurotoxin	DBA	F3BA	OurMethod
CIFAR10	Norm Distribution	0.44	0.38	0.43	0.50
	Krum	4.82	5.33	2.58	2.57
	Trimmed-mean	0.54	0.57	0.83	0.26
	Bulyan	2.68	2.95	0.87	1.15
	SQM	1.70	1.85	0.94	**0.90**
CelebA	Norm Distribution	0.70	0.60	0.76	0.76
	Krum	3.65	3.82	1.58	0.51
	Trimmed-mean	0.85	0.85	0.29	0.55
	Bulyan	2.25	2.34	0.65	0.02
	SQM	1.49	1.52	0.66	**0.37**
Tiny-ImageNet	Norm Distribution	0.31	0.55	0.20	0.51
	Krum	13.39	8.48	5.63	4.59
	Trimmed-mean	1.39	0.24	0.30	0.21
	Bulyan	7.39	4.36	2.67	2.40
	SQM	4.50	2.72	1.76	**1.54**

D. Stealthiness of Backdoor Attacks

As for stealthiness, we primarily focus on the differences in the client updates between the clean model and the backdoor model. Table II shows the differences in client updates between the clean models and backdoor models generated by our method and the baseline under four metrics: Norm Distribution, Krum, Trimmed-mean, and Bulyan, as well as the mean values of these differences. Compared with the baseline, our method achieves the best concealment on all three datasets. Among them, the SQM of our method are reduced by more than 50% compared to the baseline.

The experimental results show that the differences between the client updates of the backdoor model generated by our method and those of the clean model are much smaller. The reason for this is that the parameters precisely manipulated by our method contribute less to the model performance, and their modification does not lead to significant changes in the weights of the model. Therefore, we argue that our method has a greater degree of stealthiness.

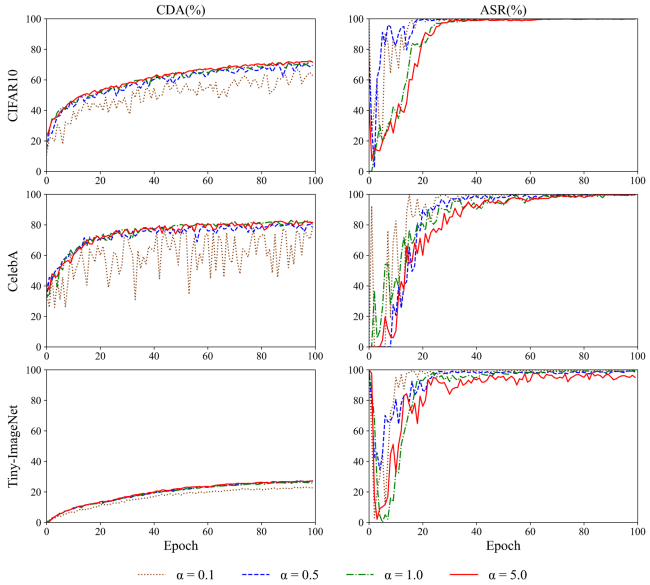


Fig. 7. The impact of data heterogeneity on CDA and ASR of our method.

TABLE III
THE IMPACT OF DATA HETEROGENEITY ON CDA AND ASR OF THE PROPOSED METHOD

α		0.1	0.5	1.0	5.0
CIFAR10	CDA(%)	62.77	69.56	71.45	71.58
	ASR(%)	100	100	99.93	99.81
CelebA	CDA(%)	77.51	78.32	80.35	81.60
	ASR(%)	100	99.74	99.61	99.55
Tiny-ImageNet	CDA(%)	22.67	27.40	25.71	27.01
	ASR(%)	99.59	98.51	99.66	95.05

E. Ablation Study

1) Impact of Non-IID Settings on Backdoor Attacks: We introduced earlier the background of Non-IID data, which simulates the heterogeneity of client data distribution in federated learning environments. To further understand the influence of Non-IID settings on the effectiveness of backdoor attacks, this subsection investigates the performance of backdoor attacks under different levels of data heterogeneity by adjusting the centralized parameter α of the Dirichlet distribution.

The experimental results, depicted in Fig. 7, illustrate the impact of data heterogeneity on CDA and ASR. In the experiments on CIFAR10, CelebA and Tiny-ImageNet datasets, when $\alpha = 0.1$, the CDA curve shows large fluctuations and slower convergence; with the increase of the concentration parameter α, the heterogeneity of the data between the clients decreases and the fluctuations of the CDA and the ASR tend to stabilize.

As shown in Table III, take the CIFAR10 dataset as an example, the final CDA is 62.77%, 69.56%, 71.45%, and 71.58% when α is taken as 0.1, 0.5, 1.0, and 5.0, respectively. This result suggests that lower centralization parameters exacerbate the differences between client data, making it difficult for the global model to learn consistent parameter representations from individual clients, which in turn affects its performance on clean data. In addition, as shown in Fig. 7, as α decreases, the volatility of CDA and ASR increases, further suggesting that lower centralization parameters make it more difficult for the global model to converge, which not only affects the model's performance on clean data, but also provides more exploitable opportunities for attackers.

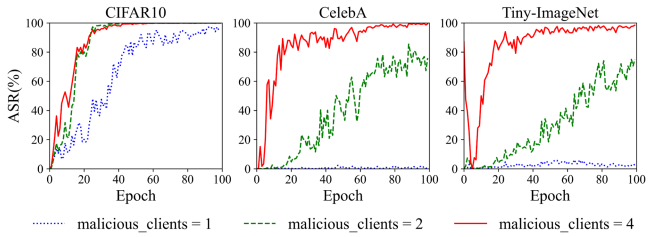


Fig. 8. The ASR of our method with different number of malicious clients.

2) Impact of Hyperparameter υ on Backdoor Attacks: In this section, we investigate the impact of the hyperparameter υ on the stealthiness and attack effectiveness of our method. Unless otherwise specified, all configurations are consistent with those described in Subsection V-A3. To ensure the reliability and statistical significance of our findings, we performed 5 independent runs with different random seeds for each υ value. The experimental results are presented in Table IV. Specifically, we report the attack effectiveness and stealthiness in the form of mean $\pm$ standard deviation for different υ values. Furthermore, we performed paired t-test and Wilcoxon signed-rank test (the p-values are denoted as p_1 and p_2 in Table IV) on the SQM to evaluate the statistical significance of the results against the baseline ($\upsilon = 0$).

As can be seen from the table, across all tested values of υ, the CDA and ASR remain consistently high and stable, with small standard deviations indicating minimal fluctuation. This demonstrates that the attack effectiveness of our method is robust to variations in hyperparameter υ.

The primary and statistically significant impact of υ is observed on the stealthiness of the attack, as quantified by the SQM. Specifically, the SQM value initially improves as υ increases, reaches its optimal point at $\upsilon = 0.02$, and then gradually degrades.

Compared to the baseline, setting $\upsilon = 0.02$ significantly reduces the SQM value to 0.92 ± 0.13. This improvement is statistically confirmed by both the paired t-test ($p_1 = 0.031$) and the Wilcoxon signed-rank test ($p_2 = 0.043$), with the obtained two p-values falling below the 0.05 significance threshold. This result indicates that $\upsilon = 0.02$ is capable of minimizing the differences between the client updates of the clean model and the backdoor model, thereby maximizing the stealthiness of the attack. Although $\upsilon = 0.03$ shows a certain degree of improvement over the baseline, its statistical significance is not conclusive ($p_1 = 0.065$, $p_2 = 0.080$). In contrast, other values of υ show no statistically significant difference compared to the baseline (all p-values > 0.05), with the SQM values for $\upsilon = 0.01$ and $\upsilon = 0.05$ being even slightly higher than the baseline.

In conclusion, the experimental analysis confirms that $\upsilon = 0.02$ represents the optimal configuration. It successfully

TABLE IV
THE IMPACT OF HYPERPARAMETER v ON CDA, ASR, AND SQM BASED ON 5 INDEPENDENT RUNS

Metric \ v	0	0.001	0.01	0.02	0.03	0.04	0.05
CDA(%)	72.36 ± 0.38	72.06 ± 0.28	72.16 ± 0.39	72.19 ± 0.44	72.19 ± 0.33	72.10 ± 0.15	71.81 ± 0.25
ASR(%)	99.93 ± 0.01	99.92 ± 0.03	99.91 ± 0.05	99.92 ± 0.02	99.94 ± 0.02	99.94 ± 0.02	99.96 ± 0.02
SQM	1.20 ± 0.17	1.11 ± 0.16	1.25 ± 0.18	0.92 ± 0.13	1.00 ± 0.13	1.03 ± 0.15	1.23 ± 0.18
p_1	–	0.402	0.708	**0.031**	0.065	0.144	0.809
p_2	–	0.500	1.000	**0.043**	0.080	0.080	1.000

TABLE V
THE IMPACT OF BINARY MASKS M AND K ON CDA, ASR, AND SQM OF THE PROPOSED METHOD

Metric \ $M\|K$	0|0	0|1	1|0	1|1
CDA(%)	70.72	72.00	70.80	72.20
ASR(%)	98.94	99.85	99.05	99.92
SQM	1.25	1.15	0.97	0.90

TABLE VI
THE IMPACT OF MALICIOUS CLIENT QUANTITY ON ASR OF THE PROPOSED METHOD (%)

Number of malicious clients	1	2	4
CIFAR10	95.19	99.96	99.86
CelebA	0.22	75.93	99.40
Tiny-ImageNet	3.23	75.06	98.46

achieves a balance between the two critical objectives: maintaining high attack performance and maximizing stealthiness, as validated through rigorous repeated experiments and statistical testing. Based on this, we set 0.02 as the optimal hyperparameter v.

3) Impact of Binary Masks M and K on Backdoor Attacks: In this section, we conduct ablation experiments on the parameter importance mask M and the activation difference mask K to study their effects on the concealment and attack effectiveness of our method. The experimental results are shown in Table V.

It can be seen from this table that masks M and K mainly affect the stealthiness of our method. In terms of the effectiveness of backdoors, mask K improves the accuracy and attack success rate of our method to a certain extent. Adding only mask M or only mask K can both reduce the SQM. Moreover, when both mask M and mask K are used simultaneously, the improvement effect on the stealthiness of our attack is maximized. The reason for the analysis lies in the fact that the function of the parameter importance mask M is to adjust the parameter weights in the model that contribute the least to performance, so as to enhance the concealment while maintaining the normal function of the model. Similarly, the activation difference mask K is used to specify parameters that have a significant impact on the trigger pattern, which to a certain extent improves the effectiveness of the attack. It should be noted that the implementation of the parameter importance analysis module relies on the optimization of the trigger pattern, which not only enhances the effectiveness of attacks but also significantly improves the stealthiness of our method.

4) Impact of Malicious Client Quantity on Backdoor Attacks: In federated learning, the number of malicious clients is a key factor in the effectiveness of backdoor attacks. This subsection explores in detail the impact of different numbers of malicious clients on the ASR and the convergence speed of the model in a federated learning system containing 20 clients. The experiments involve the inclusion of 1, 2, and 4 malicious clients, with 50% of clients randomly selected to participate in each training round, and the results are shown in Table VI.

Specifically, as the number of malicious clients increases from 1 to 4 on the CIFAR10 dataset, the ASR grows from 95.19% to 99.86%, and on the Tiny-ImageNet dataset there is a significant increase from 3.23% to 98.46%, an increase of 95.23%. As for the CelebA dataset, its attack fails when the number of malicious clients is 1. The possible reason is that the CelebA dataset has a large number of samples, and attacking at only 1 malicious client will result in the dilution of the backdoor in the model. When the number of malicious clients is increased to 4, its ASR grows to 99.40%, which further proves the robustness of our method.

In addition, the dynamic trend of the number of malicious clients on the ASR is shown in Fig. 8. From this figure, there is a significant positive correlation between the increase in the number of malicious clients and the increase in the ASR, and the convergence of the model is accelerated and the attack volatility is reduced.

5) Impact of Total Client Number on Backdoor Attacks: In federated learning, the total number of clients also affects backdoor attacks. This section discusses the impact of different numbers of clients on attack performance. The default settings are consistent with Subsection V-A3, and the experimental results are shown in Fig. 9.

It can be seen that as the total number of clients increases, both the CDA on the main task and its ASR decrease. On the CIFAR10 and Tiny-ImageNet datasets, when the total number of clients reaches 50, the ASR undergoes a significant change, with the distribution dropping from 98.71% and 94% to 54.42% and 6.29%, respectively. On the CelebA dataset, the ASR of the model undergoes a noticeable change when the total number of clients reaches 40, dropping from 97.35% to 79.52%. We believe this change occurs because, as the number of clients increases, the probability of selecting malicious clients in each training round during federated learning decreases, making it difficult to establish effective backdoors in the model.

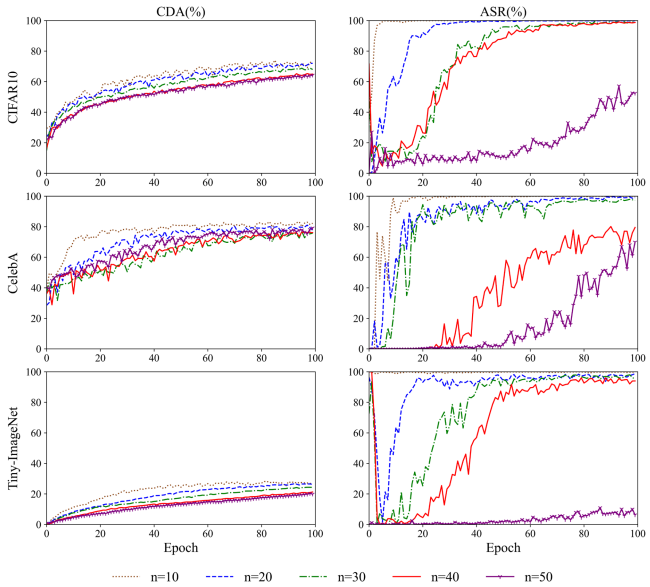


Fig. 9. The impact of total client number on CDA and ASR of our method.

VI. Discussion

A. Possible Countermeasures

In Subsection V-C, we evaluate the robustness of our method to seven defense strategies. The FedDF and FedRAD defense methods in model-refinement defense filter models through distillation during the federated aggregation process. And the FedMV removes backdoors from models by pruning and erasing weights.

In robust-aggregation defense strategies, RobustLR defends by reducing reverse gradient updates, DeepSight defends by filtering malicious clients through clustering, and AlignIns suppresses backdoor attacks by filtering abnormally aligned model updates. Among these, the first five defense methods rely on significant changes in the parameters of the backdoor model, so our method can bypass them. The abnormal alignment relied upon by AlignIns can be avoided by modifying the unimportant model parameters of our method.

The CRFL method achieves parameter smoothing by cropping the global model parameters and adding Gaussian noise during the training process. Where the addition of Gaussian noise can have a significant impact on the model's classification performance, thus we believe that our method can be defended by using the CRFL method with a large Gaussian noise. However, it should be noted that the CRFL method with higher Gaussian noise leads to a significant decrease in the clean data accuracy of the model. Therefore, the noise parameters of the CRFL method need to be carefully adjusted to realize the trade-off between CDA and ASR.

B. Limitations

Our method is a model backdoor attack for federated learning scenarios. Due to the limitations of the experimental equipment, we evaluated the effectiveness of our method on three typical visual datasets. Nevertheless, we believe that our method is effective for any federated visual classification system. However, it is difficult to apply our method to visual classification systems in a centralized environment. The reason is that it is almost impossible for an attacker in a realistic scenario to completely control the training process of a centralized classification system.

VII. Conclusion

In this paper, we discuss a model backdoor attack method for the federated learning system. This method can effectively implant and retain backdoor features without compromising the overall performance of the model through a process of finely selection and manipulation of model parameters. We provide detailed introductions to various key modules constituting the attack process: the parameter importance analysis module, the activation difference computation module, and the parameter manipulation flipping module. Each module aims to precisely manipulate and adjust parameters related to the backdoor in the model to enhance response to backdoor triggers while reducing the risk of detection. Subsequently, we conduct a series of experiments to validate the effectiveness of the proposed backdoor attack method in a federated learning environment. Meanwhile, we systematically examine the robustness of the attack method using seven advanced federated learning backdoor defense mechanisms. The experimental results demonstrate that the attack method successfully circumvents various defense measures, highlighting the importance of careful considerations in the precise selection and manipulation of model parameters as discussed in this paper. Additionally, for the widespread Non-IID data problem in federated learning systems, we conduct in-depth real parameter analysis experiments, which further confirm the challenge posed by the Non-IID problem to the security and model generalization ability of federated learning.

The content of this paper not only reveals the potential dangers of backdoor attacks in federated learning, but also emphasizes the urgency of achieving secure and reliable federated learning systems in practical environments. Through the analysis presented in this paper, the proposed method is not only theoretically innovative, but its effectiveness and stealthiness in practical application are also fully verified.

References

[1] K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[2] Y. Leng et al., "FastCorrect: Fast error correction with edit alignment for automatic speech recognition," *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 21708–21719.
[3] J. Bragg et al., "FLEX: Unifying evaluation for few-shot NLP," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 15787–15800.
[4] B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[5] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng.*, 2019, pp. 374–380.
[6] S. Mahloujifar, M. Mahmoody, and A. Mohammed, "Universal multi-party poisoning attacks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4274–4283.

[7] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–36, 2021.
[8] H. Wang et al., "Attack of the tails: Yes, you really can backdoor federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 16070–16084.
[9] A. N. Bhagoji et al., "Model poisoning attacks in federated learning," in *Proc. Workshop Secur. Mach. Learn. 32nd Conf. Neural Inf. Process. Syst.*, 2018, pp. 1–23.
[10] A. N. Bhagoji et al., "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, PMLR, 2019, pp. 634–643.
[11] M. Fang et al., "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1605–1622.
[12] N. Tajbakhsh et al., "Convolutional neural networks for medical image analysis: Full training or fine tuning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.
[13] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*, 2018, pp. 273–294.
[14] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," in *Proc. 23rd Int. Symp. Res. Attacks, Intrusionsdefenses (RAID 2020)*, 2020, pp. 301–316.
[15] S. P. Karimireddy et al., "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.
[16] T. Li et al., "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.
[17] Z. Wang, H. Xu, J. Liu, Y. Xu, H. Huang, and Y. Zhao, "Accelerating federated learning with cluster construction and hierarchical aggregation," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 3805–3822, Jul. 2023.
[18] D. Gao et al., "HHHFL: Hierarchical heterogeneous horizontal federated learning for electroencephalography," *NeurIPS Workshop Federated Learn. Data Privacy Confidentiality*, pp. 1–7, 2019.
[19] Y. Liu et al., "A communication efficient collaborative learning framework for distributed features," *IEEE Trans. Signal Process.*, pp. 1–7, 2019.
[20] S. Sharma, C. Xing, Y. Liu, and Y. Kang, "Secure and efficient federated transfer learning," in *Proc. IEEE Int. Conf. Big Data*, 2019, pp. 2569–2576.
[21] C. Zhang et al., "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, 2021, Art. no. 106775.
[22] V. Tolpegin et al., "Data poisoning attacks against federated learning systems," in *Proc. Eur. Symp. Res. Comput. Secur.*, Guildford, U.K., Springer, 2020, pp. 480–501.
[23] C. Xie et al., "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Representations*, Addis Ababa, Ethiopia, 2020, pp. 1–15.
[24] B. Wang, F. Yu, F. Wei, Y. Li, and W. Wang, "Invisible intruders: Label-consistent backdoor attack using re-parameterized noise trigger," *IEEE Trans. Multimedia*, vol. 26, pp. 10766–10778, 2024.
[25] E. Bagdasaryan et al., "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, Palermo, Sicily, Italy, PMLR, 2020, pp. 2938–2948.
[26] Z. Zhang et al., "Neurotoxin: Durable backdoors in federated learning," in *Proc. Mach. Learn. Res.*, Baltimore, MD, USA, 2022, vol. 162, pp. 26429–26446.
[27] P. Fang and J. Chen, "On the vulnerability of backdoor defenses for federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 10, pp. 11800–11808.
[28] X. Lyu et al., "Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 7, pp. 9020–9028.
[29] T. D. Nguyen et al., "Poisoning attacks on federated learning-based iot intrusion detection system," in *Proc. Workshop Decentralized IoT Syst. Secur.*, San Diego, CA, USA, 2020, vol. 79, pp. 1–7.
[30] B. Hou et al., "Mitigating the backdoor attack by federated filters for industrial IoT applications," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3562–3571, May 2022.
[31] Z. Gu and Y. Yang, "Detecting malicious model updates from federated learning on conditional variational autoencoder," in *Proc. Int. Parallel Distrib. Process. Symp.*, Portland, OR, USA, 2021, pp. 671–680.
[32] J. Gao, B. Zhang, X. Guo, T. Baker, M. Li, and Z. Liu, "Secure partial aggregation: Making federated learning more robust for industry 4.0 applications," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6340–6348, Sep. 2022.
[33] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, 2021, vol. 35, no. 10,pp. 9268–9276.
[34] C. P. Wan and Q. Chen, "Robust federated learning with attack-adaptive aggregation," in *Proc. Int. Workshop Federated Learn. User Privacy Data Confidentiality Conjunction IJCAI (FL-IJCAI)*, 2021, pp. 1–14.
[35] T. Lin et al., "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Virtual, Online, 2020, vol. 33, pp. 2351–2363.
[36] S. P. Sturluson et al., "FedRAD: Federated robust adaptive distillation," *NeurIPS 2021 Workshop New Frontiers Federated Learn. (NFFL)*, pp. 1–14, 2021.
[37] C. Wu et al., "Mitigating backdoor attacks in federated learning," 2020, *arXiv:2011.01767*.
[38] P. Rieger et al., "DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection," in *Proc. Netw. Distrib. System Secur. Symp.*, Hybrid, San Diego, CA, USA, 2022, pp. 1–18.
[39] H. Yang, W. Xi, Y. Shen, C. Wu, and J. Zhao, "RoseAgg: Robust defense against targeted collusion attacks in federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 2951–2966, 2024.
[40] J. Xu, Z. Zhang, and R. Hu, "Detecting backdoor attacks in federated learning via direction alignment inspection," in *Proc. Comput. Vis. Pattern Recognit. Conf.*, 2025, pp. 20654–20664.
[41] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2018, pp. 4447–4458.
[42] N. Ivkin et al., "Communication-efficient distributed SGD with sketching," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2019, vol. 32, pp. 1–11.
[43] C. Xie et al., "CRFL: Certifiably robust federated learning against backdoor attacks," in *Proc. Int. Conf. Mach. Learn.*, Virtual, Online, 2021, vol. 139, pp. 11372–11382.
[44] P. Blanchard et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 1–11.
[45] D. Yin et al., "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.
[46] R. Guerraoui, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.

Bo Wang (Member, IEEE) received the BS degree in electronic and information engineering, and the PhD degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2003 and 2010, respectively. From 2010 to 2012, he was a postdoctoral research associate with the Faculty of Management and Economics, Dalian University of Technology. From 2017 to 2019, he was with the State of University of New York at Buffalo, as a visiting scholar. He is currently a professor with the School of Information and Communication Engineering, Dalian University of Technology. His research interests include artificial intelligence security and multimedia security.

Yiming Yan received the BS degree in electronic and information engineering from the Dalian University of Technology, China, in 2023. He is currently working toward the master's degree with the School of Information and Communication Engineering, Dalian University of Technology. His research focuses on image and audio oriented backdoor attacks and defenses.

Maozhen Zhang received the MS degree from the School of Computer Science and Technology, Dalian Maritime University, in 2022. He is currently working toward the PhD degree with the School of Information and Communication Engineering, Dalian University of Technology. His research interests include artificial intelligence security, especially federated learning, privacy security, deep learning, and backdoor attacks and defenses.

Wei Wang (Member, IEEE) received the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2012. He is currently an associate professor with the New Laboratory of Pattern Recognition (NLPR) and State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), CASIA. His research interests include artificial intelligence safety and multimedia forensics.

Hongwei Yao received the PhD degree from the College of Computer Science and Technology, Zhejiang University, in 2024, advised by Dr. Zhan QIN. He is currently a postdoctoral fellow with the City University of Hong Kong, advised by Prof. Cong Wang. His research interests include trustworthy AI, LLM security, and safety.