FedAMM: Federated Learning Against Majority Malicious Clients Using Robust Aggregation

Keke Gai, *Senior Member, IEEE*, Dongjue Wang, Jing Yu, *Member, IEEE*, Liehuang Zhu, *Senior Member, IEEE* and Weizhi Meng, *Senior Member, IEEE*

***Abstract*—As a privacy-preserving collaborative learning framework, *Federated Learning* (FL) aims at protecting participants' privacy during model training. However, the framework currently encounters vulnerabilities deriving from poisoning attacks due to the unregulated nature of local training. Most existing solutions address scenarios where less than half of clients are malicious, i.e., which leaves a significant challenge to defend attacks when more than half of participants are malicious. In this paper, we propose a FL scheme, named FedAMM, that resists backdoor attacks across various data distributions and malicious client ratios. We develop a novel backdoor defense mechanism to filter out malicious models while minimizing the impact on model performance. The proposed scheme addresses the challenge of distance measurement in high-dimensional spaces by applying *Principal Component Analysis* (PCA) to improve clustering effectiveness. We borrow the idea of critical parameter analysis to enhance discriminative ability in non-iid data scenarios, via assessing the benign or malicious nature of models by comparing the similarity of critical parameters across different models. Finally, our scheme employs a hierarchical noise perturbation to improve the backdoor mitigation rate, which effectively removes the backdoor while minimizing the impact of the noise on the accuracy of the task. Our experiments on multiple datasets show that the proposed scheme outperforms existing methods in backdoor defense across diverse client data distributions and varying proportions of malicious clients. When 80% of the participating clients are malicious, FedAMM achieves low backdoor attack success rates of 1.14%, 0.28%, and 5.53% on the MNIST, FMNIST, and CIFAR-10 datasets, respectively, demonstrating enhanced robustness of FL against backdoor attacks.**

***Index Terms*—Federated learning, backdoor attack, robust aggregation, non-iid data, majority malicious clients.**

I. Introduction

PRIVACY protection is one of the major concerns in data collaborative training and *Federated Learning* (FL) is an option for addressing privacy issues due to its ability of enabling multi-party jointly training model without sharing local data [1]–[3]. A typical FL framework [4]–[8] consists of multiple clients performing local optimizations and a server conducting global model aggregation. The server remains unaware of the local training processes, thus safeguarding privacy [9]. The technique has attracted attention from various fields in recent years, such as healthcare [10], [11] and social governance [12].

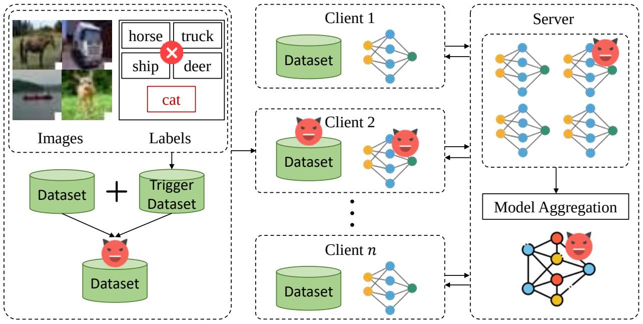


Fig. 1. The backdoor attacks of the FL training process.

However, the training paradigm of FL makes it susceptible to data and model poisoning attacks initiated by malicious clients. Backdoor attacks [13]–[16] pose threatening poisoning attack, exhibiting strong stealthiness and harm. Fig. 1 illustrates threats of backdoor attacks to FL training process. Adversaries executing backdoor attacks inject backdoors into the global model by adding triggers during the training process, manipulating the global model to make incorrect decisions on samples containing triggers. Ensuring the trustworthiness of the global model through FL global model robust aggregation is crucial for resisting backdoor poisoning attacks.

Prior defense mechanisms [17]–[19] against backdoor attacks typically rely on updates of model parameters submitted by clients to identify poisoned parameters. For instance, FLTrust [17] is a defense mechanism that utilizes trust scores, relying on a clean dataset on the server side to identify malicious clients providing poisoned model parameters. The collection of clean datasets is hindered in practical applications due to concerns about client privacy infringement. To reduce reliance on clean datasets, Nguyen *et al.* [18] proposed FLAME, a backdoor attack defense scheme utilizing HDBSCAN [20] clustering and model parameter pruning, so that client model parameters are clustered in terms of cosine distances in an unsupervised manner. Unfortunately, this scheme is only applicable when malicious clients are in the minority, as it selects the largest clustering cluster for aggregation during global model aggregation. Moreover, FLAME encounters obstacles in scenarios with non-iid data distributions due to minimal differences between benign and malicious model parameters. To enhance defense mechanisms' perfor-

K. Gai, D. Wang, and L. Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China. (E-mails: {gaikeke, 3220231818, liehuangz}@bit.edu.cn).

J. Yu is with School of Information Engineering, Minzu University of China, Beijing, China, email: jing.yu@muc.edu.cn

W. Meng is with School of Computing and Communications, Lancaster University, United Kingdom. Email: weizhi.meng@ieee.org.

This work is supported by the National Natural Science Foundation of China (Grant No.s U24B20146 and 62372044).

Corresponding author: J. Yu (jing.yu@muc.edu.cn).

mance in non-iid data distribution scenarios, FreqFed [21], as a frequency domain analysis technique, was proposed to transform model parameters into the frequency domain to extract more model weight information. The drawback of FreqFed is that this scheme only applies to scenarios with few malicious clients.

Therefore, existing defense mechanisms exhibit limitations in resisting backdoor attacks from a large number of malicious clients in non-iid data distribution scenarios. In scenarios involving data heterogeneity and a predominance of malicious clients, developing a FL method to defend against backdoor poisoning attacks and ensure the security and reliability of the global model presents a critical challenge.

To address the aforementioned issues, we integrate density-based clustering with critical parameter analysis to enhance the effectiveness of the defense scheme in scenarios with a majority of malicious clients. Specifically, the proposed backdoor defense scheme maintains following capabilities. First, to ensure its effectiveness when the client data distribution is unknown, the proposed scheme is able to effectively identify features that differ and are independent of client data distribution in both benign and malicious models. Additionally, our scheme improves the clustering accuracy for model selection and optimize the cluster selection strategy, thus enhancing the scheme's effectiveness in scenarios with a majority of malicious clients. Critical parameter analysis plays a crucial role in filtering out malicious clusters. Different model parameters have varying impacts on model performance, and studies [22] have shown that the critical parameters most affecting performance differ between benign and poisoned models, independent of client data distribution. Using this insight, we can identify clusters with benign models, enabling a robust FL aggregation scheme adaptable to different scenarios.

In this paper, we propose a robust FL aggregation algorithm that combines OPTICS (Ordering Points To Identify the Clustering Structure) [23] clustering with critical parameter analysis, called *Federated Learning Against Majority Malicious Clients Using Robust Aggregation* (FedAMM). To solve the problem of distance measurement failure in high-dimensional spaces, we apply principal component analysis to reduce the dimensionality of model parameters. We measure the cosine and Euclidean distances between the reduced low-dimensional model parameters to make differences between model parameters more visible. We use the cosine distance between the reduced model parameters for OPTICS clustering, improving robustness to noisy data and outliers. Since in the scenarios where malicious models are in the majority, we observe that malicious models contain similar top-k and bottom-k critical parameters, we combine critical parameter analysis to enhance the ability of FedAMM to detect malicious models. FedAMM evaluates client models by comparing the similarity of critical parameter sets from each training round. By combining density-based clustering with critical parameter similarity analysis, FedAMM overcomes limitations of existing defense methods constrained by the proportion of malicious clients, enhancing its practicality.

The main contributions are summarized as follows.

1) In this work, we propose a FL scheme that addresses the scenario with an assumption that more than half of the clients are malicious. The proposed scheme is a server-side defense strategy that ensures the effectiveness and robustness of the global model and avoids relying on a clean dataset.
2) To resist backdoor poisoning attacks from numerous malicious clients, we propose a robust aggregation approach using adaptive clustering and critical parameter analysis. Differing from existing strategies, the proposed approach can exclude most malicious local models and has a better performance in ensuring accuracy and robustness of the global model.
3) Our experiment evaluations also have evidenced that the proposed scheme achieves state-of-the-art defense performance under various data distributions and varying proportions of malicious clients, demonstrating its effectiveness and practicality.

TABLE I
NOTATIONS AND CORRESPONDING DEFINITIONS

Notation	Definition
D	The dataset fot the FL task
T	The total training rounds
S	The central server participating in the aggregation
C	The total set of clients
C_t	The total set of clients participating in round t
K_t	The size of total set of clients participating in round t
D_k	The local dataset of client k
G_t	The global model of round t
n_t	The number of samples for all clients participating in round t
n_t^k	The number of samples for client k participating in round t
θ_k^t	The local update of client k during the round t
Θ_k^{top}	The top-k parameter sets of client k
Θ_k^{bottom}	The bottom-k parameter sets of client k

The rest of this paper is organized as follows. Preliminaries are given in Section II. In Sections III and IV, we provide detailed descriptions about the proposed model and theoretical analysis, respectively. Section V present experiment evaluations and main findings. Related work is given in Section VI. Finally, conclusions of this work are drawn in Section VII.

II. PRELIMINARIES

A. Federated Learning

FL is a distributed machine learning paradigm designed to ensure data privacy. Table I summarizes the notations utilized in this paper. It typically involves multiple clients C that conduct local training and a central server S responsible for aggregating the global model. During each training round $t \in \{1, \ldots, T\}$, the server S sends the previous global model G_t to a randomly sampled clients $C_t \subset C$ of size K_t. Each client $k \in C_t$ trains G_t using local data D_k and send updates θ_k^t to the server. Then, the server S aggregates updates to obtain a updated global model G_{t+1}. FedAvg [4] is a widely used global model aggregation method for horizontal federated learning [24], and its aggregation process is as follows:

$$G_{t+1} = G_t + \sum_{k=1}^{K_t} \frac{n_t^k}{n_t} \theta_k^t, \tag{1}$$

where n_t^k is the number of samples for client k participating in round t, and n_t is the number of samples for all clients participating in round t.

B. Backdoor Attacks in FL

Backdoor attacks [13]–[16], [25] are active threats that significantly compromise the security of FL. Adversaries carry out these attacks by injecting malicious data or altering models to change the decision boundaries of the FL model, ultimately diminishing its effectiveness. Adversaries conducting backdoor attacks embed triggers into original samples to generate malicious samples, which are then used to train the model. Through model aggregation, global model learning will learn the association between backdoor triggers and incorrect labels. These attacks seek to maximize the *Attack Success Rate* (ASR), or in the backdoor context, the *Backdoor Accuracy* (BA). For a FL task, the adversary aims to maximize the backdoor accuracy, defined in Eq. (2), where $\mathcal{D}$ is the dataset for the FL task, y is the label for input x, y' is the target label, and $t(\cdot)$ is the backdoor trigger function.

$$\mathrm{BA} = \underset{\substack{(x,y)\sim\mathcal{D} \\ y\neq y'}}{\mathbb{E}} \left[\Pr\left(\Phi(t(x)) = y'\right)\right], \tag{2}$$

C. Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised method for reducing data dimensionality by linearly mapping high-dimensional data to a low-dimensional space while retaining the most important principal components. Main steps of PCA are as follows. (1) Represent m data points in n-dimensional space as an $n \times m$ matrix $\mathcal{H}$. (2) Subtract the mean of each row from each element in that row of matrix $\mathcal{H}$, resulting in matrix $\mathcal{H}'$. (3) Compute the covariance matrix $\mathcal{V}_m$ of matrix $\mathcal{H}'$, defined as $\mathcal{V}_m = \frac{1}{m}\mathcal{H}\mathcal{H}'$. (4) Compute the eigenvalues and eigenvectors of the covariance matrix $\mathcal{V}_m$. (5) Sort the eigenvectors by eigenvalues in descending order to a matrix $\mathcal{P}$. (6) Obtain the reduced-dimensional matrix $\mathcal{Y} = \mathcal{P}^T\mathcal{H}$. PCA effectively uncovers the internal structure of the data by analyzing its eigenvalues, providing a better representation of the data's variability.

D. OPTICS Clustering

OPTICS [23] is a density-based clustering algorithm for overcoming limitations of DBSCAN's [26] in sensitivity to parameter settings. OPTICS creates an ordered list of points, each annotated with a reachability distance and a core distance. A core distance represents the smallest radius needed for a point to qualify as a core point, while a reachability distance is the smallest distance that allows a point to be density-reachable from a core point. Unlike DBSCAN, OPTICS does not require a global density threshold to define clusters. Instead, it identifies clusters with varying densities by analyzing the reachability distances in the ordered list. OPTICS is robust to noise and outliers, determining cluster membership through density estimation rather than fixed distance thresholds. It makes OPTICS well-suited for large-scale clustering tasks involving irregular shapes and varying densities.

III. Our Proposed Model

A. Design Goals

Our primary goal is to safeguard the global model in FL from backdoor attacks by adversaries. We aim to ensure the global model's performance by implementing a robust aggregation algorithm on the server side to defend against backdoor attacks. Design objectives are given as follows.

Robustness: The primary goal of FedAMM is to enhance the robustness of the global model to ensure its utility remains intact. Existing defense strategies based on outlier detection often fail in Non-IID data scenarios by mistakenly identifying benign models as malicious, thus compromising the task accuracy. Specifically, FedAMM must tolerate some degree of poisoning during local model training while maintaining the proper execution of the aggregation process. Additionally, FedAMM aims to minimize the participation of malicious local models in the aggregation, thereby mitigating their impact on the global model.

Scalability: Addressing effective backdoor defense under unknown preconditions, proportion of malicious clients and the distribution of dataset samples are important parameters affecting the effectiveness of backdoor defenses. Previous methods are limited by variations in data distributions, poisoning rates, and the proportion of malicious clients, generally failing when over half the clients are malicious. Specifically, FedAMM needs to overcome the limitation of the proportion of malicious clients and achieve global model robust aggregation suitable for various types of data distributions.

Effectiveness: Strengthening the effectiveness of backdoor defense strategies is an important goal of FedAMM. The success rate of backdoor attacks is a key indicator for evaluating effectiveness. Due to the stealthy nature of backdoor poisoning attacks, accurately identifying backdoor samples with triggers is challenging. Preventing the global model from learning malicious trigger patterns is crucial for strengthening backdoor defenses. Specifically, FedAMM needs to reduce or eliminate the effects of poisoned model updates, thereby lowering the success rate of backdoor attacks.

B. Threat Model

Existing research [18], [22] have evidenced that some threats to FL derive from the decentralization of the training framework, as adversaries are supposed to control several clients involved in the FL model training, covering local datasets and training parameters. Considering the maintenance of generality, we assume that adversaries cannot manipulate the server's aggregation behavior and are unaware of the server's defense strategies or insights. Adversaries' goals and capabilities are based on the following assumptions.

1) Adversary's Goal.: The goal of the adversary is to embed a backdoor into the global model by poisoning the training dataset, thereby causing the model to behave as intended by the attacker. Specifically, the backdoored global model performs normally on clean inputs but misclassifies inputs containing a trigger as the attacker's target class.

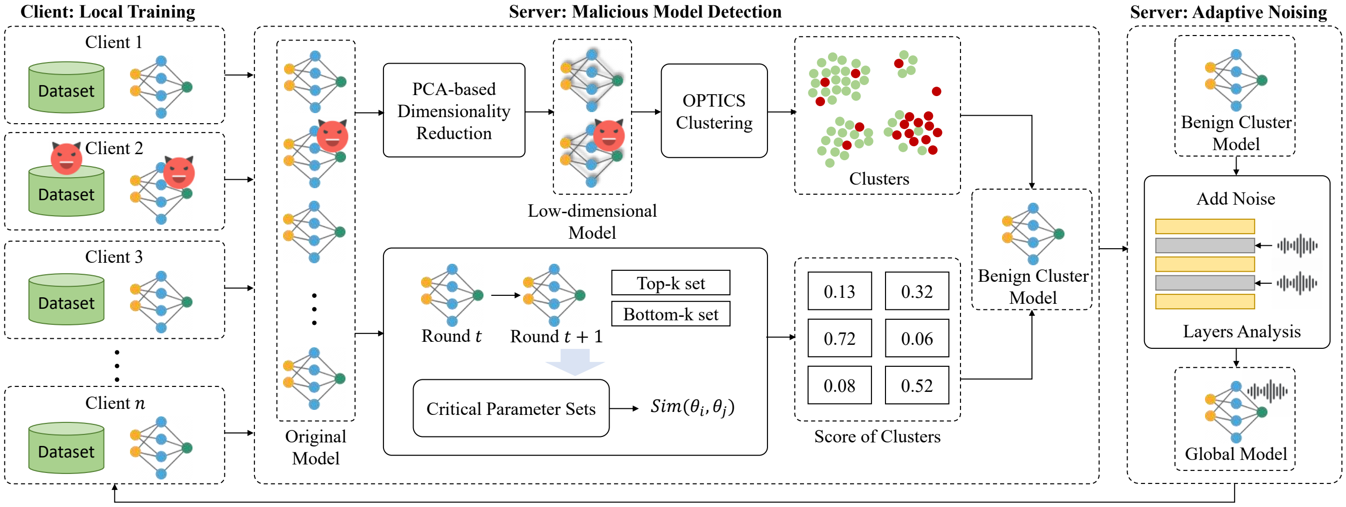


Fig. 2. The framework of FedAMM.

```
Algorithm 1 FedAMM
Input: Initial global model $G_1$, the number of rounds $T$, $N$ clients
Output: Global model $G_{t+1}$
for each training iteration $t \in [1, T]$ do
    for each client $i \in C_t$ do
        $\theta_i^t \leftarrow$ Client Update$(G_t, i)$
    end for
    $(\mathcal{N}(\theta_1^t), \dots, \mathcal{N}(\theta_N^t)) \leftarrow$ Critical Parameter Analysis$(\theta_1^t, \dots, \theta_N^t, G_t, G_{t-1})$
    $({\theta_1^t}', \dots, {\theta_N^t}') \leftarrow$ PCA$(\theta_1^t, \dots, \theta_N^t)$
    $(c_{11}, \dots, c_{nn}) \leftarrow$ Cosine Distence$({\theta_1^t}', \dots, {\theta_N^t}')$
    $(\phi_1, \dots, \phi_l) \leftarrow$ Adaptive Clustering$(c_{11}, \dots, c_{nn})$
    for each cluster $\phi_i, i \in [1, l]$ do
        $\mathcal{N}(\theta_{\phi_i}) \leftarrow \sum_{k=1}^{len(\phi_i)} \mathcal{N}(\theta_k^t), \theta_k^t \in \phi_i$
    end for
    $\mathcal{N}(\theta_{\phi_{benign}}) \leftarrow min(\{\mathcal{N}(\theta_{\phi_i}), i \in [1, l]\})$
    $G'_{t+1} = G_t + \sum_{k=1}^{n} \Delta w_k^t / n, \theta_k^t \in \phi_{benign}, n = len(\phi_{benign})$
    $G_{t+1} \leftarrow$ Adaptive Noising$(G'_{t+1}, N(0, \sigma^2))$
    return $G_{t+1}$
end for
```

2) Adversary's Capabilities.: Adversaries operate under the same FL protocol as benign clients and can monitor changes in the global model across multiple training rounds. Adversary can poison the local training dataset by embedding predefined triggers (e.g., small patches or pixel patterns) into a subset of the data and altering the labels of these poisoned samples to a specific target class. Using the backdoor training dataset, adversary can train local model over multiple rounds and subsequently submit the compromised model to the server.

C. Model Design

The framework of the proposed method is shown in Fig. 2. We assume that K clients $i \in C_t$ are selected to participate in training round t. In each round of training, the client trains the global model using the local data set to obtain the local model. The server detects and eliminates the malicious local model and gets the global model through robust aggregation. Specifically, FedAMM consists of five components, including PCA-based parameter dimension, critical parameter analysis, adaptive clustering for model parameters, aggregating of Benign cluster, and adaptive noising. The algorithm of FedAMM is shown in Alg. 1.

1) PCA-based Parameter Dimension: As the number of data dimensions increases, the volume of the data space grows exponentially, making the distances between data points more sparse. This sparsity impacts data distribution representation and reduces the performance of clustering algorithms. In high-dimensional space, traditional distance measurement is no longer valid because the distance between most data points will be close to the maximum distance, and it is difficult to distinguish the similarity between different data points. Moreover, the instability of distances in high-dimensional spaces leads to substantial noise, further degrading clustering performance. In FedAMM, to mitigate the clustering performance degradation caused by the curse of dimensionality, the server employs PCA to reduce the dimensionality of local models received from clients. Specifically, each local model of dimension d is represented as θ_i^t, and the set $(\theta_1^t, \theta_2^t, ..., \theta_k^t)$ includes K high-dimensional data points. The server organizes this data set into a matrix $\mathcal{H}^t \in \mathbb{R}^{d \times K}$. Then, the server uses the dimensionality reduction method described in Section II-C to transform this data set into a low-dimensional matrix. This low-dimensional matrix effectively uncovers the internal structure of the data, thereby enhancing the representation of data variability.

2) Critical Parameter Analysis: The differences between benign models increase in non-IID data scenarios, making it harder to distinguish them from malicious models. Research [27] shows that different model parameters contribute variably to the optimization task. Han *et al.* [22] further found that benign models on clients have similar top-k and bottom-k critical parameters, while malicious models do not. Furthermore, this property, unaffected by data distribution, enhances the ability of backdoor defense mechanisms to identify

Algorithm 2 Client Update

```
Input: Global model $G_t$ at round $t$
Output: Local model update $\theta_i^t$
$\theta_i \leftarrow G_t$
for each epoch $e \in [1, E_i]$ do
    for each batch $b \in \beta$ do
        $\theta_i^t \leftarrow \theta_i - \eta_i \nabla \ell(\theta_i; b)$
    end for
end for
return $\theta_i^t$
```

Algorithm 3 Critical Parameter Analysis

```
Input: Local models $\theta_i^t$ of client $i$ at round $t$, global model
    $G_t$ at round $t$, global model $G_{t-1}$ at round $t-1$
Output: $\mathcal{N}\left(\theta_i^t\right)$
$\Delta_s^t \leftarrow G_t - G_{t-1}$
$p_s[n] \leftarrow |\Delta_s[n] \cdot \theta_s[n]|$
$\Theta_s^{\text{bottom}}, \Theta_s^{\text{top}} \leftarrow \arg\text{sort}\left(p_s^t\right)[:k], \arg\text{sort}\left(p_s^t\right)[-k:]$
for each client $i \in C_t$ do
    $\Delta w_i^t \leftarrow \theta_i^t - G_t$
    $p_i[n] \leftarrow |\Delta w_i[n] \cdot \theta_i[n]|$
    $\Theta_i^{\text{bottom}}, \Theta_i^{\text{top}} \leftarrow \arg\text{sort}\left(p_i^t\right)[:k], \arg\text{sort}\left(p_i^t\right)[-k:]$
    $J\left(\Theta_i^*, \Theta_s^*\right)$ in Eq. (3)
    $Spearman\left(\theta_i, \theta_s\right)^*$ in Eq. (4)
end for
for each client $i \in C_t$ do
    for each client $j \in C_t$, $j \neq i$ do
        $J\left(\Theta_i^*, \Theta_j^*\right)$ in Eq. (3)
        $Spearman\left(\theta_i, \theta_j\right)^*$ in Eq. (4)
    end for
end for
$\mathcal{N}\left(\theta_i^t\right)$ in Eq. (5)
return $\mathcal{N}\left(\theta_i^t\right)$
```

malicious models. However, we find that when the malicious model is more than half of the scenarios, the top-k parameters (bottom-k parameters) of different malicious models are more similar than those of the benign model. Inspired by the above, FedAMM uses critical parameter analysis to solve the problem of malicious model detection failure in scenarios where more than half of the malicious models are distributed and the data is non-IID. Specifically, the server first calculates the change in each parameter of local models between adjacent rounds, denoted as $\Delta w_i^t = \theta_i^t - G_t$, where θ_i^t denote the model parameters of client i at round t, and G_t denote the global model parameters at round t. The server then computes the importance of parameter p_i^t as $p_i[n] = |\Delta w_i[n] \cdot \theta_i[n]|$. Then, the server ranks importance values to identify the top-k and bottom-k parameter sets (i.e., Θ_i^{top}, Θ_i^{bottom}) of model for each client. In the same way, the server calculates the set of top-k parameters Θ_s^{top} and bottom-k parameters Θ_s^{bottom} of the global model G_t. Next, the server evaluates the score of each model based on the top-k parameter set and bottom-k parameter set of the model. The higher the score is, the greater the probability that this model is malicious. The evaluation score is divided into two parts. The first part is used to measure the differences in the top-k and bottom-k parameters between each client model and the global model.

First, we calculate the Jaccard similarity between different parameter sets, as shown in Eq. (3).

$$J\left(\Theta_i^*, \Theta_s^*\right) = \frac{|\Theta_i^* \cap \Theta_s^*|}{|\Theta_i^*| + |\Theta_s^*| - |\Theta_i^* \cap \Theta_s^*|}, * \in \{\text{top,bottom}\}. \tag{3}$$

Next, we evaluate the Spearman correlation coefficients of the top-k parameter and bottom-k parameter of the two models in the common parameter set, as shown in Eq. (4).

$$\begin{aligned} &Spearman\left(\theta_i, \theta_s\right)^* = \rho(r_i(\Theta_i^* \cap \Theta_s^*), r_s(\Theta_i^* \cap \Theta_s^*)), \\ &* \in \{\text{top,bottom}\}, \end{aligned} \tag{4}$$

where $r_i(\cdot)$ denotes the functions that map indices to ranks in terms of parameter importance for client i, $\rho(\cdot)$ denotes the Spearman correlation. The scores associated with the global model of the local model are obtained by weighting the Jaccard similarity and the Spearman correlation coefficient. In the second part, we evaluate the differences in the top-k and bottom-k parameters among different client models in the same way to obtain the score of the second part. Scores obtained from the analysis of critical parameters of each client are defined in Eq. (5).

$$\begin{aligned} &\mathcal{N}\left(\theta_i^t\right) = \lambda \frac{1}{|C_t|} \sum_{j \in C_t} \left(J\left(\Theta_i^*, \Theta_j^*\right) + Spearman\left(\theta_i, \theta_j\right)^*\right) + \\ &\beta(J\left(\Theta_i^*, \Theta_s^*\right) + Spearman\left(\theta_i, \theta_s\right)^*), * \in \{\text{top,bottom}\}, \end{aligned} \tag{5}$$

where λ and β are the weights of the scores.

3) Adaptive Clustering for Model Parameters: In FedAMM, the server responsible for global model aggregation lacks a clean dataset and therefore cannot use dataset-based validation strategies [17] to evaluate the trustworthiness of local models. Clustering, as an unsupervised learning method, offers a promising solution to address the limitation of clean datasets. Existing methods [18], [21] use the HDBSCAN algorithm to differentiate between malicious and benign clients. However, HDBSCAN [20] cannot reliably identify malicious models due to the absence of distance constraints. DBSCAN [26], which includes clear distance criteria, is a more effective and accurate method for filtering out malicious clients. OPTICS [23] improves upon DBSCAN by addressing its sensitivity to parameter settings. In FedAMM, the server first calculates the cosine distance between local models after dimensionality reduction using PCA. Then, the server sets the minimum cluster size for the OPTICS algorithm based on the number of local models participating in the current aggregation round. Finally, the server derives multiple clusters through adaptive OPTICS clustering. Unlike Flame [18] and FreqFed [21], which produce only two clusters, multiple clusters enhance the identification of malicious models, especially when there are many malicious models involved. By combining multiple clusters with the critical parameter scores of each cluster, FedAMM can better identify and exclude malicious models, thereby preserving the effectiveness of the global model.

4) Aggregation of Benign Cluster: After clustering, FedAMM assesses the importance score of each cluster. Specifically, assuming the i-th cluster ϕ_i contains $len(\phi_i)$ local updates during the round t. The score for each cluster is computed as shown in Eq. (6).

$$\mathcal{N}(\theta_{\phi_i}) = \sum_{k=1}^{len(\phi_i)} \mathcal{N}(\theta_k^t), \theta_k^t \in \phi_i \tag{6}$$

The higher the score of a single model is, the greater the probability that the model is a malicious one. Therefore, the server certifies the cluster with the lowest score as a benign cluster. Subsequently, FedAMM applies boundary clipping to the client model parameters within the benign cluster. The server computes the median of the L_2 norms of all models in the benign cluster as the clipping boundary B_t, as described in Eq. (7).

$$\begin{aligned} B_t \leftarrow Median(\|\Delta_1^t\|_2, \|\Delta_2^t\|_2, \ldots, \|\Delta_k^t\|_2), \\ \Delta w_k^t = \theta_k^t - G_T, \theta_k^t \in \phi_i. \end{aligned} \tag{7}$$

The clipping boundary selected by FedAMM ensures that the majority of normal model parameters in the benign cluster are not excessively scaled, while limiting extreme updates that may potentially contain backdoors. Furthermore, the server applies clipping to the client models, as illustrated in Eq. (8).

$$\overline{\Delta w_k^t} = \frac{\Delta w_k^t}{\max\left(1, \|\Delta w_k^t\|_2 / B_t\right)}, \|\overline{\Delta w_k^t}\|_2 \le B_t \tag{8}$$

Upon completing the boundary clipping of the model parameters, FedAMM computes the aggregated global model using average aggregation, as depicted in Eq. (9).

$$\theta_{\phi_i} = G_{t-1} + \sum_{k=1}^{n} \Delta_k^t / n, \theta_k^t \in \phi_{benign}, n = len(\phi_{benign}). \tag{9}$$

5) Adaptive Noising: Prior studies have demonstrated that adding noise to model weights can effectively mitigate the impact of poisoned samples [28]. In distributed settings, client-side backdoor poisoning attacks can be addressed by injecting noise into the global model, thereby reducing its sensitivity to backdoor triggers [18]. Flame [18] uses a similar method by injecting noise into the global model to diminish the impact of backdoor triggers. However, injecting noise into the global model can lower the accuracy of the main task, reducing the model's overall utility. FedAMM evaluates the impact of each network layer on the global model by calculating the average change in weight parameters of each network layer between adjacent rounds. We believe that network layers exhibiting larger average changes between adjacent rounds have a more significant influence on global model accuracy. Therefore, FedAMM injects noise into layers with smaller average changes to minimize its impact on primary task accuracy. Additionally, determining the appropriate noise level to effectively neutralize trigger effects is a critical challenge. FedAMM performs the following steps to compute noise sensitivity. Before aggregating the models in the benign cluster, FedAMM uses the median of the L_2 norms of all models within the cluster as the clipping boundary B_t. The server then obtains the global model via average aggregation, as shown in Eq. (9). We consider average aggregation as a mapping f, for which the L_2 sensitivity (see Def. IV) is defined. Specifically, the mapping f is shown as Eq. (10).

$$f(X) = \frac{1}{n} \sum_{i=1}^{n} \Delta w_i^t \in \mathbb{R}^d, X = \left\{\Delta w_i^t\right\}_{i=1}^{n} \tag{10}$$

Accordingly, the L_2 sensitivity of mapping f is derived as shown in Eq. (11).

$$\Delta_2 f = \max_{X \sim X'} \|f(X) - f(X')\|_2 = \frac{2B_t}{n} \tag{11}$$

The detailed derivation is presented in Lemma. 1. Finally, we effectively neutralize the impact of backdoor triggers by injecting noise into the global model on a per-layer basis. The noise injection process is shown as Eq. (12).

$$\mathcal{M}(X) = f(X) + \xi, \quad \xi_j \sim \begin{cases} \mathcal{N}\left(0, \sigma^2\right), & j \in \mathcal{L} \\ 0, & j \notin \mathcal{L} \end{cases} \tag{12}$$

where $\sigma > \frac{1}{\epsilon}\sqrt{2ln\frac{1.25}{\delta}} \cdot \Delta_2 f$, and $\mathcal{L}$ denotes the layer that needs noise addition.

IV. Theoretical Analysis

Definition 1 (L_2-sensitivity)**.** *: For an arbitrary function f and two adjacent inputs $x, x' \in Dom(f)$, the l_2-sensitivity of f is $\Delta_2 f = \max \|f(x) - f(x')\|_2$.*

Definition 2 (Gaussian Mechanism)**.** *For any $\epsilon > 0$, $\delta \in (0,1)$, there exists noise $Y \sim \mathcal{N}(0, \sigma^2)$ that ensures (ϵ, δ)-Differential Privacy where $\sigma \geqslant \frac{\Delta_2 f \cdot \sqrt{2\ln(1.25/\delta)}}{\epsilon}$.*

Theorem 1. *A (ϵ, δ)-differentially private model is backdoor-free if random Gaussian noise is added to the model parameters yielding a noised model $G^* \leftarrow G + N(0, \sigma_G^2)$ where $\sigma_G \leftarrow \frac{1}{\epsilon}\sqrt{2ln\frac{1.25}{\delta}} \cdot B_t$ and B_t denotes the clipping bound.*

Lemma 1. *Define $f(X) = \frac{1}{n}\sum_{i=1}^{n} \Delta w_i^t \in \mathbb{R}^d, X = \{\Delta w_i^t\}_{i=1}^{n}$ where each Δw_i^t is clipped so that $\|\Delta w_k^t\|_2 \le B$. Then the l_2 sensitivity of f is*

$$\Delta_2 f = \max_{X \sim X'} \|f(X) - f(X')\|_2 = \frac{2B}{n}. \tag{13}$$

We consider two neighboring datasets X and X', which differ in exactly one client update, say the j-th. The difference between the outputs of f on these two datasets is

$$f(X) - f(X') = \frac{1}{n}\left(\bar{\Delta} w_j - \bar{\Delta} w_j'\right) \tag{14}$$

By construction, both $\bar{\Delta} w_j$ and $\bar{\Delta} w_j'$ are individually bounded in L_2 norm by B, so their difference is bounded by $2B$. Hence, the maximum possible change in the output is bounded by $\|\bar{\Delta} w_j - \bar{\Delta} w_j'\|_2 / n \le 2B/n$. Moreover, this upper bound is tight, as it can be achieved when the two vectors are antiparallel and both have norm exactly B. Therefore, the sensitivity of f is exactly $\Delta_2 f = 2B/n$.

Lemma 2. *Let the mechanism*

$$\mathcal{M}(X) = f(X) + \xi, \quad \xi_j \sim \begin{cases} \mathcal{N}\left(0, \sigma^2\right), & j \in \mathcal{L}, \\ 0, & j \notin \mathcal{L}. \end{cases} \qquad (15)$$

if $\sigma > \frac{1}{\epsilon}\sqrt{2ln\frac{1.25}{\delta}} \cdot \Delta_2 f$, then $\mathcal{M}$ satisfies (ϵ, δ)-Differential Privacy with respect to any single-client update and thus prevents a participant from determining whether their own update was included in the aggregation. Consequently, an adversary cannot distinguish whether its own update was included based on $\mathcal{M}(X)$.

From Lemma. 1, the function f has L_2 sensitivity $\Delta_2 f = \frac{2B}{n}$. According to the standard Gaussian mechanism, adding noise drawn from $\mathcal{N}\left(0, \sigma^2\right)$ to each coordinate of a function with sensitivity $\Delta_2 f$ ensures (ϵ, δ)-Differential Privacy, provided that $\sigma > \frac{1}{\epsilon}\sqrt{2ln\frac{1.25}{\delta}} \cdot \Delta_2 f$. Although noise is only applied on a subset $\mathcal{L}$ of coordinates, differential privacy is still preserved as long as the sensitivity analysis accounts for this, and the rest of the coordinates are released deterministically. Differential privacy allows for such partially noised outputs. Therefore, the mechanism $\mathcal{M}$ guarantees (ϵ, δ)-DP with respect to any individual user's participation. As a result, even an adversary who uploads a backdoored update cannot infer from the output whether their update was actually included in the aggregation, achieving the desired data-hiding property.

We present a formal analysis of the noise boundary required to neutralize backdoor triggers in the global model. Let $\mathcal{L}$ denote the set of layers in which noise is added, and $\mathcal{M}(X) = f(X) + \xi$ be the global aggregation mechanism, where $f(X)$ is the clipped average of local updates $\bar{\Delta} w_j$ and $\xi_j \sim \mathcal{N}\left(0, \sigma^2\right)$ for $j \in \mathcal{L}$, and zero otherwise.

Theorem 2. *Let $\mathcal{M}(X)$ be a Gaussian mechanism satisfying (ϵ, δ)-differential privacy via injecting noise into a subset of model layers $\mathcal{L}$. Let τ be the activation threshold required to trigger a backdoor, and μ be the expected benign activation of the global model on the backdoor input. Then the probability that the backdoor remains functional is bounded as:*

$$\Pr[Y \geq \tau] \leq \exp\left(-\frac{(\tau - \mu)^2}{2\sigma^2}\right), \qquad (16)$$

where Y is the noisy model's activation on the backdoor input.

We analyze the likelihood that the perturbed global model still activates on the adversary's backdoor trigger. Let the clean model's response to the backdoor input be μ, which is typically small when no attack is present. Let Y be the output of the model after noise $\xi_j \sim \mathcal{N}\left(0, \sigma^2\right)$ is injected into the relevant layers. Since noise is independent and zero-centered, the response Y is a Gaussian random variable with variance σ^2. A successful backdoor attack occurs if $Y \geqslant \tau$, where τ is a decision threshold (e.g., the logit score that determines prediction confidence). By the standard tail bound for Gaussian distributions: $\Pr[Y \geq \tau] \leq \exp\left(-\frac{(\tau-\mu)^2}{2\sigma^2}\right)$. This probability decays exponentially with the squared distance between the expected benign response μ and the threshold τ, normalized by the noise variance σ^2.

TABLE II
THE CONFIGURATION OF BADNETS ATTACK

Datasets	Trigger Size	Trigger Location	Poisoning Radio	Attack Stage
MNIST	3×3	(25,25)	0.6/0.8/1.0	Early/Mid/Late
FMNIST	3×3	(25,25)	0.6/0.8/1.0	Early/Mid/Late
CIFAR-10	5×5	(27,27)	0.6/0.8/1.0	Early/Mid/Late

V. EXPERIMENTS

A. Experiment Configuration

We evaluated the effectiveness of FedAMM on three datasets: MNIST [29], FMNIST [30], and CIFAR-10 [31]. We considered a FL scenario with 100 clients, randomly selecting a subset of them to participate in training each round, with each client having its own independent local dataset. A certain proportion of the clients were malicious and performed Badnets [13] backdoor attacks during the training process. We conducted experiments on ResNet-18 [32], and CNN [29] models. We used the simple CNN model on the MNIST and FMNIST datasets and the ResNet-18 model on the CIFAR-10 datasets. Training was performed using the SGD optimizer, with a learning rate of 0.1, and 40% of clients were selected in each round.

We selected FedAvg [4] without poisoning defense measures as the baseline for comparisons. To evaluate performance, we selected state-of-the-art FL methods that resisted poisoning attacks for comparison. Specifically, including FoolsGold [33], Krum [34], FEDCPA [22], FLAME [18], FLTrust [17], and FedDMC [35] methods.

B. Metrics

We considered evaluation metrics including *Main Task Accuracy* (MA), and *Backdoor Accuracy* (BA), which assessed the performance of the final global model. To assess effectiveness in filtering malicious model updates during the clustering process, we measured *True Positive Rate* (TPR) and *True Negative Rate* (TNR). Specifically, MA indicated the model's accuracy on the primary task, which was the proportion of correctly predicted benign input categories. A defense mechanism should not negatively impact the MA. BA measured the proportion of incorrect classifications made by the model when trigger set images were used as input. An effective defense aimed to minimize BA. TPR reflected the ability of the defense to detect poisoned models, calculated as the ratio of malicious models correctly classified as malicious (TP) out of all models classified as malicious: TPR $= \frac{\text{TP}}{\text{TP+FP}}$, where FP was False Positives indicating the number of benign models that were wrongly classified as malicious. TNR measured the proportion of benign models correctly classified as benign (TN) out of the total number of models classified as benign: TNR $= \frac{\text{TN}}{\text{TN+FN}}$, where FN (False Negatives) indicated the number of malicious models that were wrongly classified as benign.

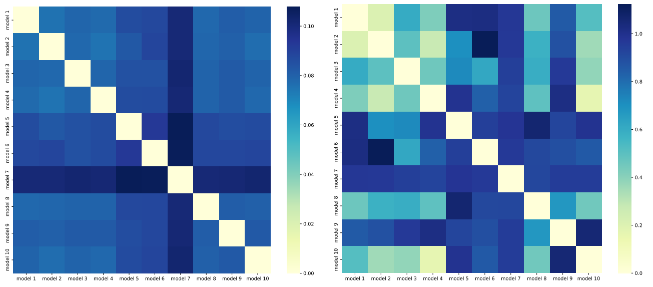
C. Attack Strategy of the Malicious Clients

In our experiments, we simulated Badnets [13] backdoor attacks carried out by malicious clients. Each malicious client injected a predefined trigger (e.g., pixel pattern) into a subset of its local training data and relabelled the corresponding samples to a specific target class. The poisoned samples were then combined with predominantly clean data to form a backdoor-injected training dataset. During local training, the malicious client optimized its local model through multiple rounds using the backdoor-injected dataset, thereby enforcing an association between inputs containing the trigger and the designated target label. In the aggregation stage, local models containing backdoors were integrated into the global model. As a result, the global model performed normally on clean inputs but misclassified inputs embedded with the trigger into the target class specified by the attacker, effectively implanting a hidden backdoor into the model. Table II showed the configuration of various parameters when the malicious client executed the Badnets attack.

D. Experiment Results and Findings

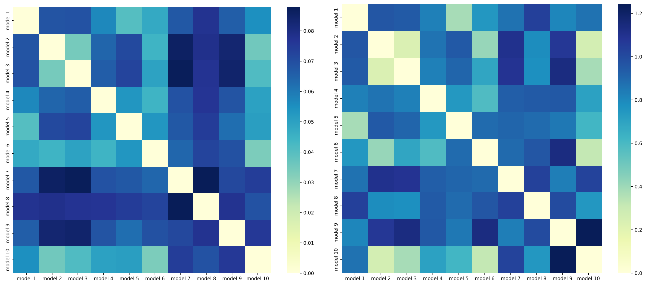
1) Effectiveness of PCA-based Dimensionality Reduction: FedAMM used PCA to reduce the dimensionality of model parameters to address the issue of distance measurement failure in high-dimensional spaces. We evaluated the cosine distance between different model parameters across two consecutive training rounds. Fig. 3 presented results for a CNN model on the MNIST dataset. As shown in Fig. 3a, cosine distances between the original model parameters ranged from 0 to 0.10. Small differences in cosine distance derived from the simplicity of the model and dataset. After applying PCA, as shown in Fig. 3b, the cosine distances between model parameters ranged from 0 to 1.0, with more observable parameter differences. Fig. 4 showed the results for the ResNet18 model on the CIFAR-10 dataset. Similarly, Fig. 4a depicted the original cosine distances ranged from 0 to 0.08. Due to the increased complexity of the dataset and model, the differences in cosine distances were slightly higher than those of in Fig. 3a. However, the small differences still impacted the effectiveness of clustering. In Fig. 4b, the cosine distances for the more complex model ranged from 0 to 1.2 after PCA. The results demonstrated that FedAMM's use of PCA effectively increased the differences in cosine distances between model parameters, successfully addressing the issue of distance measurement failure in high-dimensional models and providing a strong basis for clustering.

2) Effectiveness of Clustering: OPTICS clustering was a key component of FedAMM, which enabled the accurate classification of benign and malicious models. We applied PCA to reduce the dimensionality of model parameters, calculated cosine distances between the reduced parameters, and then clustered the parameters. Fig. 5 showed comparisons among DBSCAN, HDBSCAN, and ours under the setting of 60% malicious clients. Fig. 5a showed the distribution of benign and malicious models in two-dimensional space after dimensionality reduction. We observed that malicious model parameters were more clustered together, while some benign model parameters were shifted away from the center, which were potentially caused by differences in client data distributions. Fig. 5b presented the clustering results of DBSCAN, where many benign model parameters were discarded as outliers, resulting in a lower TNR and negatively affected MA. Fig. 5c depicted clustering results of HDBSCAN with a larger number of generated clusters, making it harder to identify the benign cluster. Additionally, HDBSCAN divided the benign model parameters into multiple clusters, so that it was unsuitable for a single-cluster selection strategy in benign models. Fig. 5d displayed results of OPTICS clustering, which achieved the highest TNR among the three methods, maintaining the model's MA. We observed that OPTICS was able to group some outlier values into the benign cluster, such that it significantly contributed to a higher-level TNR. Results demonstrated that the proposed scheme could effectively cluster benign models into a single benign cluster and provided supports for subsequent benign cluster selection.



(a) Original model parameter (b) Dimensionality reduction model parameters

Fig. 3. Comparison of cosine distances between model parameters of CNN with MNIST datasets.

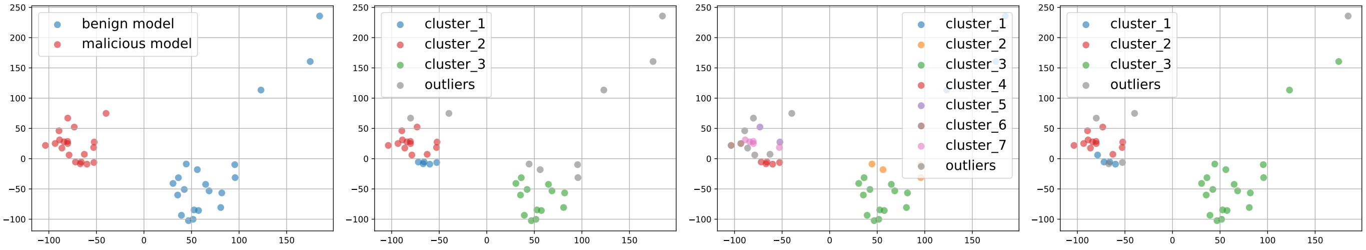


(a) Original model parameter (b) Dimensionality reduction model parameters

Fig. 4. Comparison of cosine distances between model parameters of ResNet-18 with CIFAR-10 datasets.

3) Comparison with Existing Defenses: FedAMM aimed to defend against backdoor attacks from malicious clients while minimizing impact on model performance. To assess FedAMM's defense performance under varying proportions of malicious clients, we compared it with baseline and state-of-the-art defense methods. Table III presented the defense performance of FedAMM and existing methods across different datasets and proportions of malicious clients, focusing on MA and BA.

We evaluated the defense methods on three datasets, including MNIST, FMNIST, and CIFAR-10. MNIST and FMNIST were single-channel image datasets with simple features, while CIFAR-10 was a more complex, three-channel image dataset. On the MNIST dataset, when the proportion of malicious



(a) Original (b) DBSCAN [26] (c) HDBSCAN [20] (d) OPTICS (Ours)

Fig. 5. Comparison of clustering effect of different clustering algorithms.

clients was below 0.4, defense methods such as Krum, FEDCPA, FLAME, and FedDMC exhibited strong performance, effectively reducing the BA. However, when the malicious client ratio exceeded 0.5, all existing methods except FLTrust failed to defend against the attack, with BA rising above 99%. These methods were therefore limited to scenarios where the proportion of malicious models was less than half. FLTrust was an exception, performing well when the malicious client ratio was below 0.8, second only to the FedAMM method. However, its reliance on a clean dataset reduced its practicality. FedAMM effectively controlled BA across all proportions of malicious clients, successfully defending against BadNets. Even when the malicious client ratio reached 0.9, it maintained a low BA of 3.01%, demonstrating strong resistance to backdoor attacks.

A similar pattern was observed observed on the FMNIST dataset where existing defense methods were only effective when malicious clients were in the minority. When the ratio approached or exceeded half, BA increased significantly FedAMM continued to show the best backdoor defense performance. For MA, on both MNIST and FMNIST datasets, all methods except Krum maintain similar levels of accuracy with only acceptable performance degradation. Krum showed a significant drop in MA and lacked practical utility.

On the CIFAR-10 dataset, FedAMM consistently achieved the best BA at nearly all malicious client proportions. When the proportion of malicious clients was 0.2 and 0.3, FedDMC achieved BA rates of 2.33 and 3.58 respectively, showing the best backdoor defense performance. FedAMM recorded BA rates of 4.82 and 4.50 under the same conditions, with a comparable level of defense. When the proportion of malicious clients exceeded 0.4, FedAMM consistently achieved state-of-the-art defense results and demonstrated a clear advantage across all malicious ratios. Fig. 6 illustrated the changes in MA and BA for each scheme during 300 rounds of training under targeted attacks on the CIFAR-10 dataset. Similar to the results shown in Table III, FedAMM demonstrated good performance at all stages of training. Therefore, comparing with other existing methods, FedAMM had superior performance when the proportion of malicious clients exceeded half, while ensuring the impact on main task performance remained adoptable.

4) Evaluation of Defense Performance Against Noise Attack: This section compared the defense performance of FedAMM and existing schemes against noise attacks. In noise attacks, adversaries aimed to reduce model performance, specifically lowering MA in classification tasks. During training, malicious clients initiated Gaussian noise attacks by adding noise to their model updates from the start of training until completion. As shown in Table IV, we compared the variations in accuracy during training at different proportions of malicious clients.

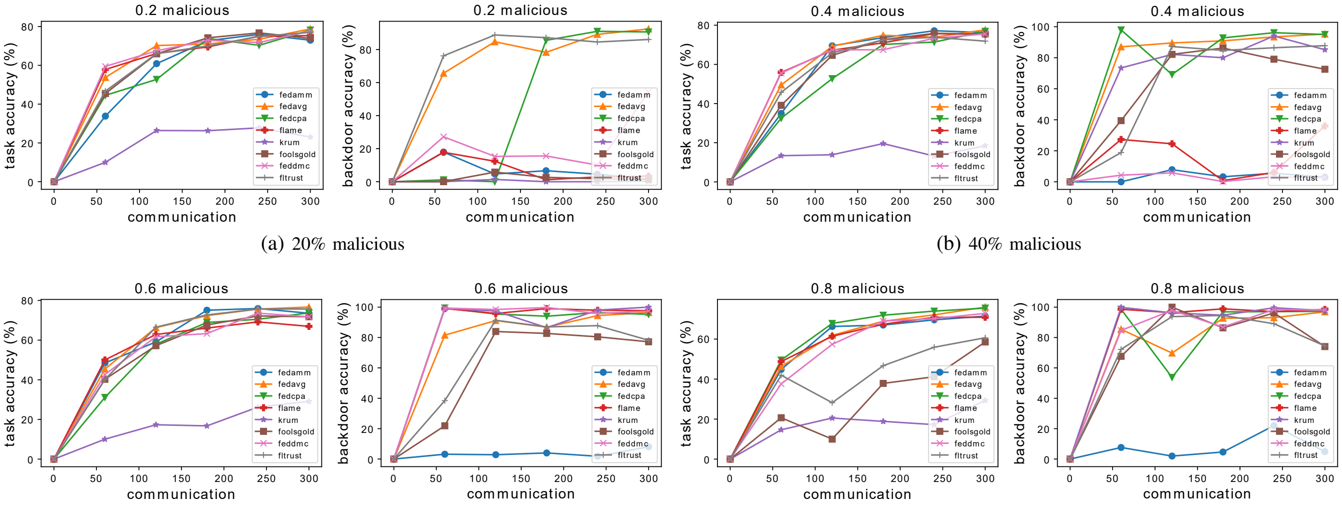
On the MNIST (FMNIST) datasets, when the proportion of malicious clients was 0.2 and 0.4, FEDCPA achieved the highest accuracies of 99.31% (91.85%) and 99.28% (91.86%), outperforming FedAMM by 0.44% (3.20%) and 1.82% (3.75%), respectively. When the proportion of malicious clients reached 0.6 and 0.8, FedAMM achieved the highest accuracy, with values of 98.44% (88.07%) and 98.31% (88.06%), respectively. On the CIFAR-10 dataset, when the proportion of malicious clients was 0.2, FedAMM achieved the highest MA of 81.58%, which was 19.37%, 3.55%, 10.71%, and 2.49% higher than Krum, FEDCPA, FLAME, and FedDMC, respectively. When the proportion of malicious clients was 0.4, FEDCPA achieved the highest accuracy of 77.22%, outperforming FedAMM by 1.81%. When the proportion of malicious clients reached 0.6 and 0.8 (more than half), similar to the results on the MNIST and FMNIST datasets, FedAMM achieved the highest accuracy, with values of 77.41% and 76.19%, respectively. We observed that FEDCPA outperformed FedAMM in defending against noise attacks when the proportion of malicious clients was below 0.5. The reason for analyzing this was because noise attacks lacked strong directionality that caused the critical parameters of malicious models to exhibit low similarity under low malicious ratios. It resulted in the failure of FedAMM's critical parameter analysis module.

In contrast, FEDCPA assumed that critical parameters of benign models exhibited high similarity, which aligned well with the characteristics of noise attacks in low-malicious-ratio scenarios, thereby achieving better defense performance. However, FedAMM still maintained a comparable level of defense capability, owing to the effectiveness of its adaptive clustering and noising module.

Fig. 7 illustrated the variation in MA during 200 rounds of training under noise attacks on the MNIST, FMNIST, and CIFAR-10 datasets for FedAMM, FLAME, and FEDCPA. As shown in Table IV, FedAMM demonstrated a clear advantage in scenarios with a majority of malicious clients, exhibiting strong performance at all stages of training. FLAME, due to its strategy of selecting the largest cluster, exhibited the lowest accuracy in scenarios with more than half malicious clients and

TABLE III
PERFORMANCE OF EXISTING SCHEMES WITH DIFFERENT PROPORTIONS OF MALICIOUS CLIENTS

Method	Datasets	Main Task Accuracy (MA) / Backdoor Accuracy (BA) (%)							
		0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
FedAvg [4]	MNIST	98.68/49.46	98.44/88.96	98.45/99.82	98.47/99.92	98.44/99.93	98.45/99.93	98.51/99.94	98.49/99.94
	FMNIST	86.21/56.41	86.23/72.37	86.13/77.74	86.25/80.78	86.12/82.60	86.16/84.01	86.18/85.08	86.25/85.78
	CIFAR-10	79.65/94.29	78.32/96.09	77.16/97.69	76.68/98.19	77.60/97.49	76.09/97.66	76.57/97.55	76.85/96.89
FoolsGold [33]	MNIST	98.69/46.34	98.49/95.91	98.51/99.87	98.52/99.92	98.47/99.93	98.51/99.93	98.51/99.92	98.50/99.93
	FMNIST	86.27/59.84	86.29/73.76	86.29/78.55	86.23/81.05	86.38/83.07	86.34/84.56	86.25/85.46	86.18/86.04
	CIFAR-10	77.68/2.62	72.52/83.56	65.28/63.65	74.83/80.20	72.39/43.64	69.41/89.13	59.35/91.20	47.54/95.69
Krum [34]	MNIST	97.28/0.24	97.12/0.61	97.03/13.77	97.25/99.88	97.40/99.86	97.16/99.91	97.38/99.89	97.23/99.97
	FMNIST	80.42/0.60	79.19/1.57	78.82/1.58	79.04/45.86	79.71/80.68	79.38/80.02	79.11/82.38	80.13/84.29
	CIFAR-10	35.62/6.58	26.63/43.87	19.74/67.59	40.01/98.16	36.30/98.10	42.72/98.04	38.16/98.12	38.45/98.61
FEDCPA [22]	MNIST	98.30/0.35	98.30/8.84	98.25/93.83	98.10/99.52	98.23/99.73	98.19/99.92	98.25/99.93	98.21/99.92
	FMNIST	85.91/6.25	85.91/66.68	85.89/69.89	85.70/77.33	85.59/79.69	85.77/81.67	85.98/83.38	86.17/84.67
	CIFAR-10	78.42/87.30	75.59/95.98	76.44/96.61	73.43/97.16	75.30/97.11	76.43/97.58	76.47/97.31	77.59/97.00
FLAME [18]	MNIST	98.31/0.19	98.27/0.20	98.29/14.36	98.25/99.23	98.24/99.88	98.19/99.87	98.21/99.87	98.17/99.93
	FMNIST	85.61/1.52	85.95/51.84	85.63/72.18	85.48/77.81	85.58/79.56	85.46/82.44	85.51/84.20	85.60/84.88
	CIFAR-10	74.28/5.35	74.75/5.38	74.80/21.98	73.85/94.11	70.43/97.43	70.50/98.07	71.89/97.57	69.37/97.95
FLTrust [17]	MNIST	98.25/0.18	98.15/0.26	98.18/0.27	98.20/0.41	98.15/1.31	98.18/5.71	98.15/98.64	98.17/99.69
	FMNIST	83.34/0.88	83.22/1.77	83.00/7.35	83.08/17.65	82.96/45.70	83.08/17.65	83.21/73.04	83.25/77.30
	CIFAR-10	73.39/82.13	72.05/68.00	72.90/85.36	75.02/88.06	77.32/88.44	72.61/48.86	64.69/76.25	72.79/89.68
FedDMC [35]	MNIST	98.01/0.17	98.28/0.24	98.36/5.54	98.06/99.68	98.46/99.85	97.35/99.86	98.31/99.85	98.33/99.87
	FMNIST	85.67/0.43	84.46/12.03	86.05/76.51	84.68/79.96	83.58/81.40	85.18/84.12	84.81/85.47	85.52/86.93
	CIFAR-10	77.94/2.33	77.45/3.58	76.27/4.29	75.87/96.94	75.24/97.32	75.78/96.50	73.85/98.58	75.33/96.76
Ours	MNIST	98.27/0.14	98.29/0.14	98.27/0.16	98.38/0.14	98.38/0.16	98.45/0.14	98.37/1.14	98.39/3.01
	FMNIST	85.80/0.19	85.32/0.18	85.65/0.18	85.48/0.39	82.92/0.29	85.30/0.30	85.61/0.28	84.72/5.31
	CIFAR-10	73.61/4.82	73.55/4.50	73.79/3.94	73.73/7.21	71.13/5.67	71.39/4.47	72.21/5.53	71.97/5.11



(a) 20% malicious

(b) 40% malicious

(c) 60% malicious

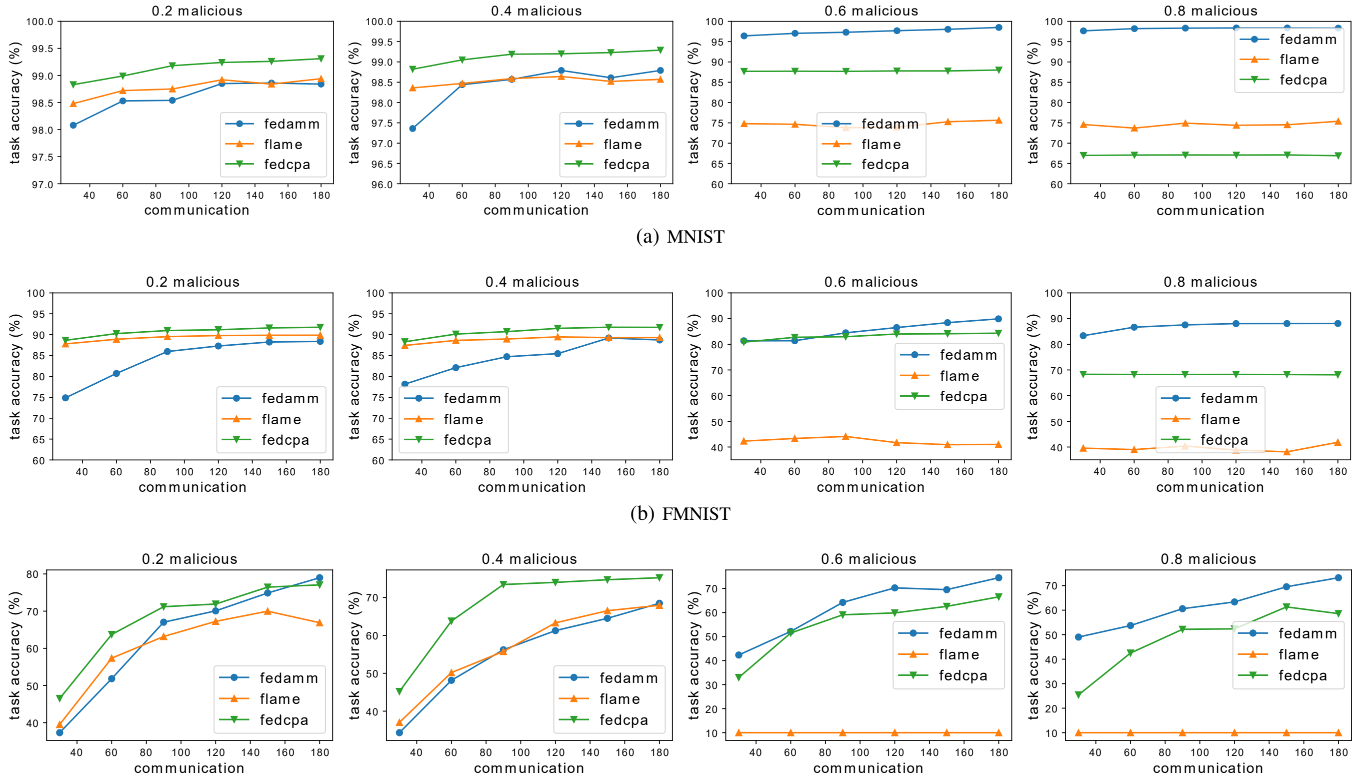
(d) 80% malicious

Fig. 6. Comparison of different schemes against backdoor attacks.

failed to defend effectively against poisoning attacks.

5) Comparison of computation and communication overheads: To evaluate the practicality and efficiency of our proposed method, we compared both the computational and communication overhead with several state-of-the-art baseline algorithms. First, we assessed the computational overhead by measuring the total training time required for each method to to achieve convergence. In addition to computation overhead, we evaluated the communication overhead by measuring the data exchanged between clients and the server during training. Specifically, we tracked the cumulative size of transmitted model parameters or gradients until convergence. This was crucial for assessing the feasibility of deploying the algorithm in bandwidth-constrained environments.

Table V presented a comparison of the computation and communication overhead of FedAMM with existing methods on the MNIST, FMNIST, and CIFAR-10 datasets. In terms of computation overhead, FEDCPA, FedDMC, and FedAMM incurred higher costs due to the intensive analysis of model parameters during the defense process. FedAMM exhibited the highest computational overhead because it incorporates clustering, critical parameter analysis, and adaptive noising modules. However, FedAMM achieved superior performance in backdoor defense, significantly outperforming all other existing methods. Moreover, the computation overhead of FedAMM, FEDCPA, and FedDMC was within the same order



(a) MNIST

(b) FMNIST

(c) CIFAR-10

Fig. 7. Comparison of different schemes against noise attacks.

TABLE IV
PERFORMANCE OF DIFFERENT SCHEMES AGAINST NOISE ATTACKS

Method	Datasets	Main Task Accuracy (%)			
		0.2	0.4	0.6	0.8
Krum [34]	MNIST	73.66	74.75	74.41	73.77
	FMNIST	33.76	31.06	31.21	30.00
	CIFAR-10	62.21	64.97	10.00	10.00
FEDCPA [22]	MNIST	99.31	99.28	88.04	66.87
	FMNIST	91.85	91.86	87.41	68.07
	CIFAR-10	78.03	77.22	76.44	74.26
FLAME [18]	MNIST	98.94	98.59	75.81	74.99
	FMNIST	89.91	89.23	42.03	41.75
	CIFAR-10	70.87	69.71	10.00	10.00
FedDMC [35]	MNIST	98.50	98.35	97.89	92.89
	FMNIST	89.98	87.78	78.40	67.40
	CIFAR-10	79.09	66.42	34.14	24.22
Ours	MNIST	98.87	97.46	98.44	98.31
	FMNIST	88.65	88.11	88.07	88.06
	CIFAR-10	81.58	75.41	77.41	76.19

TABLE V
COMPARISON OF COMPUTATION OVERHEAD AND COMMUNICATION OVERHEAD

Datasets	Method	MA/BA (%)	Comm. cost (MB)	Running time (s)
MNIST	FedAvg	98.39/99.91	89.11	6.49×10^3
	FEDCPA	98.27/99.64	188.44	1.63×10^4
	FLAME	98.08/99.78	173.37	7.76×10^3
	FLTrust	98.16/5.71	98.02	7.42×10^3
	FoolsGold	98.21/99.73	145.62	7.73×10^3
	Krum	97.31/99.06	97.31	8.58×10^3
	FedDMC	98.41/99.79	178.91	1.56×10^4
	Ours	98.38/0.16	135.67	1.67×10^4
FMNIST	FedAvg	86.27/82.51	142.57	7.04×10^3
	FEDCPA	85.61/78.74	263.81	1.72×10^4
	FLAME	85.94/79.77	260.45	8.10×10^3
	FLTrust	82.29/46.02	143.11	7.93×10^3
	FoolsGold	86.31/83.11	216.97	7.43×10^3
	Krum	79.99/80.18	156.45	8.94×10^3
	FedDMC	83.49/80.98	260.79	1.67×10^4
	Ours	85.30/0.41	230.63	1.77×10^4
CIFAR-10	FedAvg	77.31/97.49	3526.62	2.47×10^4
	FEDCPA	75.61/96.13	7472.91	7.02×10^4
	FLAME	72.19/97.71	6974.70	2.66×10^4
	FLTrust	75.97/84.09	4408.25	2.51×10^4
	FoolsGold	72.61/80.97	6129.47	2.48×10^4
	Krum	42.09/98.10	3996.81	6.20×10^4
	FedDMC	75.14/96.92	7013.08	6.94×10^4
	Ours	73.23/4.61	5802.53	8.77×10^4

of magnitude and remained acceptable. In terms of communication overhead, FedAMM outperformed FEDCPA, FLAME, FoolsGold, and FedDMC due to its faster convergence and reduced number of training rounds, which lowered the communication cost. Through these experiments, we demonstrated that the proposed method maintained acceptable overheads in both computation and communication, while achieving robust performance.

6) The Impact of Non-IID: This section examined how non-IID client data affected FedAMM's defense performance. Table VI presented the MA, BA, TPR, and false positive rate of FedAMM across varying levels of non-IID data. We used the parameter q to control the degree of non-IID, representing the extent of label shift. Training samples with label i were assigned to the i-th client group with probability q and to the other groups with probability $\frac{1-q}{n-1}$, where n denoted the num-

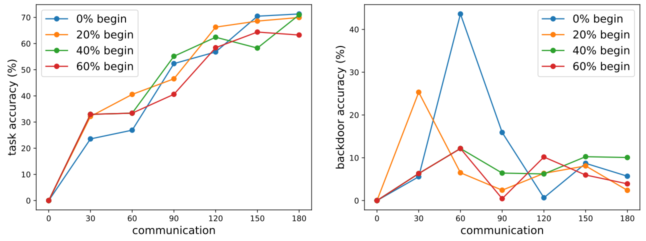
TABLE VI
PERFORMANCE OF FEDAMM WITH DIFFERENT DEGREES OF NON-IID

Datasets	Degree of Non-iid	Metrics (%)			
		MA↑	BA↓	TPR↑	TNR↑
MNIST	0.2	98.18	0.14	85.76	99.93
	0.4	98.21	0.16	85.42	99.21
	0.6	98.26	0.15	85.72	99.77
	0.8	98.27	0.17	86.76	99.87
	0.9	98.16	0.14	87.89	99.27
FMNIST	0.2	85.37	0.31	95.05	92.84
	0.4	85.81	0.39	95.36	97.80
	0.6	85.42	0.37	95.98	99.09
	0.8	85.51	0.29	91.51	94.86
	0.9	85.22	0.31	96.86	94.68
CIFAR-10	0.2	73.81	3.96	80.56	99.93
	0.4	73.73	3.56	80.24	99.84
	0.6	73.91	3.60	82.69	99.82
	0.8	73.34	3.81	83.80	99.93
	0.9	73.08	4.13	83.07	99.63

ber of groups. Once the samples were assigned to groups, they were uniformly and randomly distributed among the clients in each group. $q = 0.1$ meant the local training data of the clients is IID. As q increased, the degree of non-IID increased. As shown in Table VI, we tested the impact of different levels of non-IID data on FedAMM's defense performance using the MNIST, FMNIST, and CIFAR-10 datasets.

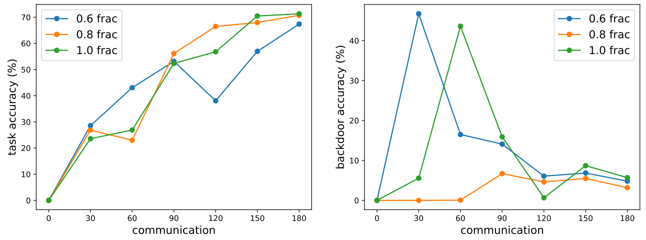
On MNIST dataset, the average MA was 98.22%, BA was 0.15%, TPR averaged 86.31%, and TNR averaged 99.61%, with standard deviations of 0.043, 0.012, 0.91, and 0.307, respectively. Non-IID data had minimal impact on the metrics for MNIST. On the FMNIST dataset, the average MA was 85.47%, BA was 0.33%, TPR averaged 94.952%, and TNR averaged 95.854%, with standard deviations of 0.196, 0.039, 1.829, and 2.268, respectively. While TPR and TNR showed slightly higher fluctuations on FMNIST under different non-IID levels, the impact on defense performance was negligible. On CIFAR-10 dataset, the average MA was 73.57%, BA was 3.81%, TPR averaged 82.072%, and TNR averaged 99.83%, with standard deviations of 0.314, 0.215, 1.415, and 0.11, respectively. For CIFAR-10, fluctuations in all metrics were small, with TNR being the most stable. These results demonstrated that FedAMM was effective in defending against poisoning attacks across various non-IID data scenarios, maintaining high MA, TPR, TNR, and low BA.

7) The Impact of Attack Initiation Conditions: This section examined how different attack initiation conditions affected the defense performance of FedAMM. Attack initiation conditions were a critical factor influencing the success of an attack. To increase the stealthiness of their attacks, adversaries may have chosen to launch poisoning attacks at various stages of model training. In this experiment, we tested four attack strategies: (1) attacking at the beginning of training (model accuracy greater than 0%), (2) attacking in the early stages of training (model accuracy greater than 20%), (3) attacking in the middle stages of training (model accuracy greater than 40%), and (4) attacking in the later stages of training (model accuracy greater than 60%). Fig. 8 showed the defense performance of FedAMM under these four attack strategies. As seen in Fig. 8a, model accuracy stabilized after a certain number of training rounds, with no noticeable differences. Fig. 8b showed that BA fluctuated significantly during the early stages of training. However, as the training progressed, BA dropped considerably. The experimental results indicated that FedAMM was effective against adversaries using different attack initiation strategies, highlighting its practicality.



(a) Task accuracy (b) Backdoor accuracy

Fig. 8. Task accuracy and backdoor accuracy change of global model under CIFAR-10 datasets for different attack initiation conditions.



(a) Task accuracy (b) Backdoor accuracy

Fig. 9. Task accuracy and backdoor accuracy change of global model under CIFAR-10 datasets for different poisoning ratio.

8) The Impact of Poisoning Ratio: This section examined the impact of different poisoning ratios on FedAMM's defense performance. The poisoning ratio in a batch was a key factor determining the strength of a poisoning attack. In this experiment, we used three poisoning ratios 0.6, 0.8, and 1.0 to simulate varying levels of attack intensity. Fig. 9 showed FedAMM's defense performance under these different attack intensities. As shown in Fig. 9a, model accuracy stabilized after a certain number of training rounds and is not affected by the poisoning intensity. In Fig. 9b, BA fluctuated significantly during the first 120 rounds. However, as training continued, BA decreased substantially across all levels of poisoning attack intensity. The experiments in this section demonstrated that FedAMM effectively defended against poisoning attacks of varying strengths, showing strong defensive capabilities.

VI. RELATED WORK

A. Outlier Detection-based Backdoor Defenses

Outlier detection-based backdoor defenses detect potential malicious gradients by comparing distances between gradients or distances between gradients and a statistical value. Common distance metrics include cosine distance, Euclidean distance, and Manhattan distance.

Auror [36], FoolsGold [33], and Krum [34] are classic outlier detection-based backdoor defense methods. Auror [36] employs K-means clustering to detect malicious gradients and selects the largest cluster as the benign cluster. However, Auror assumes that client data is identically distributed and entails considerable computational costs. To address malicious model detection in non-IID data distributions, Fung *et al.* [33] proposed FoolsGold, which identifies malicious models by computing pairwise similarities between updated model parameters. FoolsGold adjusts client learning rates in each training round, assigning higher learning rates to clients providing unique gradient updates and lowering the learning rates of clients uploading duplicate gradient updates. Blanchard *et al.* [34] introduced the Krum, which calculates the Euclidean distance between gradients and selects a single model whose distance is minimized with respect to other models for aggregation. However, Krum is vulnerable to attacks from adaptive adversaries [37]. The mentioned methods require assumptions about client data distributions and cannot serve as universal backdoor defense methods. To provide comprehensive protection, Flame [18] and FreqFed [21] integrate outlier detection and model parameter pruning to counter backdoor attacks. However, Flame and FreqFed exhibit consistently limited defense efficacy in scenarios with a disproportionately large number of malicious clients.

B. Backdoor Defenses using Clean Root Data

Backdoor defense using clean root data typically require servers to collect some clean data from clients as root data and use this root data to train the server model. During the global model aggregation process, the server evaluates the trustworthiness of client models using the server model.

In the Zeno [38] method, the server utilizes a local validation dataset to compute a gradient score for each client, selecting the top-k clients with the highest scores to update the global model. However, the effectiveness of the Zeno method is limited by the number of malicious clients. To address the constraint imposed by the number of malicious clients, Cao *et al.* [39] proposed a defense method that iteratively updates the global model using the mean of gradients from normal user updates and the server gradient, capable of resisting poisoning attacks from any number of malicious clients. To counter adaptive attacks, FLTrust [17] calculates a trust score for users by comparing the cosine similarity between the local trusted model and the user-uploaded model. Moreover, FLTrust defends against large-magnitude gradient attacks by aggregating gradients based on their normalized trust scores. The mentioned methods can withstand attacks from the majority of Byzantine adversaries and exhibit robust performance. However, the collection of root data compromises clients' privacy and poses challenges in ensuring that collected root data remains untainted.

C. Differential Privacy-based Backdoor Defenses

Differential privacy-based backdoor defenses limit the impact of malicious model parameters on the output by introducing random noise into client or global models, thereby undermining the effectiveness of backdoor triggers injected by attacker.

CRFL [40] mitigates the bias in the global model caused by poisoned models during training through pruning and adding noise to the global model. Similar to the CRFL, Flame [18] trims the high-amplitude model parameters after clustering and adds adaptive noise to reduce the influence of the poisoned model on the global model. CRFL and FLAME are server-side defense solutions that have limitations in scenarios where the global model has been poisoned. Sun *et al.* [41] proposed a client-side defense mechanism called FL-WBC, which adds noise during weight updates to reduce the influence of poisoned models. The differential privacy-based backdoor defenses not only enhance the robustness of the global model to outliers and adversarial updates but also simultaneously ensure the protection of model parameter privacy. However, these differential privacy-based methods are primarily designed to mitigate the influence of poisoned or manipulated local models on the global aggregation process, and they lack the capability to directly identify or isolate malicious client updates. Furthermore, determining appropriate pruning levels and noise injection thresholds remains a critical and nontrivial challenge, as suboptimal configurations can significantly degrade the overall performance, stability, and accuracy of the resulting global model.

VII. Conclusions

In this paper, we propose a general FL backdoor defense scheme, called FedAMM, which filters backdoor models across various data distributions and malicious client ratios, enhancing the robustness of the training process. By combining model clustering with critical parameter analysis, our scheme effectively addresses FL challenges with numerous malicious clients, showing strong practicality. To enhance the effectiveness of malicious model filtering, FedAMM refines the benign cluster selection strategy by scoring clusters, achieving high true positive and true negative rates in model filtering. Furthermore, FedAMM's hierarchical Gaussian noise perturbation approach removes residual backdoors while preserving model utility. Experiment results demonstrate that the proposed scheme effectively defends against backdoor attacks on multiple datasets.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No.s U24B20146 and 62372044). J. Yu is the corresponding author (Email: jing.yu@muc.edu.cn).

References

[1] H. Woisetschläger, A. Erben, S. Wang, R. Mayer, and H. Jacobsen, "A survey on efficient federated learning methods for foundation model training," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Jeju, South Korea, 2024, pp. 8317–8325.

[2] S. Wang, J. Yu, K. Gai, and L. Zhu, "Revfed: Representation-based privacy-preserving vertical federated learning with heterogeneous models," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2024, pp. 386–397.

[3] S. Wang, K. Gai, J. Yu, Z. Zhang, and L. Zhu, "Pravfed: Practical heterogeneous vertical federated learning via representation learning," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2693–2705, 2025.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[5] B. Soltani, Y. Zhou, V. Haghighi, and J. C. S. Lui, “A survey of federated evaluation in federated learning,” in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, Macao, SAR, China, 2023, pp. 6769–6777.

[6] S. Wang, K. Gai, J. Yu, L. Zhu, H. Wu, C. Wei, Y. Yan, H. Zhang, and K.-K. R. Choo, “Rafls: Rdp-based adaptive federated learning with shuffle model,” *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 2, pp. 1181–1194, 2025.

[7] K. Gai, D. Wang, J. Yu, M. Wang, L. Zhu, and Q. Wu, “Mfl-owner: Ownership protection for multi-modal federated learning via orthogonal transform watermark,” in *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*. Philadelphia, PA, USA: AAAI Press, 2025, pp. 3049–3058.

[8] K. Gai, Z. Wang, J. Yu, and L. Zhu, “MUFTI: multi-domain distillation-based heterogeneous federated continuous learning,” *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 2721–2733, 2025.

[9] D. Chai, L. Wang, L. Yang, J. Zhang, K. Chen, and Q. Yang, “A survey for federated learning evaluations: Goals and measures,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 10, pp. 5007–5024, 2024.

[10] W. Lu, J. Wang, Y. Chen, X. Qin, R. Xu, D. Dimitriadis, and T. Qin, “Personalized federated learning with adaptive batchnorm for healthcare,” *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 915–925, 2022.

[11] M. B. Singh, H. Singh, and A. Pratap, “Energy-efficient and privacy-preserving blockchain based federated learning for smart healthcare system,” *IEEE Transactions on Services Computing*, vol. 17, no. 5, pp. 2392–2403, 2024.

[12] Y. Zhang, H. Lu, N. Liu, Y. Xu, Q. Li, and L. Cui, “Personalized federated learning for cross-city traffic prediction,” in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, Jeju, South Korea, 2024, pp. 5526–5534.

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

[14] C. Xie, K. Huang, P.-Y. Chen, and B. Li, “Dba: Distributed backdoor attacks against federated learning,” in *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020, p. 1.

[15] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, “Attack of the tails: Yes, you really can backdoor federated learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 070–16 084, 2020.

[16] H. Zhuang, M. Yu, H. Wang, Y. Hua, J. Li, and X. Yuan, “Backdoor federated learning by poisoning backdoor-critical layers,” in *The Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024, p. 1.

[17] X. Cao, M. Fang, J. Liu, and N. Z. Gong, “Fltrust: Byzantine-robust federated learning via trust bootstrapping,” in *ISOC Network and Distributed System Security Symposium*, 2021, p. 1.

[18] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen *et al.*, “Flame: Taming backdoors in federated learning,” in *31st USENIX Security Symposium*, Boston, MA, USA, 2022, pp. 1415–1432.

[19] S. Huang, Y. Li, C. Chen, L. Shi, and Y. Gao, “Multi-metrics adaptively identifies backdoors in federated learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4652–4662.

[20] R. J. Campello, D. Moulavi, and J. Sander, “Density-based clustering based on hierarchical density estimates,” in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2013, pp. 160–172.

[21] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A. Sadeghi, “Freqfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning,” in *31st Annual Network and Distributed System Security Symposium*, San Diego, California, USA, 2024, p. 1.

[22] S. Han, S. Park, F. Wu, S. Kim, B. Zhu, X. Xie, and M. Cha, “Towards attack-tolerant federated learning via critical parameter analysis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4999–5008.

[23] E. Schubert and M. Gertz, “Improving the cluster structure extracted from optics plots.” in *LWDA*, 2018, pp. 318–329.

[24] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A. Sadeghi, T. Schneider, H. Yalame, and S. Zeitouni, “Safelearn: Secure aggregation for private federated learning,” in *IEEE Security and Privacy Workshops*, San Francisco, CA, USA, 2021, pp. 56–62.

[25] Y. Liu, S. Chang, D. Li, S. Shi, and B. Li, “Rope-door: Towards robust and persistent backdoor data poisoning attacks in federated learning,” *IEEE Network*, vol. 39, no. 3, pp. 302–310, 2025.

[26] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[27] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, and Y. Chang, “Robust early-learning: Hindering the memorization of noisy labels,” in *9th International Conference on Learning Representations*, Virtual Event, Austria, 2021, p. 1.

[28] M. Du, R. Jia, and D. Song, “Robust anomaly detection and backdoor attack detection via differential privacy,” *arXiv preprint arXiv:1911.07116*, 2019.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[31] A. Krizhevsky, “Learning multiple layers of features from tiny images,” *Master’s thesis, University of Tront*, vol. pp, no. 99, p. 1, 2009.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[33] C. Fung, C. J. Yoon, and I. Beschastnikh, “The limitations of federated learning in sybil settings,” in *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, San Sebastian, Spain, 2020, pp. 301–316.

[34] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” *Advances in neural information processing systems*, vol. 30, no. 99, pp. 119–129, 2017.

[35] X. Mu, K. Cheng, Y. Shen, X. Li, Z. Chang, T. Zhang, and X. Ma, “Feddmc: Efficient and robust federated learning via detecting malicious clients,” *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5259–5274, 2024.

[36] S. Shen, S. Tople, and P. Saxena, “Auror: Defending against poisoning attacks in collaborative deep learning systems,” in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, Los Angeles, CA, USA, 2016, pp. 508–519.

[37] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to byzantine-robust federated learning,” in *29th USENIX security symposium*, 2020, pp. 1605–1622.

[38] C. Xie, S. Koyejo, and I. Gupta, “Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6893–6901.

[39] X. Cao and L. Lai, “Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers,” *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5850–5864, 2019.

[40] C. Xie, M. Chen, P.-Y. Chen, and B. Li, “Crfl: Certifiably robust federated learning against backdoor attacks,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 372–11 382.

[41] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen, and H. Li, “Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 613–12 624, 2021.