Toward Malicious Clients Detection in Federated Learning

Zhihao Dou*
Duke University
Durham, USA

Jiaqi Wang*
Hainan Normal University
Haikou, China

Wei Sun
Wichita State University
Wichita, USA

Zhuqing Liu
University of North Texas
Denton, USA

Minghong Fang
University of Louisville
Louisville, USA

Abstract

Federated learning (FL) enables multiple clients to collaboratively train a global machine learning model without sharing their raw data. However, the decentralized nature of FL introduces vulnerabilities, particularly to poisoning attacks, where malicious clients manipulate their local models to disrupt the training process. While Byzantine-robust aggregation rules have been developed to mitigate such attacks, they remain inadequate against more advanced threats. In response, recent advancements have focused on FL detection techniques to identify potentially malicious participants. Unfortunately, these methods often misclassify numerous benign clients as threats or rely on unrealistic assumptions about the server's capabilities. In this paper, we propose a novel algorithm, SafeFL, specifically designed to accurately identify malicious clients in FL. The SafeFL approach involves the server collecting a series of global models to generate a synthetic dataset, which is then used to distinguish between malicious and benign models based on their behavior. Extensive testing demonstrates that SafeFL outperforms existing methods, offering superior efficiency and accuracy in detecting malicious clients.

CCS Concepts

• **Security and privacy → Systems security**.

Keywords

Federated learning, Poisoning Attacks, Malicious Clients Detection

ACM Reference Format:
Zhihao Dou, Jiaqi Wang, Wei Sun, Zhuqing Liu, and Minghong Fang. 2025. Toward Malicious Clients Detection in Federated Learning. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '25), August 25–29, 2025, Hanoi, Vietnam.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3708821.3736194

*Equal contribution. Zhihao Dou and Jiaqi Wang conducted this research while they were interns under the supervision of Minghong Fang.

ASIA CCS '25, August 25–29, 2025, Hanoi, Vietnam

ACM ISBN 979-8-4007-1410-8/2025/08
https://doi.org/10.1145/3708821.3736194

1 Introduction

Federated learning (FL) [48] is a distributed machine learning approach that trains models on multiple decentralized clients, each holding its own local data, without sharing that data. This method primarily mitigates privacy concerns associated with centralized training methods. In FL, a central server collects models from each participating client, maintaining data confidentiality while collectively enhancing the model. The process unfolds in three main steps: first, the server distributes the current global model to clients or a selected group. Then, these clients adjust their local models using their specific local training data. Finally, they send their local models back to the server, where these models are combined according to a predefined aggregation rule to refine the global model. FL has been adopted in various real-world applications, including credit risk evaluation [2], speech recognition [56], and next-word prediction [1].

In recent years, much research [10, 18, 36, 44–46, 49, 63, 78] has focused on enhancing the efficiency of FL. However, a significant obstacle to its widespread adoption lies in FL's inherent vulnerability to poisoning attacks carried out by malicious clients [5, 6, 8, 15, 24, 62, 67, 71, 74], due to its decentralized structure. In these attacks, adversaries can undermine the integrity of the global model by tampering with their local training data or altering the models they send to the server, thereby contaminating the aggregated global model. These malicious actions typically manifest in two primary forms. The first, *untargeted attacks* [15, 24], seeks to degrade the global model's performance on a wide range of test cases, aiming for a general reduction in accuracy across various scenarios. The second type, *targeted attacks* [5, 6, 8, 67], involves more strategic manipulation, where the attacker's goal is to influence the global model in such a way that it produces specific, desired outputs for selected test cases, often with malicious intent. These attacks pose significant challenges to the robustness, reliability, and security of FL systems, making it imperative to develop effective defense mechanisms that can detect and neutralize such threats.

To mitigate poisoning attacks, several defensive mechanisms have been proposed in the literature [9, 23, 26, 35, 39, 41, 43, 50–52, 55, 58, 59, 65, 68, 70]. These defenses can generally be categorized into *detection-based* and *prevention-based* approaches. Detection-based defenses focus on identifying malicious clients in the FL system and subsequently removing them. For instance, in FLTrust [14], the server possesses a small validation dataset that resembles the clients' training data. Using this validation data, the server generates a reference model, and a client is classified as benign if its local model aligns positively with this reference model. Similarly, FLDetector [73] predicts a client's model based on historical models,

flagging clients as malicious if their actual and predicted models significantly diverge over multiple rounds. In contrast, prevention-based defenses aim to reduce the impact of malicious clients without removing them from the system. The Median [70] method, for instance, computes the coordinate-wise median of local models to derive the global model. However, current defenses against poisoning attacks to FL face notable limitations. Existing detection-based methods struggle to accurately identify malicious clients in complex scenarios, often misclassifying benign clients as malicious, especially when both targeted and untargeted attacks are present. Furthermore, some methods like FLTrust rely on unrealistic assumptions, such as the server possessing a validation dataset that reflects the overall distribution of client data. On the other hand, prevention-based defenses cannot fully mitigate the effects of malicious clients, as these malicious clients remain within the FL system.

Our work: In this study, we aim to fill this critical gap by presenting a novel detection-based defense mechanism, referred to as SafeFL, specifically designed to identify and mitigate the impact of malicious clients in FL systems. Our proposed SafeFL relies on the server maintaining its own unique dataset, enabling it to evaluate the integrity of local models submitted by participating clients. Given the inherent challenges associated with the server's ability to have complete knowledge of the data distribution across the diverse and heterogeneous devices of clients, we propose an innovative solution. Instead of relying on a dataset that mirrors the clients' data, the server generates a synthetic dataset derived from the trajectory of global models trained over multiple rounds. While this synthetic dataset does not replicate the actual distribution of client data, it is crafted to effectively distinguish between malicious and benign behaviors in models. By leveraging this dynamic dataset generation strategy, SafeFL ensures robust detection and defense against malicious activities, enhancing the overall security and reliability of FL systems.

After creating the synthetic dataset, the server leverages it to identify potential malicious clients by analyzing the distinct behavioral differences between malicious and benign local models when evaluated on this dataset. This approach operates under the assumption that malicious models, crafted with adversarial intent, will demonstrate significantly different performance compared to benign models. To implement this detection mechanism, we introduce two variations of our defense strategy, referred to as SafeFL-ML and SafeFL-CL, each utilizing a unique methodology for identifying malicious clients. The foundation of SafeFL-ML lies in the observation that malicious local models tend to incur a higher loss on the synthetic dataset compared to benign models. Guided by this principle, the server evaluates the loss of each local model submitted by clients and calculates the median loss across all models. A client is classified as benign if the loss of its corresponding model falls below this median value; otherwise, it is flagged as potentially malicious. On the other hand, SafeFL-CL employs a different strategy while still relying on the loss evaluation of local models against the synthetic dataset. Instead of utilizing a median-based threshold, the server applies a clustering algorithm to group models based on their loss values. This approach is grounded in the premise that the loss values of benign models are more likely to cluster closely together, reflecting their consistent and non-adversarial behavior. By categorizing models into clusters, SafeFL-CL identifies outliers, which are indicative of malicious clients, with greater precision.

We conduct an extensive evaluation of our proposed SafeFL using five diverse datasets, including large-scale benchmarks such as CIFAR-10 [38], STL-10 [20], and Tiny-ImageNet [21], as well as the FEMNIST [12] dataset, which is inherently heterogeneous. These datasets span multiple domains to ensure a comprehensive assessment of our approach. Our evaluation also encompasses eleven poisoning attack scenarios and ten state-of-the-art FL defenses. Among these defenses are seven detection-based methods—FLAME [54], FLDetector [73], FLTrust [14], DeepSight [59], BackdoorIndicator [42], FreqFed [30], FedREDefense [69]—as well as three prevention-based strategies, namely Median [70], Trimmed mean [70], and Krum [9]. Beyond these benchmarks, we explore a variety of practical settings in FL that reflect real-world challenges. These include scenarios where clients operate with highly non-independent and identically distributed training data, such as datasets limited to three classes per client. We also consider cases where clients utilize complex deep learning models, such as ResNet-20 [33], for local training. On the server side, we investigate the impact of employing different aggregation rules to combine the local models submitted by clients. This comprehensive evaluation highlights the robustness and adaptability of our method under diverse and challenging conditions.

The contributions of our work can be outlined as follows:

- We propose a novel detection framework, SafeFL, designed to identify malicious clients in FL.
- We evaluate our proposed detection method on five datasets and against eleven distinct poisoning attacks, including the strong adaptive attack, and compare its performance with ten state-of-the-art FL defense baselines.
- Extensive experiments show that our proposed SafeFL not only excels at identifying malicious clients in FL but also outperforms existing detection approaches.

2 Background and Related Work

2.1 Background on federated learning (FL)

A typical federated learning (FL) system consists of a central server and n distributed clients. Each client i has its own distinct database, referred to as the local training dataset D_i, where $i = 1, 2, \ldots, n$. The combined training dataset across all clients is represented as D, where $D = \bigcup_{i=1}^{n} D_i$. In an FL framework, these n clients collaborate under the coordination of the central server to train a shared global machine learning model. The primary goal of FL is to derive the optimal global model $\mathbf{w}^*$, which is obtained by solving the following optimization problem $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} \mathcal{L}(D_i, \mathbf{w})$, where $\mathcal{L}(D_i, \mathbf{w})$ represents the loss corresponding to the local training data of client i, based on the model parameter $\mathbf{w}$; d is the dimension of $\mathbf{w}$. FL tackles the above optimization problem through a decentralized methodology. The training procedure in round t is carried out in three sequential steps:

- **Step I (Global model synchronization):** The server transmits the current global model $\mathbf{w}^t$ to all clients or a portion of clients.
- **Step II (Local models updating):** Upon receiving the global model $\mathbf{w}^t$ from the server, clients refine their local models using

stochastic gradient descent (SGD). Specifically, client i selects a mini-batch of training samples from its dataset D_i, computes a gradient $\mathbf{g}_i^t$ based on $\mathbf{w}^t$ and the sampled data, and then updates its local model as $\mathbf{w}_i^t = \mathbf{w}^t - \mu \cdot \mathbf{g}_i^t$, where μ is the learning rate. Finally, client i sends its updated model $\mathbf{w}_i^t$ to the server.

- **Step III (Local models aggregation):** After receiving the local models from the clients, the server applies a specified aggregation rule, denoted as AR, to combine the models. This is expressed as $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_n^t\}$.

FL iteratively performs the outlined three steps until it meets the convergence criteria. When all participating clients are reliable and act without malicious intent, the server can adopt the straightforward FedAvg [48] algorithm to aggregate the local model updates. This approach updates the global model by averaging the local models received from the clients, calculated as $\mathbf{w}^{t+1} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_i^t$.

2.2 Poisoning Attacks to FL

FL is inherently susceptible to poisoning attacks due to its decentralized structure. These attacks can be classified into two categories based on the attacker's objectives: untargeted attacks [15, 24, 60, 61, 66] and targeted attacks [5, 6, 8, 67]. Untargeted poisoning attacks aim to degrade the overall performance of the global model across arbitrary test inputs. For instance, in the label-flipping attack [62], malicious clients alter the labels in their local training data to mislead the model. Fang et al. [24] proposed a framework for untargeted FL attacks by formulating the problem as an optimization task. Their approach focuses on crafting malicious client updates that maximize the discrepancy between the aggregated global model updates before and after the attack. Conversely, targeted poisoning attacks, such as backdoor attacks, are designed to manipulate the global model to predict an attacker-specified label when provided with test inputs containing a specific, pre-determined trigger.

2.3 Defenses against Poisoning Attacks to FL

Various defenses [9, 25, 27–29, 35, 39, 52, 59, 70] have been proposed to counter poisoning attacks in FL. Some methods focus on identifying and excluding malicious clients from the FL system. For example, FLAME [54] leverages the HDBSCAN [13] clustering algorithm to detect potentially malicious clients, while FLDetector [73] evaluates the consistency of a client's local model to flag suspicious behavior. Other defense mechanisms aim to mitigate the impact of malicious clients without directly identifying them. For instance, Trimmed-mean [70] and Median [70] are coordinate-wise aggregation techniques that process each dimension of the local models independently. In Trimmed-mean, the values for each coordinate across clients' models are sorted, and the k largest and k smallest values are excluded. The average of the remaining $n - 2k$ values is then calculated for each coordinate.

Limitations of existing defenses: Despite their advancements, existing FL defense mechanisms exhibit notable limitations. First, many defenses either result in the misclassification of a significant number of benign clients as malicious or fail to adequately reduce the influence of malicious clients, as their presence persists within the system. Second, some methods are based on unrealistic assumptions, such as the server requiring access to a clean dataset that accurately reflects the distribution of clients' training data.

3 PROBLEM STATEMENT

Threat model: The threat model we employ follows the approach outlined in prior studies [5, 6, 24, 60, 61, 67]. To elaborate, the attacker controls a set of malicious clients, which may either be fake clients injected by the attacker or benign clients compromised by the attacker. These malicious clients have the capability to transmit arbitrary local models to the server. The extent of the attacker's knowledge of the targeted FL system may vary. In cases of partial knowledge, the attacker possesses information solely about the local models and local training data on the malicious clients. In contrast, with full knowledge, the attacker possesses information about the local models on all clients and the aggregation rule employed by the server. This full knowledge attack represents the most severe scenario, and in this paper, we employ it to assess the efficacy of our proposed approach.

Defender's knowledge and goal: Our goal is to develop a reliable FL detection method that identifies malicious clients based solely on their local model updates, without access to training data or prior knowledge of data distributions or attack strategies. The detection mechanism should ensure *robust learning integrity* by preserving benign clients in non-adversarial settings and avoiding their unintended removal. At the same time, it must achieve *effective threat mitigation* by detecting both targeted and non-targeted attacks during training, while maintaining high classification accuracy and minimizing the impact of malicious updates on the global model.

4 Our SafeFL

4.1 Overview

In our proposed SafeFL, the server begins by collecting a trajectory of the global model and uses this information to generate a synthetic dataset. This synthetic dataset serves as a tool to identify potentially malicious clients by analyzing their behavior. Specifically, malicious local models often generate loss patterns that differ noticeably from those of benign models, facilitating their identification.

4.2 Global Model Trajectory Collection

In our proposed method, the server possesses its own distinct dataset. Upon receiving local models from clients, the server distinguishes between malicious and benign models by comparing their performance on this separate dataset. Ideally, we might assume the server having a small, clean training dataset, as posited in [14], where it is assumed that both the server's dataset and the overall training dataset used by clients are drawn from the same distribution. However, in FL, this assumption does not hold in practice since clients' training data remain on their devices, making it challenging, if not impossible, for the server to have perfect knowledge of the distribution of clients' training data.

To address this challenge, our approach involves the server first gathering a trajectory of the global model and subsequently generating a synthetic dataset based on this collected trajectory. It is important to note that the generated synthetic dataset does not have to replicate the distribution of clients' training data. Instead, it merely needs to be a basis on which malicious and benign local models exhibit different performance. Let $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^\epsilon\}$ represent the trajectory gathered by the server, where ϵ is the length

of trajectory, and $\mathbf{w}^t$ denotes the global model at the t-th training round, $t = 1, 2, ..., \epsilon$. Note that we assume that the server gathers the first ϵ global models. The primary challenge here is determining how to calculate each global model in the trajectory. The straightforward solution involves computing $\mathbf{w}^t$ by directly aggregating the n received local models, that is, $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_n^t\}$ for $t = 1, 2, ..., \epsilon$. However, since some clients may act maliciously and send arbitrary local models to the server, the global model may be corrupted if potential malicious local models are not removed before aggregation. The server addresses this by excluding potentially malicious local models during the initial ϵ training rounds before constructing the global model trajectory.

Our primary insight is that, during poisoning attacks on FL, malicious clients typically alter either the directions and/or magnitudes of their local models. By leveraging this understanding, the server categorizes the received clients' local models into several clusters. The local models within the largest clusters are considered benign, underpinning the belief that the majority of clients in FL are benign. Moreover, local models from benign clients tend to cluster together. Let Cluster() represent the clustering technique employed by the server, such as K-means algorithm. Let $\mathcal{H}^t$ denote the largest cluster formed when the server categorizes the n local models received into several clusters at training round t, where $t = 1, 2, ..., \epsilon$. Thus, we can express this as:

$$\mathcal{H}^t = \text{Cluster}(\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_n^t). \tag{1}$$

After that, the server computes $\mathbf{w}^t$ by aggregating the local models within cluster $\mathcal{H}^t$ as $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_i^t, i \in \mathcal{H}^t\}$. This process enables the server to gather the trajectory $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^\epsilon\}$ over the initial ϵ training rounds.

4.3 Synthetic Data Generation

With the collection of the global model trajectory $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^\epsilon\}$, the server can now leverage this information to generate a synthetic dataset, drawing on insights from recent studies in dataset condensation [16, 37, 47, 57, 64, 76, 77]. This trajectory represents the sequence of global models generated over the first ϵ rounds of training. By analyzing this sequence, the server can effectively create synthetic data that mimics the underlying patterns learned by the model over time. To better understand how this process works, let's consider a network denoted by f, which represents the model architecture used to generate the synthetic dataset. The primary goal here is to generate a synthetic dataset, represented as $D_{\text{syn}} = \{\mathbf{X}, \mathbf{Y}\}$. In this dataset, $\mathbf{X}$ represents the input features and $\mathbf{Y}$ corresponds to the labels. The objective is to ensure that when we train the network f on this synthetic data, the results are comparable to those obtained when the network is trained on the full, real dataset D, which contains data from all participating clients in the FL process. Essentially, we want the synthetic data to be a good approximation of the real data in terms of its ability to train the network effectively.

To generate this synthetic data, consider two global models: $\mathbf{w}^\alpha$ and $\mathbf{w}^{\alpha+\Delta}$. Here, $\mathbf{w}^\alpha$ represents a global model from the collected trajectory $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^\epsilon\}$, and $\mathbf{w}^{\alpha+\Delta}$ is another global model in the same trajectory, where Δ is a step parameter that defines the number of rounds between the two models. By training the network f for Δ steps starting from the model $\mathbf{w}^\alpha$ using the synthetic dataset D_{syn}, we aim to obtain a model that closely resembles $\mathbf{w}^{\alpha+\Delta}$. The idea is that the synthetic data should enable the network to transition from $\mathbf{w}^\alpha$ to $\mathbf{w}^{\alpha+\Delta}$, just as it would if trained on real data D. This synthetic data generation can be formulated as an optimization problem, where the goal is to minimize the difference between the model obtained by training on the synthetic data and the target model $\mathbf{w}^{\alpha+\Delta}$. Formally, the problem is expressed as follows:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{Y}} \|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2, \\ &\text{s.t. } \hat{\mathbf{w}} = f(\mathbf{X}, \mathbf{Y}, \mathbf{w}^\alpha, \Delta), \end{aligned} \tag{2}$$

where $f(\mathbf{X}, \mathbf{Y}, \mathbf{w}^\alpha, \Delta)$ represents training the network f on the synthetic dataset $D_{\text{syn}} = \{\mathbf{X}, \mathbf{Y}\}$ for Δ steps, starting from the model $\mathbf{w}^\alpha$. The objective is to find the synthetic data $\mathbf{X}$ and $\mathbf{Y}$ that minimize the squared Euclidean distance $\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$, ensuring the resulting model $\hat{\mathbf{w}}$ closely aligns with the global model $\mathbf{w}^{\alpha+\Delta}$.

We can apply the gradient descent method to iteratively solve Problem (2). The synthetic dataset generation algorithm (SynGen) is outlined in Algorithm 1. Specifically, in each iteration, we begin by randomly and uniformly selecting α from the sequence $\{1, 2, \ldots, \epsilon - \Delta\}$. Then, we retrieve $\mathbf{w}^\alpha$ and $\mathbf{w}^{\alpha+\Delta}$ from the trajectory $\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^\epsilon\}$. Note that the sequence $\{1, 2, \ldots, \epsilon-\Delta\}$ starts at 1 and ends at $\epsilon - \Delta$ to ensure that both $\mathbf{w}^\alpha$ and $\mathbf{w}^{\alpha+\Delta}$ are within the trajectory. Following this, the server obtains $\hat{\mathbf{w}}$ by training the network for Δ steps (Line 5 in Algorithm 1). The server then calculates the gradient of $\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$ with respect to $\mathbf{X}$ and $\mathbf{Y}$, and proceeds to update the features and labels of the synthetic dataset using gradient descent (Lines 6-7). At the conclusion of Algorithm 1, we obtain the synthetic dataset D_{syn}. Note that in Algorithm 1, $\mathbf{X}^\kappa$ and $\mathbf{Y}^\kappa$ represent the features and labels of the synthetic dataset at iteration κ, respectively.

Algorithm 1 SynGen.

Input: Global model trajectory $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^\epsilon\}$; training iterations Ψ; network f; learning rate γ; parameter Δ.
Output: D_{syn}.
1: Initialize $\mathbf{X}^1$ and $\mathbf{Y}^1$.
2: **for** $\kappa = 1, 2, \cdots, \Psi$ **do**
3: Randomly and uniformly select the α from the sequence $\{1, 2, \ldots, \epsilon - \Delta\}$.
4: Retrieve $\mathbf{w}^\alpha$ and $\mathbf{w}^{\alpha+\Delta}$ from the model trajectory $\{\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^\epsilon\}$.
5: Train the network f on the current synthetic dataset for Δ iterations to obtain the updated model $\hat{\mathbf{w}}$.
6: Evaluate the squared Euclidean distance $\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$, then derive the gradients $\nabla_{\mathbf{X}^\kappa}\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$ and $\nabla_{\mathbf{Y}^\kappa}\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$.
7: Update the features and labels by applying gradient descent: $\mathbf{X}^{\kappa+1} = \mathbf{X}^\kappa - \gamma\nabla_{\mathbf{X}^\kappa}\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$, $\mathbf{Y}^{\kappa+1} = \mathbf{Y}^\kappa - \gamma\nabla_{\mathbf{Y}^\kappa}\|\hat{\mathbf{w}} - \mathbf{w}^{\alpha+\Delta}\|_2^2$.
8: **end for**

4.4 Malicious Clients Detection Using D_{syn}

Once we have acquired the synthetic dataset D_{syn}, we can utilize it to detect potential malicious clients. In the following, we detail two variants of our defense strategy, named SafeFL-ML and SafeFL-CL, each employing a distinct approach to detect malicious clients.

Algorithm 2 SafeFL.

Input: The n clients, each with local training datasets D_i for $i = 1, 2, \ldots, n$; the total number of global training rounds T; learning rate μ; aggregation rule AR; clustering algorithm Cluster(); network f; and parameters ϵ, Ψ, γ, and Δ.

Output: Global model $\mathbf{w}^T$.

1: Initialize $\mathbf{w}^1$.
2: $\mathcal{S} \leftarrow \emptyset$.
3: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{w}^1\}$.
4: **for** $t = 1, 2, \cdots, T$ **do**
5: // Step I (Global model synchronization).
6: Server distributes the current global model $\mathbf{w}^t$ to all clients.
7: // Step II (Local models updating).
8: **for** each client $i = 1, 2, \cdots, n$ in parallel **do**
9: Client i updates its local model $\mathbf{w}_i^t$ using $\mathbf{w}^t$ and D_i.
10: Send $\mathbf{w}_i^t$ to the server.
11: **end for**
12: // Step III (Aggregation and global model updating).
13: // Global model trajectory collection.
14: **if** $t < \epsilon$ **then**
15: $\mathcal{H}^t = \text{Cluster}(\mathbf{w}_1^t, \mathbf{w}_2^t, ..., \mathbf{w}_n^t)$.
16: $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_i^t, i \in \mathcal{H}^t\}$.
17: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{w}^{t+1}\}$.
18: **end if**
19: // Synthetic data generation.
20: **if** $t = \epsilon$ **then**
21: $D_{\text{syn}} = \text{SynGen}(\mathcal{S}, \Psi, f, \gamma, \Delta)$.
22: **end if**
23: // Malicious clients detection using D_{syn}.
24: **if** $t \geq \epsilon$ **then**
25: The server computes the loss l_i^t by applying client i's local model $\mathbf{w}_i^t$ on the synthetic dataset D_{syn}, for $i = 1, 2, \ldots, n$.
26: **if** SafeFL-ML is used **then**
27: Calculate r_i^t for each client using Eq. (4).
28: $\mathbf{w}^{t+1} = \sum_{i=1}^n r_i^t \mathbf{w}_i^t$.
29: **else if** SafeFL-CL is used **then**
30: $Q^t = \text{Cluster}(l_1^t, l_2^t, \ldots, l_n^t)$.
31: $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_i^t : i \in Q^t\}$.
32: **end if**
33: **end if**
34: **end for**

SafeFL-ML: In this section, we introduce the SafeFL-MedianLoss (SafeFL-ML), which is the first variant of our approach. In FL, malicious clients often aim to maximize their attack impact by manipulating the directions and/or magnitudes of their local models. The fundamental idea behind SafeFL-ML is that malicious local models typically result in a larger loss when evaluated on the synthetic dataset D_{syn} compared to the loss observed with benign local models. Using this observation, the server calculates the loss for every received local model, and then computes the median of these n losses, where n is the total number of clients. In particular, at the training round t, one has that:

$$l_{\text{Med}}^t = \text{Median}\{l_1^t, l_2^t, ..., l_n^t\}, \tag{3}$$

where l_i^t represents the loss when the server employs client i's local model $\mathbf{w}_i^t$ to compute the loss on the synthetic dataset D_{syn} during training round t.

In SafeFL-ML, client i is identified as benign if its loss l_i^t is smaller than l_{Med}^t, where $i = 1, 2, \ldots, n$. Furthermore, concerning clients identified as benign, higher losses suggest poorer alignment with the training data distribution, while lower losses reflect better fitting performance. Clients with lower losses are considered more reliable. Thus, at training round t, we can derive the weight of client i using the following formula:

$$r_i^t = \begin{cases} \frac{(l_i^t)^{-1}}{\sum_{j=1}^n (l_j^t)^{-1}}, & \text{if } l_i^t \leq l_{\text{Med}}^t, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

Then, the server can aggregate the local models using the weighted average approach as $\mathbf{w}^{t+1} = \sum_{i=1}^n r_i^t \mathbf{w}_i^t$.

SafeFL-CL: Our median loss selection method, SafeFL-ML, identifies half of the clients as suspicious in each round, regardless of whether their losses closely match or significantly deviate from the median. While this approach ensures consistency in detecting a fixed proportion of suspicious clients, it risks misclassifying benign clients with losses slightly above the median as malicious. Such misclassification can lead to the exclusion of valuable local training data, as a substantial number of benign local models are left out during aggregation, potentially degrading model performance. To address these shortcomings, we propose integrating a clustering-based approach for identifying potential malicious clients based on their computed loss values.

In our SafeFL-ClusterLoss (SafeFL-CL) method, the server also evaluates the loss of each received local model on the synthetic dataset D_{syn}. However, instead of relying on the median loss for classification, the server employs a clustering algorithm to group the loss values. This approach is motivated by the observation that losses derived from benign local models are more likely to form a coherent cluster. Let Q^t denote the largest cluster obtained when the server partitions the n losses into multiple clusters during training round t. Formally, one has that:

$$Q^t = \text{Cluster}(l_1^t, l_2^t, \ldots, l_n^t). \tag{5}$$

Using this cluster, the server computes the global model $\mathbf{w}^{t+1}$ as $\mathbf{w}^{t+1} = \text{AR}\{\mathbf{w}_i^t : i \in Q^t\}$, where AR denotes the aggregation rule applied to the local models within the largest cluster. This clustering-based method ensures that aggregation focuses on benign clients, thereby improving the robustness and effectiveness of the model.

Algorithm 2 provides an overview of the complete process for our proposed method, SafeFL. Lines 5-11 outline the first two steps of standard FL. In Lines 12-33, SafeFL identifies potential malicious clients and updates the global model using contributions from the remaining clients. Our SafeFL consists of three main components: "global model trajectory collection", "synthetic data generation", and "malicious clients detection using D_{syn}". Specifically, during the first ϵ training rounds, the server collects global model trajectories, forming a set denoted as $\mathcal{S}$. At training round ϵ, the server utilizes these collected trajectories to generate a synthetic dataset D_{syn}. It is important to note that D_{syn} is constructed only once at round ϵ, as detailed in Lines 19-22 of Algorithm 2. Once the synthetic dataset

D_{syn} is generated, the server uses it to detect malicious clients, as described in Lines 23-33.

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets. In our experiments, we incorporate the following five datasets: CIFAR-10 [38], MNIST [40], FEMNIST [12], STL-10 [20], and Tiny-ImageNet [21]. The details of these datasets are shown in Appendix A.1.

5.1.2 Poisoning attacks to FL. We consider four single-method poisoning attacks (Trim attack [24], Scaling attack [5], Distributed Backdoor Attack (DBA) [67], and Adaptive attack [60]), along with two hybrid poisoning strategies (Trim+DBA, and Scaling+DBA attacks) to evaluate the effectiveness of our proposed detection method. In the case of single-method attacks, every malicious client uses the same strategy to craft their local models. For instance, in a Trim attack scenario, all involved malicious clients adopt the Trim attack technique to formulate the models they submit to the server. Conversely, in hybrid attacks, different malicious clients might utilize varying attack strategies. See Appendix A.2 for a detailed description of these attacks. Note that we also consider five more advanced attacks is Section 6.

5.1.3 Defenses against Poisoning Attacks to FL. In this paper, we compare our SafeFL against seven detection-based approaches (FLAME [54], FLDetector [73], FLTrust [14], DeepSight [59], BackdoorIndicator [42], FreqFed [30], FedREDefense [69]) and three prevention-based methods (such as Median [70], Trimmed mean (TrMean) [70], and Krum [9]). Refer to Appendix A.3 for comprehensive details of these defenses.

5.1.4 Non-IID setting. FL is characterized by the non-independent and identically distributed (Non-IID) nature of training data across clients. To simulate this, we follow the approach from [24]. In a dataset with M classes, clients are divided into M groups. Each training sample with label g is assigned to group g with probability q and to other groups with a probability of $\frac{1-q}{M-1}$. The parameter q determines the degree of Non-IID distribution; when $q = \frac{1}{M}$, the data is IID, otherwise, it is Non-IID. For CIFAR-10, MNIST, STL-10, and Tiny-ImageNet, we set $q = 0.5$, while FEMNIST remains unchanged due to its inherently Non-IID distribution.

5.1.5 Evaluation metrics. We evaluate using five metrics: three for detection—detection accuracy (DACC), false positive rate (FPR), and false negative rate (FNR)—and two for the final global model—testing accuracy (TACC) and attack success rate (ASR). For detection-based methods, we assess both detection performance and final model accuracy, while for prevention-based methods, we focus solely on final model accuracy. Detection methods, including baselines and our SafeFL, identify malicious clients during each training round, with results averaged over all rounds. It is important to note that we detect malicious clients in every round. When clients are identified as malicious, the server ignores their local models for that particular round, rather than permanently removing them from the system. This approach is taken because malicious clients may choose to attack in certain rounds and refrain from attacking in others. Permanently removing clients upon detection could result in the exclusion of some benign clients.

a) Detection accuracy (DACC): DACC measures the percentage of clients correctly classified, ensuring benign clients are identified as benign and malicious clients are recognized as malicious.

b) False positive rate (FPR): FPR represents the ratio of benign clients incorrectly predicted as malicious.

c) False negative rate (FNR): FNR denotes the proportion of malicious clients erroneously classified as benign.

d) Testing accuracy (TACC): TACC represents the proportion of test samples accurately predicted by the final global model.

e) Attack success rate (ASR): ASR represents the proportion of targeted test samples that are classified into the specific label designated by the attacker.

Higher DACC indicates stronger detection performance, while lower FPR and FNR indicate better detection capabilities. For all defenses, higher TACC and lower ASR reflect a more robust model.

5.1.6 Parameter settings. By default, our experiments involve 100 clients for CIFAR-10, MNIST, and STL-10 datasets, 300 clients for FEMNIST, and 400 clients for Tiny-ImageNet. In the default setup, 30% of clients are considered malicious. For the MNIST and FEMNIST datasets, we have employed a four-layer Convolutional neural network (CNN), as detailed in Table 8 in Appendix, as the model architecture. In the case of CIFAR-10, STL-10 and Tiny-ImageNet datasets, we have adopted the widely recognized ResNet-20 architecture [33] as the architecture. The batch size is set as 64 for CIFAR-10, MNIST, FEMNIST, and Tiny-ImageNet datasets, and 32 for the STL-10 dataset. The total training rounds are set at 1500 for CIFAR-10, 1000 for STL-10, MNIST, and FEMNIST datasets, and 2000 for Tiny-ImageNet. In CIFAR-10 and Tiny-ImageNet, the initial learning rate is 0.15 for the first 1000 training rounds, subsequently reduced by a factor of 0.5 every 250 rounds thereafter. For STL-10, MNIST, and FEMNIST, the learning rate is initially set to 0.10 for the first 500 rounds and then reduced by a factor of 0.5 every 250 rounds after that. By default, we assume that all clients participate in each training round (i.e., a selection rate of 100%). We consider the worst-case setting where the attacker performs attacks in every training round.

For our SafeFL, we use K-means clustering algorithm [32] when generating the synthetic data, and we use the Mean-shift clustering algorithm [19] to group the losses in SafeFL-CL. The global model trajectory length (ϵ) for the synthetic dataset is set to 25 for MNIST, CIFAR-10, and FEMNIST, and 30 for STL-10 and Tiny-ImageNet. The parameter γ is consistently set to 0.1 across all datasets during the synthetic dataset generation. The value of Ψ is configured as 5000 for CIFAR-10, STL-10, and MNIST, 8500 for FEMNIST, and 10,000 for Tiny-ImageNet. Following [57], the server uses the same network f to generate the synthetic dataset as the clients use for local training. Furthermore, the Δ is fixed at 15 for all datasets. For each dataset, the synthetic dataset size is set to 100. In our SafeFL, once the server identifies the malicious clients, it aggregates the remaining local models using the FedAvg method, meaning the aggregation rule AR in SafeFL is configured as FedAvg.

Table 1: Detection performance of various detection-based methods is assessed using DACC (↑), FPR (↓), and FNR (↓) metrics. Here, ↑ denotes better detection performance with higher values, and ↓ denotes better performance with lower values.

Attack	Defense	CIFAR-10			MNIST			FEMNIST			STL-10			Tiny-ImageNet		
		DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR
No attack	FLAME	0.57	NA	NA	0.57	NA	NA	0.61	NA	NA	0.65	NA	NA	0.56	NA	NA
	FLDetector	0.65	NA	NA	0.52	NA	NA	0.63	NA	NA	0.59	NA	NA	0.68	NA	NA
	FLTrust	0.82	NA	NA	0.77	NA	NA	0.89	NA	NA	0.75	NA	NA	0.83	NA	NA
	DeepSight	1.00	NA	NA	0.97	NA	NA	0.93	NA	NA	0.92	NA	NA	0.98	NA	NA
	BackdoorIndicator	0.95	NA	NA	0.93	NA	NA	0.97	NA	NA	0.92	NA	NA	0.86	NA	NA
	FreqFed	0.81	NA	NA	0.84	NA	NA	0.88	NA	NA	0.82	NA	NA	0.89	NA	NA
	FedREDefense	0.78	NA	NA	0.93	NA	NA	0.91	NA	NA	0.76	NA	NA	0.74	NA	NA
	SafeFL-ML	0.94	NA	NA	0.96	NA	NA	0.92	NA	NA	0.99	NA	NA	0.95	NA	NA
	SafeFL-CL	1.00	NA	NA	1.00	NA	NA	1.00	NA	NA	0.99	NA	NA	0.98	NA	NA
Trim attack	FLAME	0.78	0.09	0.28	0.79	0.09	0.26	0.77	0.16	0.26	0.78	0.09	0.28	0.77	0.20	0.24
	FLDetector	0.96	0.02	0.05	0.99	0.00	0.01	0.93	0.03	0.09	0.79	0.09	0.26	0.83	0.07	0.21
	FLTrust	0.85	0.16	0.15	0.85	0.06	0.19	0.81	0.03	0.26	0.84	0.20	0.14	0.87	0.29	0.06
	DeepSight	0.88	0.12	0.12	0.87	0.04	0.17	0.80	0.12	0.23	0.88	0.05	0.15	0.81	0.19	0.19
	BackdoorIndicator	0.73	0.35	0.24	0.71	0.25	0.31	0.74	0.36	0.22	0.73	0.41	0.21	0.67	0.51	0.25
	FreqFed	0.89	0.04	0.14	0.84	0.06	0.20	0.88	0.05	0.15	0.89	0.04	0.14	0.82	0.07	0.23
	FedREDefense	0.85	0.27	0.10	0.92	0.07	0.08	0.94	0.02	0.08	0.98	0.00	0.03	0.99	0.00	0.01
	SafeFL-ML	0.90	0.03	0.13	0.89	0.12	0.11	0.94	0.02	0.08	0.92	0.03	0.10	0.94	0.04	0.07
	SafeFL-CL	1.00	0.00	0.00	1.00	0.00	0.00	0.96	0.01	0.05	0.97	0.00	0.04	0.98	0.01	0.02
Scaling attack	FLAME	0.84	0.07	0.20	0.87	0.05	0.16	0.79	0.03	0.29	0.80	0.00	0.29	0.86	0.11	0.15
	FLDetector	1.00	0.00	0.00	0.94	0.10	0.04	0.90	0.15	0.08	0.79	0.20	0.21	0.84	0.21	0.14
	FLTrust	0.75	0.41	0.18	0.82	0.38	0.09	0.82	0.60	0.00	0.89	0.25	0.05	0.78	0.13	0.26
	DeepSight	0.88	0.12	0.12	0.87	0.04	0.17	0.80	0.12	0.23	0.88	0.05	0.15	0.81	0.19	0.19
	BackdoorIndicator	0.94	0.00	0.09	0.95	0.03	0.06	1.00	0.00	0.00	0.75	0.16	0.29	0.81	0.15	0.21
	FreqFed	0.84	0.17	0.16	0.63	0.50	0.31	0.70	0.24	0.33	0.80	0.11	0.24	0.69	0.62	0.18
	FedREDefense	0.87	0.12	0.13	0.87	0.07	0.16	0.94	0.00	0.09	0.79	0.14	0.24	0.74	0.33	0.23
	SafeFL-ML	0.91	0.03	0.12	0.97	0.00	0.04	1.00	0.00	0.00	0.97	0.00	0.04	0.93	0.12	0.05
	SafeFL-CL	1.00	0.00	0.00	0.94	0.07	0.06	0.98	0.00	0.03	1.00	0.00	0.00	0.94	0.10	0.04
DBA attack	FLAME	0.82	0.07	0.23	0.87	0.03	0.17	0.84	0.07	0.20	0.90	0.15	0.08	0.87	0.01	0.18
	FLDetector	0.89	0.15	0.09	0.90	0.10	0.10	0.91	0.04	0.11	0.79	0.12	0.25	0.87	0.11	0.14
	FLTrust	0.80	0.21	0.20	0.79	0.19	0.22	0.78	0.22	0.22	0.81	0.30	0.14	0.79	0.25	0.19
	DeepSight	0.88	0.00	0.17	0.90	0.03	0.13	0.94	0.03	0.07	0.86	0.15	0.14	0.87	0.12	0.13
	BackdoorIndicator	1.00	0.00	0.00	0.97	0.04	0.03	0.96	0.04	0.04	0.89	0.07	0.13	0.88	0.16	0.10
	FreqFed	0.89	0.04	0.14	0.78	0.17	0.24	0.91	0.00	0.13	1.00	0.00	0.00	0.85	0.10	0.17
	FedREDefense	0.75	0.23	0.26	0.95	0.07	0.04	0.94	0.11	0.04	0.76	0.31	0.21	0.84	0.15	0.16
	SafeFL-ML	0.94	0.11	0.04	0.99	0.00	0.01	0.96	0.03	0.04	1.00	0.00	0.00	0.98	0.00	0.03
	SafeFL-CL	1.00	0.00	0.00	0.98	0.03	0.02	0.99	0.00	0.01	1.00	0.00	0.00	1.00	0.00	0.00
Trim+DBA attack	FLAME	0.81	0.04	0.25	0.84	0.10	0.19	0.79	0.07	0.27	0.90	0.10	0.10	0.86	0.04	0.18
	FLDetector	0.87	0.33	0.04	0.91	0.13	0.07	0.89	0.15	0.09	0.70	1.00	0.00	0.84	0.05	0.21
	FLTrust	0.88	0.15	0.11	0.80	0.13	0.23	0.76	0.39	0.18	0.71	0.22	0.32	0.89	0.25	0.05
	DeepSight	0.75	0.45	0.16	0.80	0.15	0.22	0.74	0.29	0.25	0.82	0.08	0.22	0.86	0.23	0.10
	BackdoorIndicator	0.78	0.39	0.15	0.81	0.04	0.25	0.77	0.19	0.25	0.80	0.20	0.20	0.85	0.15	0.15
	FreqFed	0.80	0.15	0.22	0.73	0.14	0.33	0.81	0.11	0.22	0.79	0.29	0.18	0.77	0.18	0.25
	FedREDefense	0.76	0.27	0.23	0.85	0.13	0.16	0.96	0.00	0.06	0.77	0.28	0.21	0.84	0.12	0.18
	SafeFL-ML	0.91	0.07	0.10	0.95	0.00	0.07	1.00	0.00	0.00	0.94	0.07	0.06	1.00	0.00	0.00
	SafeFL-CL	0.96	0.00	0.06	1.00	0.00	0.00	0.98	0.00	0.03	1.00	0.00	0.00	0.99	0.00	0.01
Scaling+DBA attack	FLAME	0.89	0.00	0.16	0.84	0.07	0.20	0.85	0.04	0.20	0.86	0.04	0.18	0.89	0.10	0.11
	FLDetector	0.69	0.34	0.30	0.70	0.19	0.35	0.88	0.26	0.06	0.75	0.51	0.14	0.83	0.04	0.23
	FLTrust	0.76	0.29	0.22	0.81	0.35	0.12	0.75	0.22	0.26	0.92	0.09	0.08	0.79	0.15	0.24
	DeepSight	0.86	0.07	0.17	0.89	0.10	0.11	0.90	0.16	0.07	0.90	0.00	0.14	0.91	0.15	0.06
	BackdoorIndicator	0.94	0.07	0.06	0.99	0.00	0.01	0.96	0.03	0.04	0.85	0.12	0.16	0.79	0.17	0.23
	FreqFed	0.84	0.13	0.17	0.87	0.22	0.09	0.79	0.05	0.28	0.86	0.15	0.14	0.91	0.18	0.05
	FedREDefense	0.85	0.14	0.15	0.97	0.00	0.04	0.95	0.00	0.07	0.94	0.06	0.06	0.79	0.25	0.19
	SafeFL-ML	0.95	0.04	0.05	0.97	0.00	0.04	0.99	0.00	0.01	0.92	0.00	0.11	0.87	0.15	0.12
	SafeFL-CL	0.98	0.00	0.03	1.00	0.00	0.00	1.00	0.00	0.00	0.96	0.03	0.04	0.99	0.00	0.01
Adaptive attack	FLAME	0.77	0.34	0.18	0.75	0.18	0.28	0.70	0.23	0.33	0.80	0.19	0.20	0.77	0.28	0.21
	FLDetector	0.85	0.20	0.13	0.78	0.26	0.20	0.83	0.16	0.17	0.82	0.16	0.19	0.75	0.33	0.22
	FLTrust	0.70	0.37	0.27	0.70	0.40	0.26	0.75	0.47	0.16	0.79	0.25	0.19	0.71	0.27	0.30
	DeepSight	0.77	0.29	0.20	0.75	0.27	0.24	0.72	0.28	0.28	0.75	0.29	0.23	0.81	0.19	0.19
	BackdoorIndicator	0.71	0.45	0.22	0.89	0.14	0.10	0.80	0.23	0.19	0.67	0.57	0.23	0.75	0.29	0.23
	FreqFed	0.82	0.27	0.14	0.87	0.25	0.08	0.73	0.10	0.34	0.80	0.25	0.18	0.77	0.30	0.20
	FedREDefense	0.85	0.10	0.17	0.95	0.00	0.07	0.79	0.30	0.17	0.75	0.20	0.27	0.70	0.23	0.33
	SafeFL-ML	0.89	0.20	0.07	0.94	0.07	0.06	0.98	0.03	0.02	0.92	0.10	0.07	0.93	0.13	0.04
	SafeFL-CL	0.95	0.03	0.06	0.97	0.00	0.04	0.94	0.00	0.09	0.96	0.03	0.04	0.95	0.00	0.07

5.2 Experimental results

SafeFL is effective: In Table 1, we present the detection performance of our SafeFL and other detection-based approaches. "No attack" means all clients are benign (there are no malicious clients in the system). "NA" means not applicable. We observe that our proposed SafeFL method demonstrates remarkable detection efficacy. First, when all clients are benign, our proposed SafeFL ensures maximum preservation of benign clients and prevents their unintended exclusion. This demonstrates that SafeFL achieves the objective of "Robust learning integrity". For instance, on the CIFAR-10 dataset,

Table 2: Performance of final global models obtained through various detection-based methods, where TACC (↑) and ASR (↓) metrics are considered. ↑ denotes better performance with higher values, and ↓ denotes better performance with lower values.

Attack	Defense	CIFAR-10		MNIST		FEMNIST		STL-10		Tiny-ImageNet	
		TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR
No attack	FLAME	0.72	NA	0.91	NA	0.60	NA	0.47	NA	0.46	NA
	FLDetector	0.77	NA	0.96	NA	0.63	NA	0.45	NA	0.41	NA
	FLTrust	0.81	NA	0.97	NA	0.67	NA	0.50	NA	0.47	NA
	DeepSight	0.75	NA	0.98	NA	0.67	NA	0.50	NA	0.51	NA
	BackdoorIndicator	0.81	NA	0.98	NA	0.66	NA	0.49	NA	0.50	NA
	FreqFed	0.79	NA	0.98	NA	0.64	NA	0.51	NA	0.47	NA
	FedREDefense	0.81	NA	0.97	NA	0.65	NA	0.50	NA	0.47	NA
	SafeFL-ML	0.82	NA	0.98	NA	0.67	NA	0.53	NA	0.51	NA
	SafeFL-CL	0.84	NA	0.98	NA	0.68	NA	0.54	NA	0.54	NA
Trim attack	FLAME	0.77	NA	0.95	NA	0.59	NA	0.46	NA	0.39	NA
	FLDetector	0.79	NA	0.98	NA	0.65	NA	0.46	NA	0.49	NA
	FLTrust	0.74	NA	0.96	NA	0.66	NA	0.49	NA	0.46	NA
	DeepSight	0.74	NA	0.91	NA	0.62	NA	0.44	NA	0.37	NA
	BackdoorIndicator	0.62	NA	0.80	NA	0.54	NA	0.37	NA	0.32	NA
	FreqFed	0.75	NA	0.90	NA	0.62	NA	0.46	NA	0.46	NA
	FedREDefense	0.69	NA	0.94	NA	0.65	NA	0.38	NA	0.42	NA
	SafeFL-ML	0.80	NA	0.97	NA	0.67	NA	0.50	NA	0.50	NA
	SafeFL-CL	0.81	NA	0.97	NA	0.67	NA	0.52	NA	0.51	NA
Scaling attack	FLAME	0.79	0.03	0.97	0.02	0.63	0.14	0.46	0.05	0.48	0.29
	FLDetector	0.79	0.02	0.98	0.07	0.65	0.07	0.47	0.03	0.50	0.01
	FLTrust	0.62	0.45	0.94	0.03	0.65	0.04	0.42	0.45	0.48	0.65
	DeepSight	0.76	0.17	0.95	0.09	0.65	0.04	0.46	0.19	0.47	0.15
	BackdoorIndicator	0.80	0.03	0.98	0.02	0.66	0.06	0.49	0.26	0.49	0.38
	FreqFed	0.69	0.20	0.93	0.08	0.66	0.04	0.47	0.06	0.48	0.10
	FedREDefense	0.77	0.14	0.95	0.12	0.67	0.02	0.46	0.25	0.48	0.67
	SafeFL-ML	0.81	0.03	0.96	0.04	0.66	0.07	0.50	0.04	0.49	0.05
	SafeFL-CL	0.79	0.03	0.98	0.02	0.67	0.06	0.52	0.03	0.52	0.04
DBA attack	FLAME	0.78	0.06	0.95	0.04	0.64	0.07	0.49	0.17	0.50	0.11
	FLDetector	0.77	0.16	0.93	0.10	0.67	0.06	0.48	0.28	0.47	0.45
	FLTrust	0.78	0.35	0.91	0.15	0.60	0.33	0.49	0.70	0.48	0.27
	DeepSight	0.80	0.15	0.96	0.07	0.65	0.02	0.46	0.21	0.49	0.09
	BackdoorIndicator	0.80	0.03	0.97	0.11	0.65	0.06	0.48	0.15	0.47	0.33
	FreqFed	0.78	0.05	0.97	0.15	0.68	0.03	0.50	0.04	0.47	0.63
	FedREDefense	0.70	0.49	0.94	0.08	0.62	0.15	0.48	0.29	0.49	0.17
	SafeFL-ML	0.78	0.20	0.98	0.02	0.66	0.04	0.51	0.05	0.50	0.04
	SafeFL-CL	0.82	0.05	0.97	0.03	0.67	0.05	0.52	0.04	0.52	0.02
Trim+DBA attack	FLAME	0.79	0.03	0.92	0.05	0.62	0.04	0.46	0.05	0.48	0.05
	FLDetector	0.65	0.19	0.89	0.07	0.49	0.19	0.50	0.06	0.52	0.13
	FLTrust	0.77	0.20	0.85	0.21	0.49	0.13	0.29	0.39	0.48	0.29
	DeepSight	0.49	0.22	0.85	0.19	0.54	0.07	0.48	0.04	0.40	0.15
	BackdoorIndicator	0.60	0.07	0.83	0.09	0.50	0.04	0.34	0.21	0.45	0.07
	FreqFed	0.76	0.14	0.91	0.06	0.63	0.07	0.39	0.11	0.47	0.57
	FedREDefense	0.59	0.27	0.93	0.08	0.67	0.03	0.49	0.12	0.46	0.20
	SafeFL-ML	0.79	0.02	0.98	0.07	0.67	0.04	0.47	0.11	0.50	0.05
	SafeFL-CL	0.83	0.03	0.98	0.02	0.66	0.01	0.52	0.05	0.53	0.03
Scaling+DBA attack	FLAME	0.80	0.03	0.84	0.04	0.64	0.20	0.49	0.18	0.49	0.03
	FLDetector	0.54	0.19	0.70	0.13	0.63	0.06	0.27	0.61	0.47	0.13
	FLTrust	0.49	0.20	0.81	0.72	0.49	0.68	0.47	0.18	0.48	0.24
	DeepSight	0.80	0.22	0.89	0.04	0.62	0.31	0.50	0.03	0.45	0.22
	BackdoorIndicator	0.80	0.07	0.99	0.13	0.62	0.04	0.48	0.16	0.46	0.20
	FreqFed	0.78	0.14	0.87	0.39	0.75	0.28	0.42	0.20	0.49	0.28
	FedREDefense	0.72	0.27	0.97	0.12	0.67	0.04	0.49	0.12	0.49	0.27
	SafeFL-ML	0.80	0.02	0.97	0.17	0.67	0.04	0.51	0.03	0.47	0.23
	SafeFL-CL	0.83	0.03	0.98	0.04	0.68	0.02	0.52	0.04	0.52	0.02
Adaptive attack	FLAME	0.59	NA	0.75	NA	0.44	NA	0.32	NA	0.29	NA
	FLDetector	0.65	NA	0.80	NA	0.57	NA	0.48	NA	0.29	NA
	FLTrust	0.62	NA	0.74	NA	0.37	NA	0.44	NA	0.41	NA
	DeepSight	0.62	NA	0.84	NA	0.52	NA	0.39	NA	0.43	NA
	BackdoorIndicator	0.48	NA	0.90	NA	0.58	NA	0.38	NA	0.44	NA
	FreqFed	0.79	NA	0.93	NA	0.62	NA	0.47	NA	0.42	NA
	FedREDefense	0.78	NA	0.90	NA	0.59	NA	0.45	NA	0.42	NA
	SafeFL-ML	0.78	NA	0.94	NA	0.62	NA	0.48	NA	0.49	NA
	SafeFL-CL	0.80	NA	0.97	NA	0.63	NA	0.50	NA	0.51	NA

under no attack, SafeFL-CL achieves a perfect DACC of 1.00, indicating its capability to maintain high detection accuracy without attack. SafeFL-ML also can recognize the most benign clients. However, other detection-based approaches like FLAME and FLDetector can recognize only half of the benign clients.

Second, in the presence of malicious clients, our proposed SafeFL effectively detects the majority of them while minimizing false detections of benign clients. For instance, under the hybrid attack strategy, the Trim+DBA attack poses a significant challenge, yet SafeFL-CL achieves a robust DACC of 0.96 on CIFAR-10, showcasing its resilience. In contrast, other baselines struggle significantly

Table 3: The performance of the final global model obtained through various prevention-based methods and our approach, where TACC (↑) and ASR (↓) metrics are considered. Here, ↑ denotes better performance with higher values, and ↓ denotes better performance with lower values.

Attack	Defense	CIFAR-10		MNIST		FEMNIST		STL-10		Tiny-ImageNet	
		TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR
No attack	Median	0.71	NA	0.96	NA	0.63	NA	0.47	NA	0.45	NA
	TrMean	0.70	NA	0.94	NA	0.62	NA	0.49	NA	0.44	NA
	Krum	0.65	NA	0.84	NA	0.59	NA	0.44	NA	0.39	NA
	SafeFL-ML	0.82	NA	0.98	NA	0.67	NA	0.53	NA	0.51	NA
	SafeFL-CL	0.84	NA	0.98	NA	0.68	NA	0.54	NA	0.54	NA
Trim attack	Median	0.27	NA	0.52	NA	0.35	NA	0.19	NA	0.09	NA
	TrMean	0.23	NA	0.44	NA	0.29	NA	0.15	NA	0.15	NA
	Krum	0.18	NA	0.60	NA	0.39	NA	0.11	NA	0.17	NA
	SafeFL-ML	0.80	NA	0.97	NA	0.67	NA	0.50	NA	0.50	NA
	SafeFL-CL	0.81	NA	0.97	NA	0.67	NA	0.52	NA	0.51	NA
Scaling attack	Median	0.67	0.75	0.91	0.82	0.61	0.52	0.44	0.72	0.44	0.49
	TrMean	0.65	0.81	0.90	0.75	0.61	0.68	0.42	0.58	0.41	0.65
	Krum	0.67	0.08	0.79	0.07	0.62	0.09	0.45	0.16	0.37	0.16
	SafeFL-ML	0.81	0.03	0.96	0.04	0.66	0.07	0.50	0.04	0.49	0.05
	SafeFL-CL	0.79	0.03	0.98	0.02	0.67	0.06	0.52	0.03	0.52	0.04
DBA attack	Median	0.64	0.24	0.94	0.29	0.64	0.32	0.41	0.29	0.45	0.23
	TrMean	0.66	0.24	0.93	0.34	0.58	0.27	0.45	0.15	0.41	0.24
	Krum	0.66	0.16	0.90	0.11	0.64	0.05	0.46	0.07	0.44	0.26
	SafeFL-ML	0.78	0.20	0.98	0.02	0.66	0.04	0.51	0.05	0.50	0.03
	SafeFL-CL	0.82	0.05	0.97	0.03	0.67	0.05	0.52	0.04	0.52	0.00
Trim+DBA attack	Median	0.22	0.14	0.65	0.27	0.25	0.17	0.22	0.14	0.19	0.07
	TrMean	0.16	0.18	0.62	0.18	0.21	0.19	0.19	0.15	0.21	0.11
	Krum	0.09	0.03	0.67	0.06	0.16	0.05	0.19	0.06	0.11	0.06
	SafeFL-ML	0.79	0.02	0.98	0.07	0.67	0.04	0.47	0.11	0.50	0.05
	SafeFL-CL	0.83	0.03	0.98	0.02	0.66	0.01	0.52	0.05	0.53	0.00
Scaling+DBA attack	Median	0.64	0.57	0.90	0.77	0.64	0.38	0.46	0.52	0.45	0.43
	TrMean	0.65	0.67	0.93	0.77	0.65	0.58	0.48	0.48	0.45	0.55
	Krum	0.66	0.09	0.82	0.04	0.66	0.07	0.41	0.09	0.42	0.03
	SafeFL-ML	0.80	0.02	0.97	0.17	0.67	0.04	0.51	0.03	0.47	0.23
	SafeFL-CL	0.83	0.03	0.98	0.04	0.68	0.02	0.52	0.04	0.52	0.02
Adaptive attack	Median	0.21	NA	0.51	NA	0.23	NA	0.25	NA	0.25	NA
	TrMean	0.14	NA	0.49	NA	0.27	NA	0.29	NA	0.28	NA
	Krum	0.11	NA	0.42	NA	0.13	NA	0.07	NA	0.06	NA
	SafeFL-ML	0.78	NA	0.94	NA	0.62	NA	0.48	NA	0.49	NA
	SafeFL-CL	0.80	NA	0.97	NA	0.63	NA	0.50	NA	0.51	NA

under this attack. On CIFAR-10, their DACC scores are limited to 0.75, 0.78, and 0.76, with corresponding FPR of 0.45, 0.39, and 0.27 for DeepSight, BackdoorIndicator, and FedREDefense, respectively, indicating a tendency to misclassify many benign clients as malicious. Similarly, the Scaling+DBA attack continues to mislead FLDetector, FLTrust, and FreqFed, resulting in DACC values no larger than 0.87 on both CIFAR-10 and MNIST datasets. In contrast, SafeFL-ML demonstrates robust performance, achieving DACC values of 0.95 and 0.97 on CIFAR-10 and MNIST, respectively, closely approaching the DACC values of 0.98 and 1.00 achieved by SafeFL-CL. In addition, other baseline methods fail to detect malicious clients effectively when the FL training process is applied to STL-10 and Tiny-ImageNet under the Scaling+DBA attack.

Table 2 presents the TACC and ASR of the final global model obtained using various detection methods. It is important to note that the ASR metric is relevant only for targeted attacks, including the Scaling attack, DBA attack, Trim+DBA attack, and Scaling+DBA attack. From the table, we observe that SafeFL-CL effectively defends against diverse attack types, achieving high DACC while maintaining elevated TACC and low ASR across multiple datasets. In contrast, other prevention-based methods fail to sustain high task accuracy. For example, on the MNIST dataset, the TACC of DeepSight under no attack is 0.98 but drops significantly to 0.84 under the Adaptive attack. This indicates that the global model trained using DeepSight lacks accuracy. Conversely, global models trained with SafeFL under different attack scenarios remain almost as accurate as those trained in the absence of attacks. For example, on the CIFAR-10 dataset, the TACC of SafeFL-ML remains consistently at 0.82 under no attack, and it can be maintained at 0.80 under the strong Trim attack. Table 3 highlights the TACC and ASR of the final global models produced by various prevention-based defense mechanisms. These defenses are generally ineffective in mitigating the impact of malicious clients. For instance, with the TrMean method on CIFAR-10, the TACC drops from 0.70 in the absence of attacks to 0.23 under the Trim attack, rendering the resulting global model highly inaccurate. In summary, compared to other detection-based and prevention-based methods, the final global model trained using our SafeFL demonstrates superior accuracy, thereby achieving the "Effective Threat Mitigation" objective.

Figures 3–7 in Appendix show the loss values of local models for benign and malicious clients, evaluated on the synthetic dataset using SafeFL-ML, under six attacks across five datasets at the 750th training round. Similarly, Figures 8–12 in Appendix present the corresponding results for SafeFL-CL under the same settings. Note that, by default, our experimental setup assumes that the first 30% of clients are malicious. As shown in Figures 3–12, the loss values of malicious local models are significantly higher than those of benign models. This observation reinforces the motivation behind our method: malicious local models often exhibit distinct loss patterns compared to benign ones, making their detection feasible.

Impact of the fraction of malicious clients: Table 4a presents the DACC of various detection-based methods as the fraction of malicious clients ranges from 0% to 40%, considering the Trim attack, Scaling attack, DBA attack, and the CIFAR-10 dataset. The results for Trim+DBA attack, Scaling+DBA attack, and Adaptive attack are shown in Table 10a in Appendix. Note that for all the ablation study experiments, unless otherwise specified, only the DACC values are reported. As illustrated in Table 4a and Table 10a, FLAME exhibits substantial variations in response to changes in the fraction of malicious clients. On the other hand, as the proportion of malicious clients increases from 0% to 40%, the DACC of SafeFL-ML remains consistently at least 0.90 under both Trim and Trim+DBA. Similarly, SafeFL-CL consistently achieves a DACC of at least 0.95 across all attack scenarios, demonstrating superior stability and performance compared to the other methods.

Impact of the total number of clients: Table 4b shows the DACC under Trim attack, Scaling attack, DBA attack for different detection-based methods with different total numbers of clients. The results of the Trim+DBA attack, Scaling+DBA attack, and Adaptive attack are presented in Table 10b in the Appendix. The fraction of malicious clients is still set to 30% by default. As the number of total clients increases, all method performance generally remains stable. For our SafeFL-CL under Trim attack, which achieves its highest DACC at 60 and 100 total clients. Additionally, SafeFL-CL maintains a significant advantage, with almost all of its DACC values outperforming those of the other methods. However, FLAME and FLTrust consistently exhibit a DACC no larger than 0.86 under the Trim attack, regardless of the total number of clients.

Table 4: The impact of the malicious client ratio, total client number, and Non-IID degree is analyzed using the CIFAR-10 dataset. DACC values are reported for the Trim attack, Scaling attack, and DBA attack. The results of Trim+DBA attack, Scaling+DBA attack, and Adaptive attack are shown in Table 10 in Appendix. "BDIndicator" refers to the "BackdoorIndicator" method.

(a) Impact of fraction of malicious clients.

Attack	Defense	Malicious client ratio				
		0%	10%	20%	30%	40%
Trim attack	FLAME	0.57	0.62	0.73	0.78	0.81
	FLDetector	0.65	0.96	0.95	0.96	0.94
	FLTrust	0.80	0.90	0.91	0.85	0.87
	DeepSight	1.00	0.89	0.88	0.88	0.87
	BDIndicator	0.95	0.73	0.75	0.73	0.76
	FreqFed	0.81	0.84	0.86	0.89	0.87
	FedREDefense	0.78	0.85	0.89	0.85	0.83
	SafeFL-ML	0.94	0.94	0.94	0.90	0.97
	SafeFL-CL	1.00	0.99	1.00	1.00	0.97
Scaling attack	FLAME	0.57	0.69	0.78	0.84	0.89
	FLDetector	0.65	0.96	0.98	1.00	0.75
	FLTrust	0.80	0.72	0.70	0.75	0.82
	DeepSight	1.00	0.82	0.85	0.88	0.92
	BDIndicator	0.95	0.95	0.97	0.94	0.93
	FreqFed	0.81	0.84	0.86	0.84	0.87
	FedREDefense	0.78	0.84	0.87	0.87	0.86
	SafeFL-ML	0.94	0.97	0.94	0.91	0.99
	SafeFL-CL	1.00	0.99	1.00	1.00	1.00
DBA attack	FLAME	0.57	0.70	0.74	0.82	0.84
	FLDetector	0.65	0.91	0.88	0.89	0.92
	FLTrust	0.80	0.84	0.87	0.80	0.82
	DeepSight	1.00	0.85	0.87	0.88	0.90
	BDIndicator	0.95	0.95	0.94	1.00	0.97
	FreqFed	0.81	0.87	0.89	0.89	0.89
	FedREDefense	0.78	0.72	0.77	0.75	0.75
	SafeFL-ML	0.94	0.93	0.97	0.94	0.95
	SafeFL-CL	1.00	0.97	0.97	1.00	1.00

(b) Impact of total number of clients.

Attack	Defense	Total client number				
		60	80	100	120	150
Trim attack	FLAME	0.74	0.79	0.78	0.82	0.83
	FLDetector	0.90	0.95	0.96	0.97	0.94
	FLTrust	0.79	0.84	0.85	0.85	0.86
	DeepSight	0.85	0.87	0.88	0.87	0.85
	BDIndicator	0.71	0.76	0.73	0.74	0.73
	FreqFed	0.85	0.85	0.89	0.87	0.89
	FedREDefense	0.88	0.88	0.85	0.86	0.87
	SafeFL-ML	0.94	0.96	0.90	0.94	0.92
	SafeFL-CL	1.00	0.97	1.00	0.95	0.99
Scaling attack	FLAME	0.83	0.82	0.84	0.80	0.84
	FLDetector	0.90	0.93	1.00	0.95	0.92
	FLTrust	0.74	0.75	0.75	0.77	0.74
	DeepSight	0.87	0.85	0.88	0.86	0.83
	BDIndicator	0.95	0.92	0.94	0.97	0.90
	FreqFed	0.85	0.86	0.84	0.84	0.85
	FedREDefense	0.84	0.85	0.87	0.87	0.83
	SafeFL-ML	0.94	0.97	0.94	0.91	0.99
	SafeFL-CL	1.00	0.99	1.00	1.00	1.00
DBA attack	FLAME	0.83	0.85	0.82	0.81	0.84
	FLDetector	0.87	0.85	0.89	0.88	0.88
	FLTrust	0.80	0.81	0.80	0.82	0.76
	DeepSight	0.83	0.84	0.88	0.87	0.84
	BDIndicator	0.97	0.92	1.00	0.94	0.95
	FreqFed	0.85	0.86	0.89	0.89	0.84
	FedREDefense	0.75	0.77	0.75	0.77	0.76
	SafeFL-ML	0.95	0.94	0.94	0.97	0.94
	SafeFL-CL	1.00	0.97	1.00	0.99	0.98

(c) Impact of degree of Non-IID.

Attack	Defense	Non-IID degree				
		0.1	0.3	0.5	0.7	0.9
Trim attack	FLAME	0.71	0.76	0.78	0.79	0.77
	FLDetector	0.89	0.87	0.96	0.98	1.00
	FLTrust	0.81	0.85	0.85	0.82	0.89
	DeepSight	0.82	0.84	0.88	0.87	0.94
	BDIndicator	0.71	0.69	0.73	0.77	0.79
	FreqFed	0.80	0.82	0.89	0.89	0.89
	FedREDefense	0.89	0.83	0.85	0.78	0.86
	SafeFL-ML	0.93	0.91	0.90	0.93	0.95
	SafeFL-CL	0.95	0.99	1.00	0.97	0.97
Scaling attack	FLAME	0.82	0.84	0.84	0.81	0.90
	FLDetector	0.86	0.95	1.00	0.97	0.92
	FLTrust	0.77	0.76	0.75	0.75	0.78
	DeepSight	0.72	0.75	0.88	0.82	0.84
	BDIndicator	0.88	0.92	0.94	0.96	0.86
	FreqFed	0.80	0.85	0.87	0.89	0.89
	FedREDefense	0.89	0.83	0.91	0.78	0.86
	SafeFL-ML	0.91	0.93	0.90	0.95	0.95
	SafeFL-CL	0.93	0.94	1.00	1.00	1.00
DBA attack	FLAME	0.75	0.79	0.82	0.84	0.83
	FLDetector	0.82	0.85	0.89	0.90	0.92
	FLTrust	0.82	0.83	0.80	0.81	0.82
	DeepSight	0.84	0.85	0.88	0.87	0.94
	BDIndicator	0.95	0.97	1.00	0.97	0.98
	FreqFed	0.86	0.88	0.89	0.89	0.91
	FedREDefense	0.72	0.77	0.75	0.76	0.79
	SafeFL-ML	0.91	0.91	0.94	0.97	0.99
	SafeFL-CL	0.93	0.95	1.00	1.00	0.98

Impact of degree of Non-IID: Table 4c presents the DACC results for different detection-based defense methods across Non-IID levels ranging from 0.1 to 0.9, consider the Trim attack, Scaling attack, DBA attack. Table 10c in the Appendix presents the results for the Trim+DBA attack, Scaling+DBA attack, and Adaptive attack. According to Table 4c and Table 10c, FLDetector is significantly affected by the degree of Non-IID. The reason is that when the clients' training data are highly heterogeneous, the server in FLDetector faces difficulty in predicting clients' local models using their historical information. Regardless of the degree of Non-IID, SafeFL-CL consistently maintains the highest DACC among all methods. Under the Trim attack, when the Non-IID value is 0.1, SafeFL-CL outperforms FLAME by 0.24.

Impact of the selection rate: Under our default configuration, all clients are assumed to participate in every training round. In this section, we explore a more practical scenario in which the server randomly selects only a fraction of clients to engage in each round. In this setting, a malicious client can carry out an attack only if it is selected. Table 11 in Appendix presents the detection performance of various methods on the CIFAR-10 dataset under different client selection rates. As shown, our proposed detection method remains effective at identifying malicious clients even when only a subset of clients participate in each round.

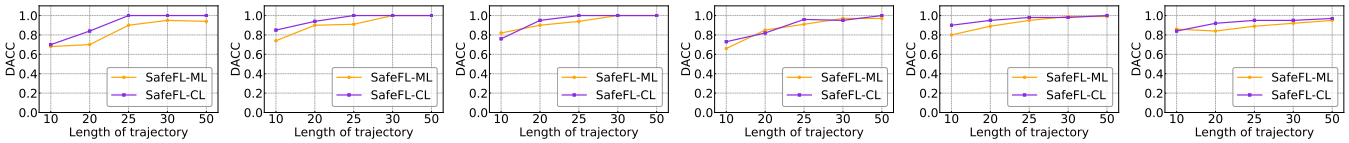
Impact of trajectory length: Figure 1 illustrates the impact of trajectory length, defined as the number of global models used to generate synthetic data or the value of ϵ, on the detection of malicious clients in SafeFL. Figure 1 reveals a positive correlation between trajectory length and DACC for both SafeFL-ML and SafeFL-CL. A longer trajectory length consistently enhances performance and improves DACC for SafeFL. However, when the trajectory length reaches 25, the improvement in DACC begins to plateau.

Impact of number of synthetic data: Figure 2 depicts the effect of the number of synthetic data on SafeFL's detection performance. The number of synthetic data refers to the total number of examples in the synthetic dataset. Similar to the effect of trajectory length, there is a positive correlation between the amount of synthetic data and DACC. Notably, a larger volume of synthetic data more significantly enhances SafeFL-CL's DACC. For instance, under the Trim+DBA attack, increasing the synthetic data from 10 to 150 raises SafeFL-CL's DACC from 0.72 to 0.98, whereas SafeFL-ML's DACC increases from 0.63 to 0.92.

Different variants of SafeFL-CL: SafeFL-CL performs clustering twice, as indicated in Line 15 and Line 30 of Algorithm 2. We refer to these two clustering instances as A & B, where A corresponds to the clustering algorithm used in Line 15, and B pertains to the clustering algorithm applied to the losses (see Line 30). Table 5 examines the performance of various variants of our SafeFL-CL. In the different variants, we apply various clustering algorithms, such as K-means [32], Mean-shift [19] or DBSCAN [11], to both A and B. As shown in Table 5, the "Mean-shift & Mean-shift" variant exhibits poor detection performance against the Trim attack, whereas the "Kmeans & Mean-shift" variant (which corresponds to our proposed SafeFL-CL) achieves the best detection results.

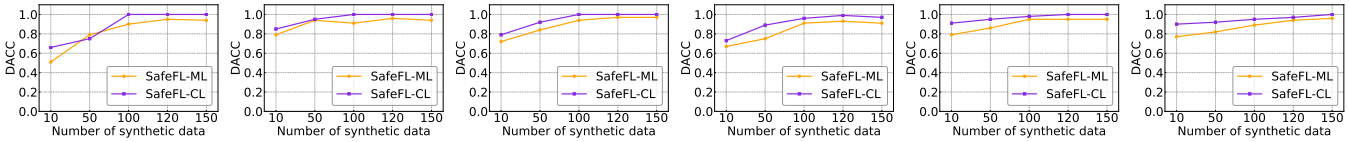
6 Discussion and Limitations

More extreme Non-IID distribution: This section examines a more extreme Non-IID scenario, as detailed in [48]. The training data distribution among clients is purely label-based, with each



(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 1: Impact of length of trajectory, where CIFAR-10 dataset is considered.



(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 2: Impact of the number of synthetic data, where CIFAR-10 dataset is considered.

Table 5: Different variants of SafeFL-CL on CIFAR-10, with DACC values reported.

	K-means & K-means	Mean-shift & Mean-shift	K-means & DBSCAN	K-means & Mean-shift (SafeFL-CL)
Trim attack	0.98	0.79	0.70	1.00
Scaling attack	0.79	0.83	0.87	1.00
DBA attack	0.94	0.72	0.75	1.00
Trim+DBA attack	0.92	0.75	0.72	0.96
Scaling+DBA attack	0.83	0.75	0.80	0.98
Adaptive attack	0.92	0.79	0.72	0.95

Table 6: Detection results on CIFAR-10, with each client having three classes of training data and DACC values reported.

Attack	FLAME	FLDetector	FLTrust	DeepSight	BackdoorIndicator	FreqFed	FedREDefense	SafeFL-ML	SafeFL-CL
Trim attack	0.75	0.92	0.85	0.78	0.65	0.84	0.82	0.90	0.97
Scaling attack	0.80	0.84	0.82	0.87	0.89	0.87	0.78	0.88	0.96
DBA attack	0.81	0.72	0.79	0.85	0.74	0.72	0.84	0.93	0.99
Trim+DBA attack	0.79	0.85	0.73	0.79	0.72	0.79	0.81	0.87	0.96
Scaling+DBA attack	0.80	0.78	0.72	0.82	0.79	0.80	0.81	0.91	0.99
Adaptive attack	0.67	0.74	0.70	0.77	0.72	0.77	0.80	0.89	0.92

Table 7: DACC of SafeFL with different aggregation rules on CIFAR-10.

Attack	Defense	Median	TrMean	Krum
Trim attack	SafeFL-ML	0.95	0.96	0.93
	SafeFL-CL	1.00	0.99	0.98
Scaling attack	SafeFL-ML	0.97	0.93	0.88
	SafeFL-CL	1.00	0.99	0.98
DBA attack	SafeFL-ML	0.92	0.92	0.95
	SafeFL-CL	0.94	1.00	1.00
Trim+DBA attack	SafeFL-ML	0.87	0.91	0.91
	SafeFL-CL	1.00	0.98	0.98
Scaling+DBA attack	SafeFL-ML	0.92	0.94	0.92
	SafeFL-CL	0.98	0.96	0.98
Adaptive attack	SafeFL-ML	0.88	0.87	0.85
	SafeFL-CL	0.91	0.90	0.94

client receiving data from only three specific classes. For example, Client A's training dataset consists solely of labels from 0 to 2, whereas Client B's dataset is restricted to labels from 3 to 5. Under such condition, detection accuracy results of various methods are shown in Table 6. Our method still significantly outperforms existing approaches, demonstrating the effectiveness of our SafeFL under extreme Non-IID setting.

More untargeted attacks: In this section, we use extra sophisticated untargeted attacks to further examine our method's detection ability. We implement the experiments on CIFAR-10. Table 12 shows the detection results of different methods for the Label flipping attack [62], and "A little is enough" (LIE) attack [6]. Note that in the Label Flipping attack, the attacker alters the labels of training examples on malicious clients. In the LIE attack, the attacker strategically introduces small perturbations to the local models of malicious clients to avoid detection. Table 12 in Appendix shows that our SafeFL, particularly SafeFL-CL, achieves a perfect DACC of 1.00 against both advanced attacks. In contrast, BackdoorIndicator performs poorly under the Label flipping attack.

More backdoor attacks: To further assess the robustness of our proposed detection methods, we evaluate them against three advanced backdoor attacks: Neurotoxin attack [75], Irreversible backdoor attack [53], and Clean-label backdoor attack [72]. Table 13 in Appendix reports the detection performance of our methods alongside all baseline detection approaches across five datasets, while Table 14 in Appendix presents the performance of the final global models trained using each detection method. As shown in both tables, baseline methods struggle to effectively identify and mitigate these sophisticated attacks. For example, FLTrust yields a high false positive rate (FPR) of 0.49 under the Irreversible backdoor attack on CIFAR-10 dataset. In contrast, our methods maintain strong detection capabilities and remain robust even in the presence of these advanced threats.

More evaluation metrics: To provide a more comprehensive assessment of global model performance, we also incorporate three additional metrics: precision, recall, and F1-score. For these metrics, higher values indicate better detection effectiveness. Table 15 in Appendix presents the precision, recall, and F1-score of various detection methods under six standard attacks (Trim, Scaling, DBA, Trim+DBA, Scaling+DBA, and Adaptive attacks), while Table 16 in Appendix shows the corresponding results for three advanced backdoor attacks (Neurotoxin, Irreversible backdoor, and Clean-label backdoor attacks). As shown in both tables, our proposed detection

methods consistently achieve high precision, recall, and F1-scores across different attacks and datasets.

Computational overhead and storage usage of different methods: In our methods, the server constructs a synthetic dataset based on the trajectory of global models accumulated over multiple training rounds. While this step enhances detection capabilities, it may introduce additional computational and storage overhead. Figure 13 in Appendix presents the total running time across all datasets under the Trim attack, with similar patterns observed for other attack types. For our methods, the reported runtime includes the time required for synthetic data generation, clustering, and loss-based filtering. As shown, our methods incur only a modest increase in computational cost compared to FedAvg, whereas FedREDefense exhibits the highest computational overhead among all methods. Table 17 in Appendix summarizes the additional server-side storage requirements for FLDetector, FedREDefense, SafeFL-ML, and SafeFL-CL. Note that FedAvg and other baseline detection methods do not require any extra server storage. As shown in Table 17, our methods require no more than 22.40 GB of additional storage, which is considered reasonable for high-capacity servers such as those commonly deployed in modern data centers.

SafeFL uses different aggregation rules: By default, our proposed SafeFL utilizes the FedAvg rule to aggregate the detected benign local models. As observed in the experimental results in Section 5.2, using the FedAvg rule is sufficient, as SafeFL effectively detects the majority of malicious clients. In this section, we examine the scenario where SafeFL employs prevention-based methods such as Median, TrMean, or Krum to aggregate the detected benign local models. The results, presented in Table 7, indicate that SafeFL maintains strong detection performance even when prevention-based methods are used. For example, when SafeFL-CL applies the Median method for aggregation and Trim attack is considered, it achieves a DACC of 1.00.

Discussion on the threat model for the ratio of malicious clients: Table 4a and Table 10a demonstrate that our proposed SafeFL framework effectively detects malicious clients even when 40% of the clients are compromised. Fractions higher than 40%, such as 45% or 50%, were not included in our experiments, as such scenarios are considered impractical. As highlighted in [61], achieving such a high proportion of malicious clients is unlikely in real-world FL environments. The decentralized nature of FL systems makes it difficult or even impossible for the attacker to control such a significant fraction of participating clients for malicious activities.

Discussion of biases introduced by the synthetic dataset: In this section, we examine whether the synthetic dataset generated by our methods introduces bias into the global model. Ideally, the final model should perform equitably across groups defined by sensitive attributes, such as sex. To evaluate fairness, we use two standard metrics: Equalized odds [31], which measures bias conditioned on the true label, and Demographic parity [22], which requires predictions to be independent of the sensitive attribute. Our analysis is conducted on the CIFAR-10 dataset under two highly Non-IID scenarios. In Setting I, each client has access to samples from only three specific classes (note that further reducing this to one or two classes prevents convergence even without attacks). In Setting II, we simulate extreme data heterogeneity by setting the Non-IID degree to 0.1, as suggested in [24]. Since CIFAR-10 lacks predefined sensitive attributes, we define a proxy sensitive attribute by grouping labels based on parity (odd vs. even). Table 18 in Appendix presents the Equalized odds and Demographic parity scores of our methods under various attack conditions on CIFAR-10, with lower values indicating greater fairness. Detection performance under Setting I is reported in Table 6, while results under Setting II are provided in Table 4c and Table 10c. Table 18 reveals two key findings: (1) Highly Non-IID data induces bias in the global model even without attacks, consistent with prior work [4, 17] (e.g., FedAvg scores 0.79 in Equalized Odds under Setting I). (2) Our methods maintain similar fairness to FedAvg in the no-attack case, introducing no extra bias.

Potential challenges introduced by SafeFL: In our proposed method, the server collects and stores multiple global models, which are then utilized to generate the synthetic dataset. This approach enhances the framework's ability to identify malicious clients effectively. However, it also introduces potential privacy concerns, as the storage and usage of multiple global models may inadvertently expose sensitive information about the clients' data. Although addressing privacy is not the primary focus of this paper, these concerns can be effectively alleviated by leveraging well-established privacy-preserving techniques. As an example, incorporating differential privacy [3] allows for the protection of individual client data while still supporting the generation of synthetic datasets for detection purposes. In particular, each client applies differential privacy by adding noise to its local model before transmitting it to the server. In our experiments, the noise is sampled from a Gaussian distribution $N(0, \varrho)$, where ϱ denotes the noise level. Table 19 in the Appendix reports the impact of varying noise levels on the CIFAR-10 dataset, with results measured by DACC and TACC. As shown in the table, under a non-adversarial setting, excessive noise can negatively affect the global model's testing accuracy (TACC), even when using the FedAvg aggregation rule. For instance, when the noise level is set to 2, the TACC of FedAvg drops to 0.70, compared to 0.85 in the absence of noise. This highlights a trade-off: while differential privacy strengthens client data protection, it can also compromise model performance. Despite the presence of noise, our detection methods preserve high global model accuracy. In particular, the TACC values achieved by our methods remain close to those of the noise-free FedAvg baseline, suggesting that our methods effectively balances privacy protection and model utility.

7 Conclusion and Future Work

FL is vulnerable to poisoning attacks due to its decentralized nature. Existing detection methods often perform poorly. To address this, we propose SafeFL, a detection method where the server uses a synthetic dataset, generated from global model trajectories, to distinguish between benign and malicious clients. Experiments confirm its effectiveness. A limitation of SafeFL is potential privacy concerns from the server's actions. Future work will focus on privacy-preserving detection. We also plan to extend SafeFL to decentralized FL [7, 34], where no trusted central server exists.

Acknowledgments

We thank the anonymous reviewers for their comments.

References

[1] [n. d.]. *Federated Learning: Collaborative Machine Learning without Centralized Training Data.* https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[2] [n. d.]. *Utilization of FATE in Risk Management of Credit in Small and Micro Enterprises.* https://www.fedai.org/cases/utilization-of-fate-in-risk-management-of-credit-in-small-and-micro-\enterprises/

[3] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *CCS*.

[4] Maryam Badar, Sandipan Sikdar, Wolfgang Nejdl, and Marco Fisichella. 2024. Fairtrade: Achieving pareto-optimal trade-offs between balanced accuracy and fairness in federated learning. In *AAAI*.

[5] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *AISTATS*.

[6] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A little is enough: Circumventing defenses for distributed learning. In *NeurIPS*.

[7] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. 2022. Decentralized Federated Learning: Fundamentals, State-of-the-art, Frameworks, Trends, and Challenges. In *arXiv preprint arXiv:2211.08413*.

[8] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *ICML*.

[9] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *NeurIPS*.

[10] Keith Bonawitz. 2019. Towards federated learning at scale: System design. In *SysML*.

[11] Beyza Bozdemir, Sébastien Canard, Orhan Ermis, Helen Möllering, Melek Önen, and Thomas Schneider. 2021. Privacy-preserving density-based clustering. In *ASIACCS*.

[12] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konen, H. Brendan Mcmahan, Virginia Smith, and Ameet Talwalkar. 2019. LEAF: A Benchmark for Federated Settings. In *NeurIPS*.

[13] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *PAKDD*.

[14] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *NDSS*.

[15] Xiaoyu Cao and Neil Zhenqiang Gong. 2022. Mpaf: Model poisoning attacks to federated learning based on fake clients. In *CVPR Workshops*.

[16] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *CVPR*.

[17] Hongyan Chang and Reza Shokri. 2023. Bias propagation in federated learning. In *ICLR*.

[18] Min Chen, Yang Xu, Hongli Xu, and Liusheng Huang. 2023. Enhancing decentralized federated learning for non-iid data on heterogeneous devices. In *ICDE*.

[19] Yizong Cheng. 1995. Mean shift, mode seeking, and clustering. In *IEEE transactions on pattern analysis and machine intelligence*.

[20] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

[22] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS*.

[23] El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. 2021. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). In *NeurIPS*.

[24] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to Byzantine-robust federated learning. In *USENIX Security Symposium*.

[25] Minghong Fang, Jia Liu, Neil Zhenqiang Gong, and Elizabeth S Bentley. 2022. Aflguard: Byzantine-robust asynchronous federated learning. In *ACSASC*.

[26] Minghong Fang, Zhuqing Liu, Xuecen Zhao, and Jia Liu. 2025. Byzantine-Robust Federated Learning over Ring-All-Reduce Distributed Computing. In *The Web Conference*.

[27] Minghong Fang, Seyedsina Nabavirazavi, Zhuqing Liu, Wei Sun, Sundararaja Sitharama Iyengar, and Haibo Yang. 2025. Do We Really Need to Design New Byzantine-robust Aggregation Rules?. In *NDSS*.

[28] Minghong Fang, Xilong Wang, and Neil Zhenqiang Gong. 2025. Provably Robust Federated Reinforcement Learning. In *The Web Conference*.

[29] Minghong Fang, Zifan Zhang, Hairi, Prashant Khanduri, Jia Liu, Songtao Lu, Yuchen Liu, and Neil Gong. 2024. Byzantine-robust decentralized federated learning. In *CCS*.

[30] Hossein Fereidooni, Alessandro Pegoraro, Phillip Rieger, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. 2024. FreqFed: A Frequency Analysis-Based Approach for Mitigating Poisoning Attacks in Federated Learning. In *NDSS*.

[31] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*.

[32] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. In *Journal of the royal statistical society. series c (applied statistics)*.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[34] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and HR Tizhoosh. 2023. Decentralized federated learning through proxy model sharing. In *Nature Communications*.

[35] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. 2022. Byzantine-robust learning on heterogeneous datasets via bucketing. In *ICLR*.

[36] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*.

[37] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. 2022. Dataset condensation via efficient synthetic-data parameterization. In *ICML*.

[38] A. Krizhevsky and G. Hinton. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases* (2009).

[39] Kavita Kumari, Phillip Rieger, Hossein Fereidooni, Murtuza Jadliwala, and Ahmad-Reza Sadeghi. 2023. BayBFed: Bayesian Backdoor Defense for Federated Learning. In *IEEE Symposium on Security and Privacy*.

[40] Yann LeCun, Corinna Cortes, and CJ Burges. 1998. MNIST handwritten digit database. *Available: http://yann. lecun. com/exdb/mnist* (1998).

[41] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. 2019. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI*.

[42] Songze Li and Yanbo Dai. 2024. BackdoorIndicator: Leveraging OOD Data for Proactive Backdoor Detection in Federated Learning. In *USENIX Security Symposium*.

[43] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *ICML*.

[44] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *MLSys*.

[45] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020. Fair resource allocation in federated learning. In *ICLR*.

[46] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the convergence of fedavg on non-iid data. In *ICLR*.

[47] Songhua Liu, Jingwen Ye, Runpeng Yu, and Xinchao Wang. 2023. Slimmable dataset condensation. In *CVPR*.

[48] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.

[49] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic federated learning. In *ICML*.

[50] Hamid Mozaffari, Virat Shejwalkar, and Amir Houmansadr. 2023. Every Vote Counts: Ranking-Based Training of Federated Learning to Resist Poisoning Attacks. In *USENIX Security Symposium*.

[51] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. 2019. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125* (2019).

[52] Mohammad Naseri, Yufei Han, Enrico Mariconti, Yun Shen, Gianluca Stringhini, and Emiliano De Cristofaro. 2022. Cerberus: Exploring Federated Prediction of Security Events. In *CCS*.

[53] Thuy Dung Nguyen, Tuan A Nguyen, Anh Tran, Khoa D Doan, and Kok-Seng Wong. 2023. Iba: Towards irreversible backdoor attacks in federated learning. In *NeurIPS*.

[54] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. 2022. FLAME: Taming backdoors in federated learning. In *USENIX Security Symposium*.

[55] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. 2021. Defending against backdoors in federated learning with robust learning rate. In *AAAI*.

[56] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. 2021. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503* (2021).

[57] Renjie Pi, Weizhong Zhang, Yueqi Xie, Jiahui Gao, Xiaoyu Wang, Sunghun Kim, and Qifeng Chen. 2023. Dynafed: Tackling client data heterogeneity with global dynamics. In *CVPR*.

[58] Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. 2019. DETOX: A redundancy-based framework for faster and more robust gradient aggregation. In *NeurIPS*.

[59] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. 2022. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. In *NDSS*.
[60] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.
[61] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. 2022. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *IEEE Symposium on Security and Privacy*.
[62] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *ESORICS*.
[63] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *NeurIPS*.
[64] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. 2022. Cafe: Learning to condense dataset by aligning features. In *CVPR*.
[65] Ning Wang, Yang Xiao, Yimin Chen, Yang Hu, Wenjing Lou, and Y Thomas Hou. 2022. Flare: defending federated learning against model poisoning attacks via latent space representations. In *ASIACCS*.
[66] Wenbin Wang, Qiwen Ma, Zifan Zhang, Yuchen Liu, Zhuqing Liu, and Minghong Fang. 2025. Poisoning Attacks and Defenses to Federated Unlearning. In *The Web Conference*.
[67] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. Dba: Distributed backdoor attacks against federated learning. In *ICLR*.
[68] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *ICML*.
[69] Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. 2024. FedREDefense: Defending against Model Poisoning Attacks for Federated Learning using Model Update Reconstruction Error. In *ICML*.
[70] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *ICML*.
[71] Ming Yin, Yichang Xu, Minghong Fang, and Neil Zhenqiang Gong. 2024. Poisoning federated recommender systems with fake users. In *The Web Conference*.
[72] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. 2023. Narcissus: A practical clean-label backdoor attack with limited information. In *CCS*.
[73] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *KDD*.
[74] Zifan Zhang, Minghong Fang, Jiayuan Huang, and Yuchen Liu. 2024. Poisoning attacks on federated learning-based wireless traffic prediction. In *IFIP Networking Conference*.
[75] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. 2022. Neurotoxin: Durable backdoors in federated learning. In *ICML*.
[76] Bo Zhao and Hakan Bilen. 2023. Dataset condensation with distribution matching. In *WACV*.
[77] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset condensation with gradient matching. In *ICLR*.
[78] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*.

Table 8: The CNN architecture.

Layer	Size
Input	$28 \times 28 \times 1$
Convolution + ReLU	$3 \times 3 \times 30$
Max Pooling	2×2
Convolution + ReLU	$3 \times 3 \times 5$
Max Pooling	2×2
Fully Connected + ReLU	100
Soft	10 (62 for FEMNIST)

A Details of Datasets, Poisoning Attacks, Compared Defenses

A.1 Details of Datasets

a) CIFAR-10 [38]: The CIFAR-10 dataset is a color image classification dataset containing 50,000 training examples and 10,000 testing examples, each categorized into one of ten classes.

b) MNIST [40]: MNIST dataset contains 10 different classes and includes 60,000 examples for training and 10,000 for testing.

c) FEMNIST [12]: The FEMNIST dataset, derived from the extended MNIST dataset, is specifically designed for FL purposes. It contains a meticulously selected collection of handwritten character images, encompassing a diverse range of characters and digits, totaling 62 distinct classes. This dataset is inherently heterogeneous.

d) STL-10 [20]: The STL-10 dataset comprises 13,000 labeled images distributed among 10 object classes such as birds, cats, and trucks. Among these, 5,000 images are allocated for training, while the remaining 8,000 are reserved for testing.

e) Tiny-ImageNet [21]: Tiny-ImageNet is a subset of the ImageNet dataset, comprising 100,000 images distributed across 200 classes, with 500 images per class.

A.2 Details of Poisoning Attacks

a) Trim attack [24]: The Trim attack is an untargeted local model poisoning attack specifically designed to manipulate the Trimmed-mean and Median aggregation rules. We adopt the default parameter settings outlined in [24] to execute the Trim attack.

b) Scaling attack [5]: For this targeted attack, the attacker duplicates local training instances on malicious clients, adds a trigger, and assigns a chosen label. Local models are then computed using the augmented training data. Malicious clients amplify these local models before sending them to the server.

c) Distributed backdoor (DBA) attack [67]: DBA attack involves dividing the trigger pattern into four segments. These segments are then incorporated into the training data of four separate groups of malicious clients. Each client calculates its own local model and adjusts it using a scaling factor.

d) Trim+DBA attack: This strategy involves a hybrid attack strategy where various malicious clients employ different methods to craft their local models. In our experiments, half of the malicious clients utilized the Trim attack to shape their local models, whereas the other half implemented the DBA strategy.

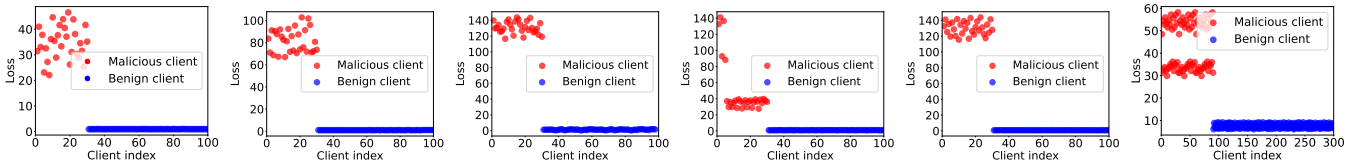
e) Scaling+DBA attack: In this hybrid attack, half of the malicious clients used the Scaling attack to craft their local models, while the other half employed the DBA attack.

e) Adaptive attack [60]: In the worst-case scenario, the attacker has complete information about the FL system, including the local models of all clients and the server's aggregation method, such as the SafeFL described in our work. Leveraging this knowledge, the attacker designs an adaptive attack to disrupt and deceive the FL process. In our experiments, we implement the Adaptive attack following the methodology outlined in [60].

A.3 Details of Compared Defenses

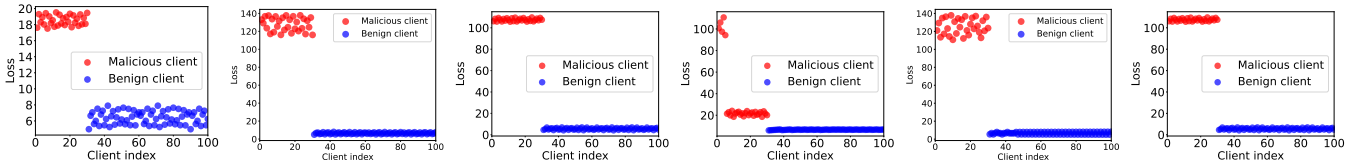
a) FLAME [54]: FLAME is a defense strategy against targeted attacks, including backdoor attacks. It uses clustering to remove suspected malicious local models, truncates the remaining models to limit their influence, and adds random noise to the aggregated model to eliminate backdoors.

b) FLDetector [73]: FLDetector identifies malicious clients by analyzing the consistency of their local models. It leverages the



(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 3: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-ML with the CIFAR-10 dataset.



(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 4: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-ML with the MNIST dataset.

observation that benign clients follow the FL algorithm and their local data, while malicious clients deviate by crafting inconsistent models across training rounds.

c) FLTrust [14]: FLTrust assumes the server has a clean validation dataset from the same distribution as the clients' training data. The server trains a server model on this dataset, and a client's local model is deemed benign if it aligns positively with the server model.

d) DeepSight [59]: DeepSight analyzes model updates, examining output parameters and data homogeneity to detect backdoor attacks. Using classifiers and clustering, it distinguishes malicious updates from benign ones, even with diverse data distributions.

e) BackdoorIndicator [42]: This proactive FL backdoor detection method uses out-of-distribution (OOD) data to identify malicious updates. The server injects an OOD indicator task into the global model, and after client training, evaluates its accuracy, adjusting for batch normalization shifts. Updates exceeding a set accuracy threshold are flagged as suspicious and excluded from aggregation.

f) FreqFed [30]: FreqFed mitigates targeted and untargeted poisoning attacks by analyzing model weights in the frequency domain, detecting and removing malicious updates while maintaining global model performance.

g) FedREDefense [69]: FedREDefense detects malicious models by measuring the reconstruction error of each client's model. Using distilled local knowledge, it reconstructs models and compares the error to a threshold. Updates with errors exceeding the threshold are flagged as malicious.

h) Median [70]: For every dimension, the server calculates the median value for each coordinate from all clients' local models.

i) Trimmed mean (TrMean) [70]: Like the Median method, the server removes the largest and smallest k values for each dimension, then averages the remaining n values, where k is the number of malicious clients, and n is the total number of clients.

j) Krum [9]: Upon receiving local models from clients, the server outputs a single local model that minimizes the sum of distances to its neighboring subset.

Table 10: The impact of the malicious client ratio, total client number, and Non-IID degree is analyzed using the CIFAR-10 dataset. DACC values are reported for the Trim+DBA attack, Scaling+DBA attack, and Adaptive attack. "BDIndicator" refers to the "BackdoorIndicator" method.

(a) Impact of fraction of malicious clients.

Attack	Defense	Malicious client ratio				
		0%	10%	20%	30%	40%
Trim+DBA attack	FLAME	0.57	0.74	0.77	0.81	0.87
	FLDetector	0.65	0.82	0.89	0.87	0.84
	FLTrust	0.80	0.83	0.91	0.88	0.85
	DeepSight	1.00	0.72	0.79	0.75	0.83
	BDIndicator	0.95	0.75	0.72	0.78	0.80
	FreqFed	0.81	0.83	0.72	0.80	0.76
	FedREDefense	0.78	0.79	0.75	0.76	0.80
	SafeFL-ML	0.94	0.93	0.95	0.91	0.95
	SafeFL-CL	1.00	0.99	0.95	0.96	0.97
Scaling+DBA attack	FLAME	0.57	0.75	0.82	0.89	0.93
	FLDetector	0.65	0.69	0.62	0.69	0.74
	FLTrust	0.80	0.85	0.74	0.76	0.72
	DeepSight	1.00	0.78	0.80	0.86	0.82
	BDIndicator	0.95	0.89	0.97	0.94	0.95
	FreqFed	0.81	0.86	0.83	0.84	0.87
	FedREDefense	0.78	0.89	0.84	0.85	0.87
	SafeFL-ML	0.94	0.97	0.92	0.95	0.96
	SafeFL-CL	1.00	0.99	0.97	0.98	1.00
Adaptive attack	FLAME	0.57	0.68	0.74	0.77	0.82
	FLDetector	0.65	0.72	0.79	0.85	0.84
	FLTrust	0.80	0.76	0.74	0.70	0.62
	DeepSight	1.00	0.82	0.75	0.77	0.71
	BDIndicator	0.95	0.75	0.76	0.71	0.70
	FreqFed	0.81	0.84	0.80	0.82	0.82
	FedREDefense	0.78	0.80	0.86	0.85	0.88
	SafeFL-ML	0.94	0.84	0.89	0.89	0.93
	SafeFL-CL	1.00	0.93	0.94	0.95	0.91

(b) Impact of total number of clients.

Attack	Defense	Total client number				
		60	80	100	120	150
Trim+DBA attack	FLAME	0.82	0.85	0.81	0.83	0.84
	FLDetector	0.84	0.86	0.87	0.85	0.82
	FLTrust	0.84	0.85	0.88	0.81	0.84
	DeepSight	0.77	0.74	0.75	0.79	0.83
	BDIndicator	0.72	0.76	0.78	0.76	0.75
	FreqFed	0.81	0.84	0.80	0.80	0.77
	FedREDefense	0.75	0.76	0.76	0.75	0.75
	SafeFL-ML	0.95	0.90	0.91	0.94	0.95
	SafeFL-CL	0.96	0.99	0.96	1.00	0.99
Scaling+DBA attack	FLAME	0.82	0.85	0.81	0.83	0.84
	FLDetector	0.84	0.86	0.87	0.85	0.82
	FLTrust	0.84	0.85	0.88	0.81	0.84
	DeepSight	0.77	0.74	0.75	0.79	0.83
	BDIndicator	0.72	0.76	0.78	0.76	0.75
	FreqFed	0.81	0.84	0.80	0.80	0.77
	FedREDefense	0.75	0.76	0.77	0.75	0.75
	SafeFL-ML	0.95	0.90	0.91	0.94	0.95
	SafeFL-CL	0.96	0.99	0.96	1.00	0.99
Adaptive attack	FLAME	0.80	0.74	0.77	0.75	0.78
	FLDetector	0.76	0.83	0.85	0.82	0.84
	FLTrust	0.74	0.72	0.70	0.73	0.75
	DeepSight	0.76	0.78	0.77	0.74	0.73
	BDIndicator	0.73	0.72	0.71	0.74	0.71
	FreqFed	0.77	0.81	0.82	0.84	0.80
	FedREDefense	0.82	0.83	0.85	0.83	0.82
	SafeFL-ML	0.84	0.87	0.89	0.87	0.86
	SafeFL-CL	0.94	0.95	0.95	0.97	0.95

(c) Impact of degree of Non-IID.

Attack	Defense	Non-IID degree				
		0.1	0.3	0.5	0.7	0.9
Trim+DBA attack	FLAME	0.74	0.77	0.81	0.83	0.83
	FLDetector	0.82	0.77	0.87	0.91	0.93
	FLTrust	0.84	0.86	0.88	0.85	0.85
	DeepSight	0.70	0.72	0.75	0.75	0.77
	BDIndicator	0.79	0.80	0.78	0.82	0.83
	FreqFed	0.67	0.74	0.80	0.81	0.82
	FedREDefense	0.69	0.63	0.76	0.78	0.80
	SafeFL-ML	0.95	0.90	0.91	0.95	0.97
	SafeFL-CL	0.95	0.95	0.96	0.96	0.99
Scaling+DBA attack	FLAME	0.87	0.85	0.89	0.90	0.90
	FLDetector	0.71	0.68	0.69	0.73	0.74
	FLTrust	0.75	0.76	0.76	0.78	0.79
	DeepSight	0.85	0.85	0.86	0.89	0.90
	BDIndicator	0.89	0.95	0.94	0.95	0.95
	FreqFed	0.83	0.84	0.84	0.87	0.90
	FedREDefense	0.86	0.86	0.85	0.88	0.89
	SafeFL-ML	0.96	0.95	0.95	0.98	1.00
	SafeFL-CL	0.97	0.97	0.98	0.99	1.00
Adaptive attack	FLAME	0.75	0.78	0.77	0.79	0.77
	FLDetector	0.67	0.75	0.85	0.84	0.81
	FLTrust	0.74	0.72	0.70	0.73	0.82
	DeepSight	0.70	0.79	0.77	0.82	0.83
	BDIndicator	0.64	0.68	0.71	0.76	0.79
	FreqFed	0.75	0.79	0.82	0.81	0.83
	FedREDefense	0.80	0.82	0.85	0.87	0.86
	SafeFL-ML	0.82	0.87	0.89	0.90	0.89
	SafeFL-CL	0.93	0.94	0.95	0.92	0.94

Table 11: The impact of the selection rate is analyzed using the CIFAR-10 dataset, where DACC values are reported.

(a) Results under Trim, Scaling, and DBA attacks.

Attack	Defense	Selection rate				
		20%	30%	50%	80%	100%
Trim attack	FLAME	0.75	0.76	0.77	0.76	0.78
	FLDetector	0.75	0.77	0.76	0.75	0.96
	FLTrust	0.93	0.96	0.95	0.94	0.85
	DeepSight	0.84	0.87	0.86	0.86	0.88
	BackdoorIndicator	0.85	0.87	0.89	0.87	0.73
	FreqFed	0.77	0.82	0.80	0.74	0.89
	FedREDefense	0.85	0.87	0.89	0.90	0.85
	SafeFL-ML	0.87	0.82	0.87	0.85	0.90
	SafeFL-CL	0.92	0.94	0.91	0.96	1.00
Scaling attack	FLAME	0.80	0.82	0.81	0.83	0.84
	FLDetector	0.79	0.92	0.84	0.99	1.00
	FLTrust	0.78	0.83	0.72	0.77	0.75
	DeepSight	0.85	0.87	0.84	0.86	0.88
	BackdoorIndicator	0.93	0.91	0.89	0.92	0.94
	FreqFed	0.85	0.83	0.86	0.86	0.84
	FedREDefense	0.85	0.83	0.84	0.88	0.87
	SafeFL-ML	0.92	0.90	0.93	0.94	0.91
	SafeFL-CL	1.00	0.97	0.95	0.96	1.00
DBA attack	FLAME	0.85	0.83	0.84	0.84	0.82
	FLDetector	0.84	0.87	0.87	0.82	0.89
	FLTrust	0.84	0.76	0.78	0.77	0.80
	DeepSight	0.92	0.90	0.89	0.87	0.88
	BackdoorIndicator	1.00	1.00	0.97	0.93	1.00
	FreqFed	0.84	0.87	0.88	0.86	0.89
	FedREDefense	0.80	0.77	0.78	0.78	0.75
	SafeFL-ML	0.95	0.96	0.94	0.97	0.94
	SafeFL-CL	1.00	1.00	0.97	0.98	1.00

(b) Results under Trim+DBA, Scaling+DBA, and Adaptive attacks.

Attack	Defense	Selection rate				
		20%	30%	50%	80%	100%
Trim+DBA attack	FLAME	0.84	0.87	0.83	0.85	0.81
	FLDetector	0.84	0.86	0.85	0.86	0.87
	FLTrust	0.90	0.87	0.91	0.85	0.88
	DeepSight	0.79	0.74	0.78	0.81	0.75
	BackdoorIndicator	0.82	0.78	0.79	0.81	0.78
	FreqFed	0.78	0.83	0.80	0.82	0.80
	FedREDefense	0.79	0.72	0.74	0.77	0.76
	SafeFL-ML	0.90	0.94	0.92	0.93	0.91
	SafeFL-CL	0.95	0.93	0.94	0.94	0.96
Scaling+DBA attack	FLAME	0.86	0.92	0.89	0.91	0.89
	FLDetector	0.74	0.66	0.62	0.73	0.69
	FLTrust	0.77	0.75	0.72	0.74	0.76
	DeepSight	0.83	0.87	0.85	0.86	0.86
	BackdoorIndicator	0.92	0.89	0.94	0.96	0.94
	FreqFed	0.84	0.83	0.85	0.83	0.84
	FedREDefense	0.81	0.86	0.83	0.87	0.85
	SafeFL-ML	0.93	0.94	0.97	0.97	0.95
	SafeFL-CL	0.96	0.95	0.97	0.98	0.98
Adaptive attack	FLAME	0.74	0.77	0.78	0.76	0.77
	FLDetector	0.82	0.84	0.83	0.86	0.85
	FLTrust	0.73	0.71	0.72	0.74	0.70
	DeepSight	0.74	0.76	0.67	0.77	0.77
	BackdoorIndicator	0.74	0.76	0.73	0.73	0.71
	FreqFed	0.80	0.81	0.83	0.82	0.82
	FedREDefense	0.83	0.85	0.85	0.84	0.85
	SafeFL-ML	0.89	0.88	0.87	0.90	0.89
	SafeFL-CL	0.94	0.97	0.96	0.94	0.95

Table 12: Detection results of various methods under advanced untargeted attacks on CIFAR-10, with DACC values reported.

Attack	FLAME	FLDetector	FLTrust	DeepSight	BackdoorIndicator	FreqFed	FedREDefense	SafeFL-ML	SafeFL-CL
Label flipping attack	0.89	0.91	0.82	1.00	0.76	0.79	0.85	0.96	1.00
LIE attack	0.87	0.73	0.79	0.94	0.91	0.85	0.78	0.95	1.00

Table 13: Detection performance of various detection-based methods under Neurotoxin, Irreversible backdoor, and Clean-label backdoor attacks, evaluated using DACC (↑), FPR (↓), and FNR (↓) metrics. Here, ↑ denotes better detection performance with higher values, and ↓ denotes better performance with lower values.

Attack	Defense	CIFAR-10			MNIST			FEMNIST			STL-10			Tiny-ImageNet		
		DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR	DACC	FPR	FNR
Neurotoxin attack	FLAME	0.84	0.27	0.11	0.88	0.25	0.06	0.86	0.17	0.13	0.88	0.20	0.09	0.86	0.15	0.14
	FLDetector	0.83	0.30	0.11	0.78	0.45	0.12	0.88	0.25	0.06	0.91	0.18	0.05	0.79	0.22	0.21
	FLTrust	0.72	0.41	0.22	0.74	0.35	0.22	0.77	0.29	0.20	0.82	0.23	0.16	0.86	0.19	0.12
	DeepSight	0.88	0.29	0.05	0.86	0.19	0.12	0.84	0.25	0.12	0.90	0.07	0.11	0.85	0.22	0.12
	BackdoorIndicator	0.76	0.39	0.18	0.89	0.16	0.09	0.93	0.09	0.06	0.77	0.15	0.26	0.86	0.13	0.14
	FreqFed	0.84	0.22	0.13	0.88	0.15	0.11	0.85	0.14	0.15	0.84	0.23	0.13	0.90	0.23	0.04
	FedREDefense	0.87	0.20	0.10	0.80	0.17	0.21	0.94	0.09	0.05	0.80	0.15	0.22	0.82	0.22	0.16
	SafeFL-ML	0.92	0.15	0.05	0.91	0.09	0.09	0.92	0.07	0.08	0.93	0.11	0.05	0.94	0.03	0.07
	SafeFL-CL	0.96	0.04	0.04	0.98	0.00	0.03	0.97	0.00	0.04	0.98	0.03	0.02	0.95	0.13	0.02
Irreversible backdoor attack	FLAME	0.75	0.23	0.26	0.80	0.33	0.14	0.82	0.28	0.14	0.72	0.24	0.30	0.77	0.29	0.20
	FLDetector	0.68	0.23	0.36	0.77	0.20	0.24	0.74	0.23	0.27	0.81	0.37	0.11	0.76	0.27	0.23
	FLTrust	0.70	0.49	0.22	0.74	0.37	0.21	0.74	0.48	0.17	0.80	0.27	0.17	0.84	0.39	0.06
	DeepSight	0.84	0.13	0.17	0.81	0.09	0.23	0.84	0.23	0.13	0.80	0.17	0.21	0.74	0.15	0.31
	BackdoorIndicator	0.80	0.15	0.22	0.87	0.13	0.13	0.87	0.15	0.12	0.89	0.14	0.10	0.78	0.23	0.22
	FreqFed	0.75	0.11	0.31	0.83	0.28	0.12	0.87	0.15	0.12	0.89	0.17	0.08	0.84	0.20	0.14
	FedREDefense	0.83	0.29	0.12	0.82	0.18	0.18	0.81	0.32	0.13	0.78	0.16	0.25	0.87	0.23	0.09
	SafeFL-ML	0.90	0.04	0.13	0.97	0.09	0.00	0.97	0.10	0.00	1.00	0.00	0.00	0.97	0.03	0.03
	SafeFL-CL	0.98	0.07	0.00	0.96	0.03	0.04	0.94	0.00	0.09	0.98	0.06	0.00	0.99	0.00	0.01
Clean-label backdoor attack	FLAME	0.84	0.13	0.17	0.87	0.22	0.09	0.81	0.15	0.21	0.77	0.17	0.26	0.80	0.14	0.23
	FLDetector	0.80	0.23	0.19	0.83	0.13	0.19	0.87	0.17	0.11	0.84	0.23	0.10	0.82	0.17	0.19
	FLTrust	0.76	0.24	0.24	0.79	0.37	0.14	0.84	0.17	0.16	0.82	0.31	0.12	0.85	0.19	0.13
	DeepSight	0.82	0.24	0.15	0.83	0.17	0.17	0.91	0.15	0.06	0.88	0.29	0.05	0.83	0.24	0.14
	BackdoorIndicator	0.90	0.07	0.11	0.84	0.21	0.14	0.88	0.25	0.06	0.80	0.34	0.14	0.87	0.18	0.11
	FreqFed	0.80	0.21	0.20	0.83	0.17	0.17	0.89	0.14	0.10	0.85	0.17	0.14	0.86	0.15	0.14
	FedREDefense	0.87	0.15	0.12	0.89	0.25	0.05	0.92	0.13	0.06	0.88	0.25	0.06	0.89	0.31	0.02
	SafeFL-ML	0.94	0.05	0.06	0.97	0.00	0.04	1.00	0.00	0.00	0.94	0.06	0.06	0.97	0.00	0.04
	SafeFL-CL	1.00	0.00	0.00	1.00	0.00	0.00	0.97	0.03	0.03	0.96	0.10	0.01	1.00	0.00	0.00

Table 14: Performance of final global models obtained through various detection-based methods under Neurotoxin, Irreversible backdoor, and Clean-label backdoor attacks, where TACC (↑) and ASR (↓) metrics are considered. ↑ denotes better performance with higher values, and ↓ denotes better performance with lower values.

Attack	Defense	CIFAR-10		MNIST		FEMNIST		STL-10		Tiny-ImageNet	
		TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR	TACC	ASR
Neurotoxin attack	FLAME	0.77	0.35	0.93	0.27	0.63	0.32	0.49	0.34	0.48	0.27
	FLDetector	0.74	0.34	0.95	0.48	0.65	0.44	0.50	0.20	0.50	0.35
	FLTrust	0.78	0.57	0.94	0.50	0.67	0.50	0.52	0.23	0.47	0.19
	DeepSight	0.73	0.35	0.94	0.29	0.64	0.54	0.52	0.17	0.51	0.26
	BackdoorIndicator	0.80	0.34	0.98	0.23	0.64	0.12	0.47	0.15	0.50	0.17
	FreqFed	0.79	0.60	0.92	0.39	0.66	0.17	0.51	0.35	0.46	0.07
	FedreDefense	0.80	0.27	0.95	0.17	0.63	0.29	0.51	0.22	0.47	0.38
	SafeFL-ML	0.82	0.04	0.98	0.08	0.67	0.07	0.54	0.06	0.50	0.10
	SafeFL-CL	0.83	0.06	0.97	0.03	0.68	0.11	0.53	0.04	0.53	0.05
Irreversible backdoor attack	FLAME	0.77	0.25	0.95	0.19	0.62	0.40	0.49	0.29	0.48	0.41
	FLDetector	0.80	0.45	0.93	0.14	0.64	0.27	0.51	0.64	0.49	0.50
	FLTrust	0.79	0.28	0.94	0.52	0.59	0.71	0.47	0.39	0.49	0.80
	DeepSight	0.77	0.37	0.96	0.18	0.63	0.30	0.50	0.30	0.51	0.27
	BackdoorIndicator	0.80	0.09	0.97	0.29	0.66	0.18	0.48	0.14	0.51	0.36
	FreqFed	0.78	0.17	0.97	0.27	0.62	0.17	0.48	0.23	0.52	0.29
	FedreDefense	0.79	0.23	0.98	0.27	0.63	0.43	0.50	0.17	0.52	0.37
	SafeFL-ML	0.83	0.07	0.98	0.04	0.65	0.09	0.52	0.06	0.53	0.11
	SafeFL-CL	0.83	0.04	0.97	0.05	0.66	0.06	0.54	0.06	0.54	0.09
Clean-label backdoor attack	FLAME	0.80	0.19	0.94	0.28	0.64	0.29	0.45	0.29	0.48	0.09
	FLDetector	0.78	0.45	0.95	0.16	0.65	0.37	0.49	0.53	0.46	0.24
	FLTrust	0.74	0.48	0.96	0.64	0.66	0.22	0.49	0.67	0.49	0.22
	DeepSight	0.80	0.29	0.97	0.27	0.64	0.18	0.50	0.47	0.50	0.32
	BackdoorIndicator	0.81	0.20	0.96	0.35	0.66	0.28	0.52	0.43	0.49	0.27
	FreqFed	0.78	0.33	0.93	0.29	0.65	0.14	0.49	0.34	0.50	0.23
	FedreDefense	0.79	0.28	0.95	0.32	0.66	0.13	0.50	0.40	0.50	0.19
	SafeFL-ML	0.81	0.07	0.97	0.02	0.67	0.07	0.50	0.13	0.51	0.07
	SafeFL-CL	0.83	0.03	0.98	0.05	0.67	0.09	0.52	0.09	0.50	0.07

Table 15: Detection performance of various detection-based methods is assessed using Precision (↑), Recall (↑), and F1-score (↑).

Attack	Detection	CIFAR-10			MNIST			FEMNIST			STL-10			Tiny-ImageNet		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
No attack	FLAME	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	FLDetector	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	FLTrust	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	DeepSight	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	BackdoorIndicator	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	FreqFed	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	FedREDefense	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	SafeFL-ML	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
	SafeFL-CL	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Trim attack	FLAME	0.95	0.72	0.82	0.95	0.74	0.83	0.92	0.74	0.82	0.95	0.72	0.82	0.90	0.76	0.82
	FLDetector	0.99	0.95	0.97	1.00	0.99	0.99	0.99	0.91	0.95	0.95	0.74	0.83	0.96	0.79	0.87
	FLTrust	0.93	0.85	0.89	0.97	0.81	0.88	0.98	0.74	0.84	0.91	0.86	0.88	0.88	0.94	0.91
	DeepSight	0.94	0.88	0.91	0.98	0.83	0.90	0.94	0.77	0.85	0.98	0.85	0.91	0.91	0.81	0.86
	BackdoorIndicator	0.84	0.76	0.80	0.87	0.69	0.77	0.83	0.78	0.81	0.82	0.79	0.80	0.77	0.75	0.76
	FreqFed	0.98	0.86	0.92	0.97	0.80	0.88	0.98	0.85	0.91	0.98	0.86	0.92	0.96	0.77	0.86
	FedREDefense	0.89	0.90	0.89	0.97	0.92	0.94	0.99	0.92	0.95	1.00	0.97	0.98	1.00	0.99	0.99
	SafeFL-ML	0.99	0.87	0.92	0.95	0.89	0.92	0.99	0.92	0.95	0.99	0.90	0.94	0.98	0.93	0.96
	SafeFL-CL	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.97	1.00	0.96	0.98	1.00	0.98	0.99
Scaling attack	FLAME	0.96	0.80	0.87	0.98	0.84	0.90	0.98	0.71	0.82	1.00	0.71	0.83	0.95	0.85	0.90
	FLDetector	1.00	1.00	1.00	0.96	0.96	0.96	0.93	0.92	0.93	0.90	0.79	0.84	0.91	0.86	0.88
	FLTrust	0.82	0.82	0.82	0.85	0.91	0.88	0.80	1.00	0.89	0.90	0.95	0.92	0.93	0.74	0.82
	DeepSight	0.94	0.88	0.91	0.98	0.83	0.90	0.94	0.77	0.85	0.98	0.85	0.91	0.91	0.81	0.86
	BackdoorIndicator	1.00	0.91	0.95	0.99	0.94	0.96	1.00	1.00	1.00	0.91	0.71	0.80	0.92	0.79	0.85
	FreqFed	0.92	0.84	0.88	0.76	0.69	0.72	0.87	0.67	0.76	0.94	0.76	0.84	0.76	0.82	0.79
	FedREDefense	0.94	0.87	0.91	0.97	0.84	0.90	1.00	0.91	0.95	0.93	0.76	0.84	0.84	0.77	0.81
	SafeFL-ML	0.99	0.88	0.93	1.00	0.96	0.98	1.00	1.00	1.00	1.00	0.96	0.98	0.95	0.95	0.95
	SafeFL-CL	1.00	1.00	1.00	0.97	0.94	0.95	1.00	0.97	0.98	1.00	1.00	1.00	0.96	0.96	0.96
DBA attack	FLAME	0.96	0.77	0.86	0.98	0.83	0.90	0.96	0.80	0.87	0.93	0.92	0.93	0.99	0.82	0.90
	FLDetector	0.93	0.91	0.92	0.95	0.90	0.93	0.98	0.89	0.93	0.94	0.75	0.83	0.95	0.86	0.90
	FLTrust	0.90	0.80	0.85	0.91	0.78	0.84	0.89	0.78	0.83	0.87	0.86	0.86	0.88	0.81	0.85
	DeepSight	1.00	0.83	0.91	0.99	0.87	0.92	0.99	0.93	0.96	0.93	0.86	0.89	0.94	0.87	0.91
	BackdoorIndicator	1.00	1.00	1.00	0.98	0.97	0.98	0.98	0.96	0.97	0.97	0.87	0.92	0.93	0.90	0.91
	FreqFed	0.98	0.86	0.92	0.91	0.76	0.83	1.00	0.87	0.93	1.00	1.00	1.00	0.95	0.83	0.89
	FedREDefense	0.88	0.74	0.80	0.97	0.96	0.96	0.95	0.96	0.96	0.86	0.79	0.82	0.93	0.84	0.88
	SafeFL-ML	0.95	0.96	0.96	1.00	0.99	0.99	0.99	0.96	0.97	1.00	1.00	1.00	1.00	0.97	0.98
	SafeFL-CL	1.00	1.00	1.00	0.99	0.98	0.98	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Trim+DBA attack	FLAME	0.98	0.75	0.85	0.95	0.81	0.87	0.96	0.73	0.83	0.95	0.90	0.93	0.98	0.82	0.89
	FLDetector	0.87	0.96	0.91	0.94	0.93	0.94	0.93	0.91	0.92	0.70	1.00	0.82	0.97	0.79	0.87
	FLTrust	0.93	0.89	0.91	0.93	0.77	0.84	0.83	0.82	0.83	0.88	0.68	0.77	0.90	0.95	0.92
	DeepSight	0.81	0.84	0.83	0.92	0.78	0.85	0.86	0.75	0.80	0.96	0.78	0.86	0.90	0.90	0.90
	BackdoorIndicator	0.84	0.85	0.84	0.98	0.75	0.85	0.90	0.75	0.82	0.90	0.80	0.85	0.93	0.85	0.89
	FreqFed	0.92	0.78	0.85	0.92	0.67	0.77	0.94	0.78	0.85	0.87	0.82	0.84	0.91	0.75	0.82
	FedREDefense	0.87	0.77	0.82	0.94	0.84	0.89	1.00	0.94	0.97	0.87	0.79	0.83	0.94	0.82	0.88
	SafeFL-ML	0.97	0.90	0.93	1.00	0.93	0.96	1.00	1.00	1.00	0.97	0.94	0.95	1.00	1.00	1.00
	SafeFL-CL	1.00	0.94	0.97	1.00	1.00	1.00	1.00	0.97	0.98	1.00	1.00	1.00	1.00	0.99	0.99
Scaling+DBA attack	FLAME	1.00	0.84	0.91	0.96	0.80	0.87	0.98	0.80	0.88	0.98	0.82	0.89	0.95	0.89	0.92
	FLDetector	0.83	0.70	0.76	0.89	0.65	0.75	0.89	0.94	0.92	0.80	0.86	0.83	0.98	0.77	0.86
	FLTrust	0.86	0.78	0.82	0.85	0.88	0.87	0.89	0.74	0.81	0.96	0.92	0.94	0.92	0.76	0.83
	DeepSight	0.97	0.83	0.89	0.95	0.89	0.92	0.93	0.93	0.93	1.00	0.86	0.92	0.94	0.94	0.94
	BackdoorIndicator	0.97	0.94	0.95	1.00	0.99	0.99	0.99	0.96	0.97	0.94	0.84	0.89	0.91	0.77	0.84
	FreqFed	0.94	0.83	0.88	0.91	0.91	0.91	0.97	0.72	0.83	0.93	0.86	0.89	0.92	0.95	0.94
	FedREDefense	0.93	0.85	0.89	1.00	0.96	0.98	1.00	0.93	0.96	0.97	0.94	0.96	0.88	0.81	0.85
	SafeFL-ML	0.98	0.95	0.97	1.00	0.96	0.98	1.00	0.99	0.99	1.00	0.89	0.94	0.93	0.88	0.91
	SafeFL-CL	1.00	0.97	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.96	0.97	1.00	0.99	0.99
Adaptive attack	FLAME	0.85	0.82	0.83	0.90	0.72	0.80	0.87	0.67	0.76	0.91	0.80	0.85	0.87	0.79	0.83
	FLDetector	0.91	0.87	0.89	0.88	0.80	0.84	0.92	0.83	0.87	0.92	0.81	0.86	0.85	0.78	0.81
	FLTrust	0.82	0.73	0.77	0.81	0.74	0.77	0.81	0.84	0.82	0.88	0.81	0.85	0.86	0.70	0.77
	DeepSight	0.87	0.80	0.83	0.87	0.76	0.81	0.86	0.72	0.78	0.86	0.77	0.81	0.91	0.81	0.86
	BackdoorIndicator	0.80	0.78	0.79	0.94	0.90	0.92	0.89	0.81	0.85	0.76	0.77	0.76	0.86	0.77	0.81
	FreqFed	0.88	0.86	0.87	0.90	0.92	0.91	0.94	0.66	0.78	0.88	0.82	0.85	0.86	0.80	0.83
	FedREDefense	0.95	0.83	0.89	1.00	0.93	0.96	0.87	0.83	0.85	0.89	0.73	0.80	0.87	0.67	0.76
	SafeFL-ML	0.92	0.93	0.92	0.97	0.94	0.95	0.99	0.98	0.98	0.96	0.93	0.94	0.95	0.96	0.95
	SafeFL-CL	0.99	0.94	0.96	1.00	0.96	0.98	1.00	0.91	0.95	0.99	0.96	0.97	1.00	0.93	0.96

Table 16: Detection performance of various detection-based methods is assessed using Precision (↑), Recall (↑), and F1-score (↑), under three advanced backdoor attacks.

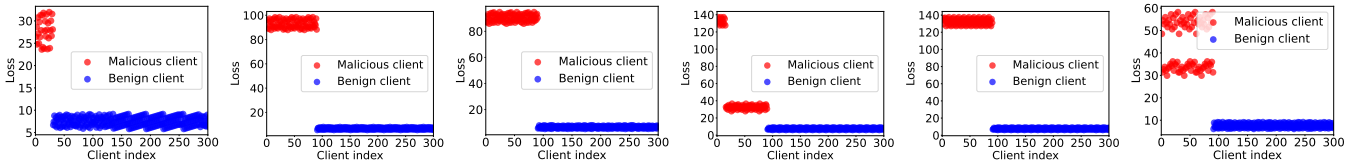
Attack	Defense	CIFAR-10			MNIST			FEMNIST			STL-10			Tiny-ImageNet		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Neurotoxin attack	FLAME	0.86	0.89	0.87	0.89	0.94	0.91	0.90	0.87	0.88	0.91	0.91	0.91	0.92	0.86	0.89
	FLDetector	0.84	0.89	0.86	0.79	0.88	0.83	0.86	0.94	0.90	0.93	0.95	0.94	0.85	0.79	0.82
	FLTrust	0.76	0.78	0.77	0.78	0.78	0.78	0.81	0.80	0.80	0.85	0.84	0.85	0.92	0.88	0.90
	DeepSight	0.88	0.95	0.91	0.90	0.88	0.89	0.87	0.88	0.88	0.94	0.89	0.91	0.89	0.88	0.89
	BackdoorIndicator	0.81	0.82	0.82	0.92	0.91	0.91	0.96	0.94	0.95	0.83	0.74	0.78	0.92	0.86	0.89
	FreqFed	0.88	0.87	0.87	0.91	0.89	0.90	0.89	0.85	0.87	0.86	0.87	0.87	0.93	0.90	0.91
	FedREDefense	0.90	0.90	0.90	0.85	0.80	0.82	0.96	0.95	0.96	0.85	0.78	0.81	0.87	0.84	0.86
	SafeFL-ML	0.95	0.95	0.95	0.96	0.91	0.93	0.97	0.92	0.94	0.96	0.95	0.96	0.98	0.94	0.96
	SafeFL-CL	0.99	0.96	0.98	1.00	0.98	0.99	1.00	0.97	0.98	0.98	0.98	0.98	0.95	0.95	0.95
Irreversible backdoor attack	FLAME	0.83	0.74	0.78	0.85	0.86	0.85	0.87	0.86	0.86	0.79	0.70	0.74	0.83	0.80	0.81
	FLDetector	0.80	0.64	0.71	0.84	0.76	0.80	0.82	0.73	0.77	0.84	0.89	0.86	0.83	0.77	0.80
	FLTrust	0.74	0.78	0.76	0.78	0.79	0.78	0.78	0.83	0.80	0.84	0.83	0.83	0.88	0.94	0.91
	DeepSight	0.91	0.83	0.87	0.92	0.77	0.84	0.87	0.87	0.87	0.88	0.79	0.83	0.83	0.69	0.75
	BackdoorIndicator	0.90	0.78	0.83	0.93	0.87	0.90	0.93	0.88	0.90	0.94	0.90	0.92	0.86	0.78	0.82
	FreqFed	0.89	0.69	0.78	0.88	0.88	0.88	0.93	0.87	0.90	0.93	0.92	0.93	0.90	0.86	0.88
	FedREDefense	0.87	0.88	0.88	0.89	0.82	0.85	0.85	0.87	0.86	0.86	0.75	0.80	0.91	0.91	0.91
	SafeFL-ML	0.97	0.90	0.93	0.97	1.00	0.98	0.97	1.00	0.98	1.00	1.00	1.00	0.98	0.97	0.98
	SafeFL-CL	0.98	1.00	0.99	0.99	0.96	0.97	1.00	0.94	0.97	0.98	1.00	0.99	1.00	0.99	0.99
Clean-label backdoor attack	FLAME	0.91	0.83	0.87	0.90	0.91	0.90	0.88	0.79	0.83	0.85	0.74	0.79	0.88	0.77	0.82
	FLDetector	0.87	0.81	0.84	0.91	0.81	0.86	0.91	0.89	0.90	0.88	0.90	0.89	0.90	0.83	0.86
	FLTrust	0.85	0.76	0.80	0.83	0.86	0.84	0.90	0.84	0.87	0.86	0.88	0.87	0.91	0.87	0.89
	DeepSight	0.87	0.85	0.86	0.89	0.83	0.86	0.94	0.94	0.94	0.90	0.95	0.92	0.88	0.86	0.87
	BackdoorIndicator	0.96	0.90	0.93	0.89	0.86	0.87	0.88	0.94	0.91	0.83	0.86	0.84	0.93	0.89	0.91
	FreqFed	0.88	0.80	0.84	0.89	0.83	0.86	0.93	0.89	0.91	0.91	0.86	0.88	0.92	0.86	0.89
	FedREDefense	0.93	0.88	0.90	0.91	0.95	0.93	0.95	0.94	0.95	0.91	0.94	0.93	0.90	0.98	0.94
	SafeFL-ML	0.98	0.94	0.96	1.00	0.97	0.98	1.00	1.00	1.00	0.96	0.94	0.95	1.00	0.97	0.98
	SafeFL-CL	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.98	0.96	0.99	0.98	1.00	1.00	1.00

Table 17: Storage costs of different methods.

Defense	CIFAR-10	MNIST	FEMNIST	STL-10	Tiny-ImageNet
FLDetector	21.73 GB	15.76 GB	22.95 GB	38.46 GB	31.28 GB
FedREDefense	19.53 GB	10.72 GB	20.15 GB	28.83 GB	27.46 GB
SafeFL-ML	13.66 GB	7.96 GB	13.57 GB	22.40 GB	19.93 GB
SafeFL-CL	13.66 GB	7.96 GB	13.57 GB	22.40 GB	19.93 GB

Table 18: Equalized odds and Demographic parity scores of our methods under different attacks, where the CIFAR-10 dataset is considered.

Attack	Defense	Setting I		Setting II	
		Equalized odds	Demographic parity	Equalized odds	Demographic parity
No attack	FedAvg	0.79	0.57	0.69	0.48
	SafeFL-ML	0.77	0.59	0.66	0.47
	SafeFL-CL	0.78	0.56	0.71	0.52
Trim attack	SafeFL-ML	0.81	0.58	0.71	0.49
	SafeFL-CL	0.77	0.57	0.70	0.49
Scaling attack	SafeFL-ML	0.77	0.56	0.70	0.50
	SafeFL-CL	0.79	0.56	0.69	0.48
DBA attack	SafeFL-ML	0.78	0.57	0.70	0.48
	SafeFL-CL	0.76	0.55	0.64	0.45
Trim+DBA attack	SafeFL-ML	0.82	0.58	0.70	0.49
	SafeFL-CL	0.77	0.55	0.69	0.51
Scaling + DBA attack	SafeFL-ML	0.80	0.59	0.72	0.50
	SafeFL-CL	0.77	0.55	0.68	0.49
Adaptive attack	SafeFL-ML	0.78	0.55	0.66	0.47
	SafeFL-CL	0.80	0.61	0.68	0.50

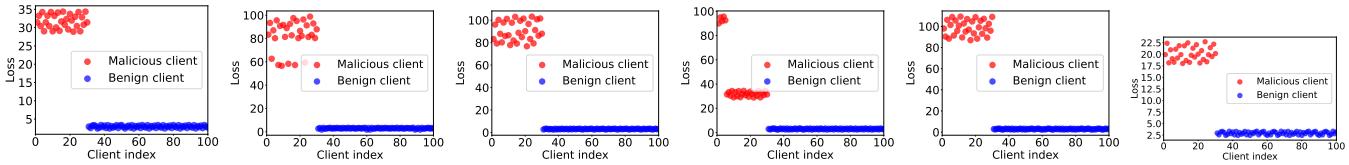


(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 5: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-ML with the FEMNIST dataset.

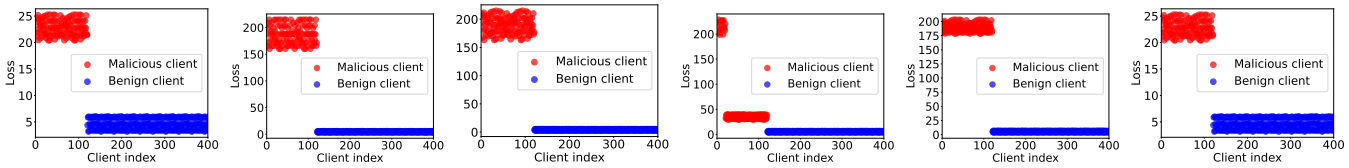
Table 19: The impact of the noise level is analyzed using the CIFAR-10 dataset, where DACC and TACC values are reported.

Attack	Defense	Noise level									
		0		0.5		1		1.5		2	
		DACC	TACC	DACC	TACC	DACC	TACC	DACC	TACC	DACC	TACC
No attack	FedAvg	NA	0.85	NA	0.81	NA	0.80	NA	0.76	NA	0.70
	FLAME	0.57	0.72	0.55	0.71	0.56	0.70	0.52	0.66	0.58	0.62
	FLDetector	0.65	0.77	0.66	0.74	0.64	0.73	0.62	0.66	0.59	0.63
	FLTrust	0.82	0.81	0.77	0.74	0.75	0.72	0.70	0.67	0.70	0.65
	DeepSight	1.00	0.75	0.94	0.74	0.86	0.70	0.83	0.67	0.81	0.66
	BackdoorIndicator	0.95	0.81	0.93	0.81	0.83	0.75	0.77	0.74	0.73	0.67
	FreqFed	0.81	0.79	0.83	0.77	0.77	0.72	0.69	0.70	0.69	0.64
	FedREDefense	0.78	0.81	0.74	0.80	0.72	0.77	0.70	0.67	0.70	0.65
	SafeFL-ML	0.94	0.82	0.92	0.80	0.92	0.78	0.90	0.73	0.88	0.67
	SafeFL-CL	1.00	0.84	0.98	0.83	0.95	0.79	0.94	0.75	0.89	0.69
Trim attack	FLAME	0.78	0.77	0.75	0.76	0.70	0.72	0.67	0.66	0.61	0.58
	FLDetector	0.96	0.79	0.92	0.77	0.89	0.71	0.84	0.67	0.75	0.55
	FLTrust	0.85	0.74	0.82	0.74	0.81	0.70	0.74	0.64	0.70	0.57
	DeepSight	0.88	0.74	0.86	0.77	0.85	0.73	0.81	0.62	0.77	0.49
	BackdoorIndicator	0.73	0.62	0.70	0.57	0.67	0.53	0.66	0.47	0.64	0.47
	FreqFed	0.89	0.79	0.83	0.79	0.80	0.67	0.75	0.60	0.70	0.55
	FedREDefense	0.85	0.81	0.83	0.78	0.81	0.62	0.74	0.59	0.69	0.51
	SafeFL-ML	0.90	0.82	0.87	0.80	0.87	0.79	0.86	0.72	0.84	0.68
	SafeFL-CL	1.00	0.84	0.93	0.79	0.93	0.77	0.89	0.74	0.86	0.67
Scaling attack	FLAME	0.84	0.79	0.83	0.79	0.83	0.77	0.78	0.74	0.72	0.67
	FLDetector	1.00	0.79	0.94	0.78	0.92	0.75	0.90	0.74	0.88	0.69
	FLTrust	0.75	0.62	0.72	0.61	0.70	0.64	0.64	0.63	0.63	0.62
	DeepSight	0.88	0.76	0.87	0.76	0.85	0.72	0.82	0.70	0.82	0.69
	BackdoorIndicator	0.94	0.80	0.94	0.78	0.92	0.74	0.88	0.72	0.86	0.67
	FreqFed	0.84	0.69	0.82	0.69	0.81	0.69	0.77	0.67	0.74	0.64
	FedREDefense	0.87	0.77	0.86	0.75	0.83	0.72	0.82	0.70	0.79	0.66
	SafeFL-ML	0.91	0.81	0.89	0.80	0.88	0.78	0.87	0.75	0.85	0.69
	SafeFL-CL	1.00	0.79	0.97	0.78	0.94	0.77	0.93	0.76	0.91	0.70
DBA attack	FLAME	0.82	0.78	0.80	0.77	0.77	0.74	0.77	0.72	0.71	0.67
	FLDetector	0.89	0.77	0.85	0.77	0.83	0.72	0.79	0.69	0.72	0.65
	FLTrust	0.80	0.78	0.77	0.76	0.76	0.73	0.74	0.70	0.72	0.64
	DeepSight	0.88	0.80	0.84	0.78	0.82	0.74	0.77	0.73	0.71	0.67
	BackdoorIndicator	1.00	0.80	0.93	0.77	0.88	0.75	0.82	0.70	0.75	0.65
	FreqFed	0.89	0.78	0.83	0.74	0.80	0.74	0.75	0.70	0.71	0.68
	FedREDefense	0.75	0.70	0.72	0.69	0.70	0.69	0.69	0.62	0.64	0.60
	SafeFL-ML	0.94	0.78	0.89	0.78	0.88	0.77	0.85	0.75	0.81	0.70
	SafeFL-CL	1.00	0.82	0.94	0.81	0.88	0.77	0.87	0.75	0.87	0.71
Trim+DBA attack	FLAME	0.81	0.79	0.79	0.76	0.77	0.67	0.76	0.62	0.69	0.53
	FLDetector	0.87	0.65	0.83	0.63	0.79	0.63	0.74	0.57	0.73	0.56
	FLTrust	0.88	0.77	0.87	0.76	0.80	0.75	0.77	0.67	0.74	0.61
	DeepSight	0.75	0.49	0.74	0.47	0.72	0.46	0.67	0.47	0.63	0.44
	BackdoorIndicator	0.78	0.60	0.75	0.62	0.74	0.57	0.70	0.55	0.67	0.55
	FreqFed	0.80	0.76	0.78	0.75	0.74	0.72	0.70	0.66	0.68	0.58
	FedREDefense	0.76	0.59	0.74	0.58	0.70	0.52	0.65	0.52	0.62	0.49
	SafeFL-ML	0.91	0.79	0.88	0.78	0.88	0.72	0.87	0.74	0.86	0.69
	SafeFL-CL	0.96	0.83	0.92	0.80	0.89	0.77	0.87	0.74	0.81	0.71
Scaling+DBA attack	FLAME	0.89	0.80	0.87	0.79	0.82	0.77	0.77	0.72	0.69	0.70
	FLDetector	0.69	0.54	0.69	0.53	0.67	0.50	0.65	0.49	0.63	0.48
	FLTrust	0.76	0.49	0.75	0.47	0.73	0.47	0.70	0.44	0.69	0.44
	DeepSight	0.86	0.80	0.82	0.78	0.78	0.67	0.75	0.64	0.71	0.59
	BackdoorIndicator	0.94	0.80	0.90	0.77	0.88	0.70	0.78	0.67	0.74	0.62
	FreqFed	0.84	0.78	0.83	0.75	0.74	0.71	0.70	0.65	0.69	0.62
	FedREDefense	0.85	0.72	0.83	0.71	0.79	0.69	0.77	0.67	0.72	0.60
	SafeFL-ML	0.95	0.80	0.93	0.77	0.92	0.76	0.89	0.74	0.84	0.69
	SafeFL-CL	0.98	0.83	0.95	0.80	0.90	0.80	0.88	0.76	0.84	0.70
Adaptive attack	FLAME	0.77	0.59	0.74	0.58	0.70	0.52	0.69	0.47	0.67	0.35
	FLDetector	0.85	0.65	0.82	0.60	0.82	0.60	0.78	0.54	0.78	0.51
	FLTrust	0.70	0.62	0.69	0.59	0.67	0.52	0.64	0.52	0.61	0.43
	DeepSight	0.77	0.62	0.76	0.60	0.72	0.54	0.67	0.50	0.63	0.44
	BackdoorIndicator	0.71	0.48	0.69	0.48	0.65	0.44	0.65	0.42	0.62	0.38
	FreqFed	0.82	0.79	0.80	0.72	0.77	0.68	0.76	0.62	0.72	0.60
	FedREDefense	0.85	0.78	0.83	0.77	0.80	0.77	0.77	0.71	0.73	0.64
	SafeFL-ML	0.89	0.78	0.87	0.77	0.86	0.76	0.85	0.74	0.83	0.69
	SafeFL-CL	0.95	0.80	0.92	0.79	0.89	0.78	0.88	0.75	0.84	0.69



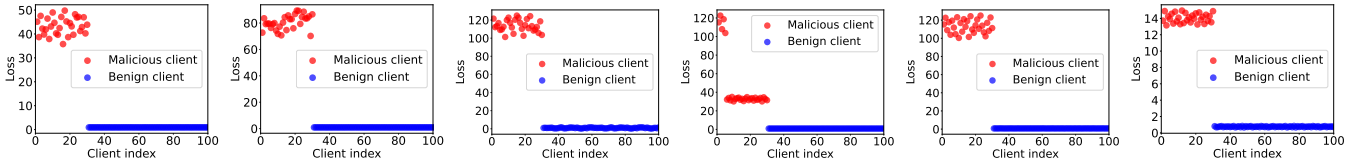
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 6: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-ML with the STL-10 dataset.



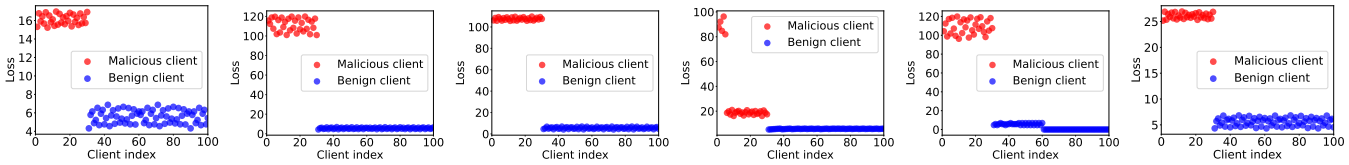
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 7: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-ML with the Tiny-ImageNet dataset.



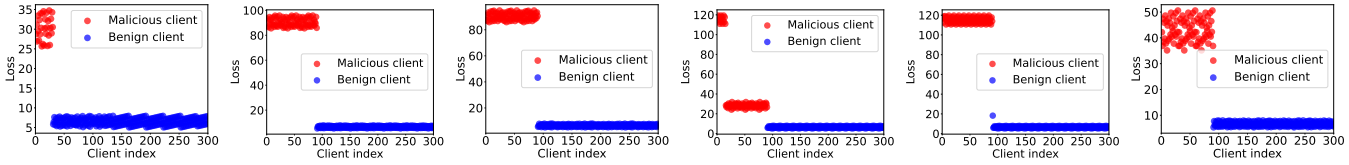
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 8: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-CL with the CIFAR-10 dataset.



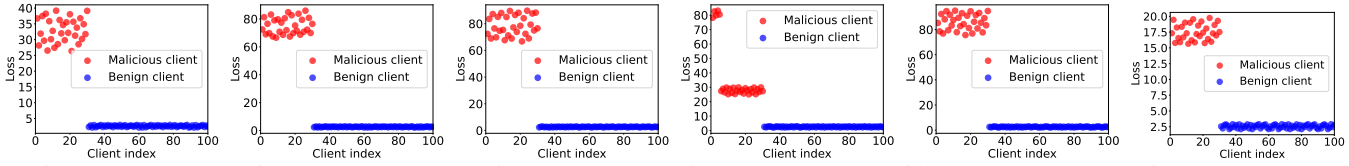
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 9: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-CL with the MNIST dataset.



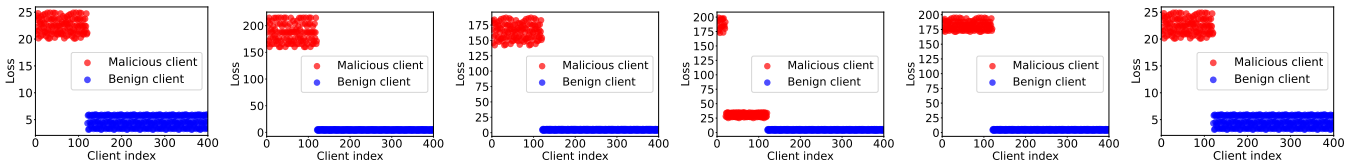
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 10: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-CL with the FEMNIST dataset.



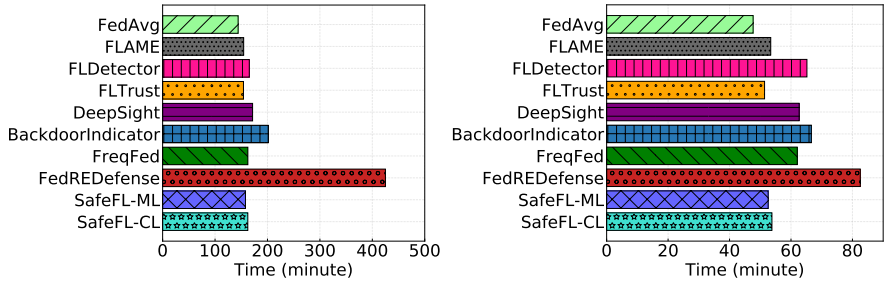
(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 11: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-CL with the STL-10 dataset.

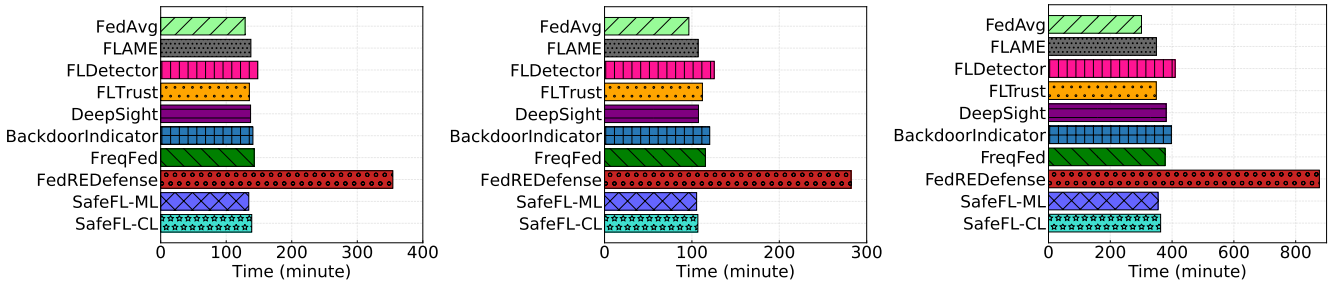


(a) Trim attack (b) Scaling attack (c) DBA attack (d) Trim+DBA attack (e) Scaling+DBA attack (f) Adaptive attack

Figure 12: The loss values of benign and malicious clients' local models computed on the synthetic dataset, using SafeFL-CL with the Tiny-ImageNet dataset.



(a) CIFAR-10 (b) MNIST



(c) FEMNIST (d) STL-10 (e) Tiny-ImageNet

Figure 13: Computation costs of different methods.