FEDROLA: Robust Federated Learning Against Model Poisoning via Layer-based Aggregation

Gang Yan
Binghamton University
Binghamton, NY, USA
gyan2@binghamton.edu

Hao Wang
Stevens Institute of Technology
Hoboken, NJ, USA
hwang9@stevens.edu

Xu Yuan
University of Delaware
Newark, DE, USA
xyuan@udel.edu

Jian Li
Stony Brook University
Stony Brook, NY, USA
jian.li.3@stonybrook.edu

ABSTRACT

Federated Learning (FL) is increasingly vulnerable to model poisoning attacks, where malicious clients degrade the global model's accuracy with manipulated updates. Unfortunately, most existing defenses struggle to handle the scenarios when multiple adversaries exist, and often rely on historical or validation data, rendering them ill-suited for the dynamic and diverse nature of real-world FL environments. Exacerbating these limitations is the fact that most existing defenses also fail to account for the distinctive contributions of Deep Neural Network (DNN) layers in detecting malicious activity, leading to the unnecessary rejection of benign updates. To bridge these gaps, we introduce FEDROLA, a cutting-edge similarity-based defense method optimized for FL. Specifically, FEDROLA leverages global model parameters and client updates independently, moving away from reliance on historical or validation data. It features a unique layer-based aggregation with dynamic layer selection, enhancing threat detection, and includes a dynamic probability method for balanced security and model performance. Through comprehensive evaluations using different DNN models and real-world datasets, FEDROLA demonstrates substantial improvements over the status quo approaches in global model accuracy, achieving up to 4% enhancement in terms of accuracy, reducing false positives to 6.4%, and securing an 92.8% true positive rate.

CCS CONCEPTS

• **Security and privacy** → *Trust frameworks*; *Domain-specific security and privacy architectures*; • **Computing methodologies** → **Distributed algorithms**; **Classification and regression trees**.

KEYWORDS

Robust Federated Learning, Defense, Model Poisoning Attacks, Cosine Similarity, Layer-based Algorithm

ACM Reference Format:
Gang Yan, Hao Wang, Xu Yuan, and Jian Li. 2024. FEDROLA: Robust Federated Learning Against Model Poisoning via Layer-based Aggregation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, Barcelona, Spain, 12 pages. https://doi.org/10.1145/3637528.3671906

1 INTRODUCTION

Federated Learning (FL) [15, 25, 34, 42, 44] represents a groundbreaking advancement in distributed learning. It allows numerous clients to collaboratively train a Deep Neural Network (DNN), referred to as the global model. A key benefit of FL is the preservation of data privacy, as individual datasets remain with the clients. This aspect is especially crucial in today's context, where privacy concerns are heightened due to emerging regulations like the General Data Protection Regulation (GDPR) [36]. The FL process involves a central server that aggregates local updates from clients using a defined rule, which progressively refines the global model throughout the FL training cycles.

Despite its innovative approach, FL is susceptible to a range of model poisoning attacks. These attacks fall into two main categories: targeted [2, 4, 35, 47], which aim to compromise the model's performance on specific test inputs, and untargeted [3, 5, 12, 13, 24, 31, 32, 41, 43], which seek to broadly degrade the overall accuracy of the global model. The untargeted attacks are particularly damaging as they diminish the model's accuracy across all test inputs. Advanced instances of untargeted attacks, such as Fang [13] and Min-Max/Min-Sum [31], have proven capable of circumventing current Byzantine-robust aggregation techniques like Krum [5] and Trimmed-mean [40, 45]. This situation underscores the urgent need for developing more resilient defense mechanisms in FL to counter these evolving threats effectively.

Current defense strategies primarily revolve around Byzantine-robust aggregation rules. While these strategies offer some degree of effectiveness, they are hindered by several notable limitations [5, 6, 11, 40, 43, 45, 46]. One of the key limitations is their diminished efficacy in scenarios with a larger number of malicious clients, which often the case in practice. Additionally, existing defense mechanisms often rely on historical update data for detecting anomalies, or necessitating access to representative validation datasets that closely align with the overall training data distribution. This requirement poses a significant challenge, particularly in diverse and evolving data environments since such clean validation datasets are often not available. Another critical issue in existing defenses is the high false positive rate generated, which can lead to the unwarranted exclusion of benign clients. Furthermore, current strategies tend to overlook the nuanced importance of different DNN layers in FL, raising questions about their reliability and comprehensive effectiveness, especially those with substantial impact [4, 13].

In response to the urgent need for more adaptable and efficient defense mechanisms in FL, we develop FedRoLa (Robust Layer-based Aggregation for FL), a novel similarity-based defense method that harnesses the structural nuances of DNNs leveraged in FL training. FedRoLa is distinct in its approach, as it operates without the need for representative benign data at the server, leveraging global model parameters and client updates for analysis. This design enables it to function effectively regardless of the number of malicious clients in a given communication round and eliminates reliance on historical client data. Comprehensive experimental evaluations show that FedRoLa markedly improves global model accuracy by up to 4%, surpassing existing defenses in the literature. In addition, FedRoLa substantially reduces the false positive rate to about 6.4%, a significant improvement from the 33.4% average in contemporary methods, and achieves a true positive rate of 92.8%, considerably higher than the average 67.3% found in current defense methods. These enhancements position FedRoLa as an efficient defense mechanism against advanced FL security threats.

In summary, this work makes the following contributions.

- We propose FedRoLa, which incorporates a layer-based aggregation that utilizes similarity metrics, coupled with dynamic layer selection. This novel approach facilitates more precise threat detection and significantly reduces false positive rates.
- We develop a dynamic probability method within FedRoLa to maintain an optimal balance between the model's performance and its security. This feature is particularly effective in scenarios with minimal malicious activities without compromising the model's efficiency.
- We conduct extensive evaluations, including AlexNet on Fashion MNIST, VGG-11 on CIFAR-10, ResNet-18 on CIFAR-100, LSTM on the Shakespeare dataset, and DNN on the HARBOx dataset. Our comparative analysis against seven status quo defenses, including recent advancements like FLAIR [1], cosDefense [11], and FLTrust [6], demonstrates that FedRoLa consistently delivers robust performance across different settings and scenarios.

The remainder of this paper provides a detailed exploration of FedRoLa's framework and effectiveness. In Section 2, we outline the state-of-the-art defense and attacks, and highlight the vulnerabilities in current defense methods. Section 3 is dedicated to presenting the design and architecture of FedRoLa. Section 4 presents the comprehensive evaluations and discussions on the performance FedRoLa. Additional experimental details and results are provided in the appendix.

2 BACKGROUND AND RELATED WORK

This section provides an overview of pertinent defense methods and fundamental concepts in federated learning.

2.1 Federated Learning

In Federated Learning (FL), a set of clients, denoted as $\mathcal{N} = \{1, \cdots, N\}$, collaborates to train a model using decentralized data, guided by a central server. The primary objective of FL is to address the following optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{L}(\mathbf{w}, \mathcal{D}) := \sum_{i\in\mathcal{N}} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot \mathcal{L}_i(\mathbf{w}, \mathcal{D}_i), \tag{1}$$

where $\mathbf{w}$ denotes the model parameters, $\mathcal{D}_i$ is the local dataset of client $i \in \mathcal{N}$, the entire training dataset is $\mathcal{D} = \cup_{i\in\mathcal{N}}\mathcal{D}_i$, and $\mathcal{L}_i(\mathbf{w}, \mathcal{D}_i)$ is the local loss function of client i. The first state-of-the-art method to this optimization problem is *FedAvg* [25], which initializes with a random model $\mathbf{w}^{(0)}$ and iterates the following steps between clients within each communication round t:

- **Local training.** The central server sends the goal model $\mathbf{w}^{(t-1)}$ to a randomly selected subset of clients $\mathcal{N}^{(t)} \subset \mathcal{N}$. Each client $i \in \mathcal{N}^{(t)}$ performs local training using $\mathcal{D}_i$:

$$\mathbf{w}_i^{(t)}(k) \leftarrow \mathbf{w}_i^{(t)}(k-1) - \eta\nabla\mathcal{L}_i(\mathbf{w}_i^{(t)}(k-1), \mathcal{D}_i), \tag{2}$$

  where η is the learning rate, k is the index of local iterations and initialize $\mathbf{w}_i^{(t)}(0) = \mathbf{w}^{(t-1)}$.
- **Global aggregation.** The central server obtains a new global model $\mathbf{w}^{(t)}$ by weighted-averaging the local models collected from the selected clients in round t:

$$\mathbf{w}^{(t)} \leftarrow \sum_{i\in\mathcal{N}^{(t)}} \frac{|\mathcal{D}_i|}{|\cup_{i\in\mathcal{N}^{(t)}} \mathcal{D}_i|}\mathbf{w}_i^{(t)}. \tag{3}$$

FedAvg is a fundamental aggregation method in FL, yet it lacks robustness against attacks. To address this, various Byzantine-robust aggregation methods have been developed, each offering unique strengths to safeguard FL's integrity against adversarial threats. For example, FLDetector [46] focuses on identifying malicious clients through the alignment of their updates with server predictions based on historical data. However, its reliance on extensive historical records limits its practicality in scenarios with infrequent malicious activities. FLTrust [6] uses a validation dataset to assign trust scores to client updates, facing challenges in environments where accessing an untainted validation dataset is problematic. Its effectiveness is also reduced by malicious updates that mimic benign inputs, a common tactic like Min-Max [31]. Similarly, AFA [27] and Multi-krum [5] struggle with high false discovery rates. The Trimmed-mean method [40, 45] aggregates gradients but often fails to effectively distinguish between benign and malicious updates. The more recent cosDefense [11] and FLAIR [1] methods show advancements in detecting malicious updates, yet they too face challenges with high false positive rates and reliance on predefined thresholds of malicious client participation, respectively.

To address these issues, in this paper, we introduce an adaptive, layer-specific defense method named FedRoLa, which is partially inspired by observations in [11, 27]. From a high-level perspective, our FedRoLa utilizes a granular analysis of each layer in a DNN utilized for FL training, and employs cosine similarity to enhance the detection of malicious clients. FedRoLa aims to strike a balance between accurately detecting attacks and minimizing false positives, and therefore addresses the common trade-offs in current models between security and model efficacy.

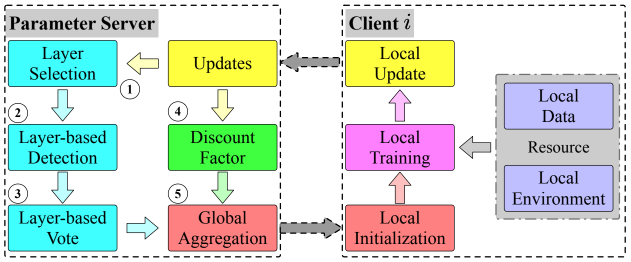

Figure 1: Design of FedRoLa Framework.

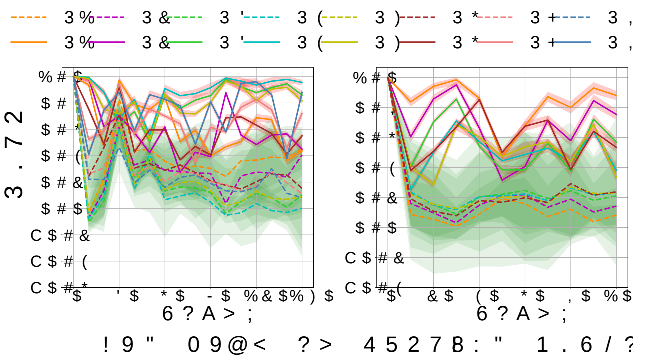

Figure 2: Layer Alignment Similarity Index (*LASI*). Dashed lines depict the similarity indices for benign updates, whereas solid lines indicate those for malicious updates.

2.2 Model Poisoning Attacks

Security threats manifest through diverse poisoning attacks can be categorized by their goals as either untargeted [3, 5, 12, 13, 24, 31, 32, 41, 43], which aim for broad degradation of the global model's accuracy, or targeted [2, 4, 35, 47], such as backdoor attacks [2, 47], which compromise accuracy on specific inputs while maintaining overall high accuracy of other inputs.

These attacks are further classified by capability into data poisoning [16, 26] and model poisoning [2–4, 12, 13, 32, 39, 41, 43]. Data poisoning, less effective due to its indirect gradient manipulation via tainted training datasets, has been seen in diverse machine learning scenarios. Model poisoning, more impactful, directly alters gradients on malicious devices. Recent studies [4, 13] highlight the greater threat of local model poisoning compared to data poisoning, leading to our focus on below attack methods as baselines:

- **Fang** [13]: An optimization-based approach manipulating gradient directions by solving for a global coefficient λ.
- **LIE** [3]: Adds noise to benign gradient averages to subtly impact the model while avoiding detection, based on a calculated coefficient.
- **Min-Sum** [31]: Limits the sum of squared distances between malicious and benign gradients, ensuring it does not exceed the maximum squared distance of benign gradients.
- **Min-Max** [31]: Limits the maximum distance of malicious gradients, camouflaging them among benign ones.
- **MPHM** [32]: Utilizes momentum from historical training data to craft stealthy, disruptive updates.

These methods can be used to undermine FL, either through direct gradient manipulation or subtle gradient alterations.

3 DESIGN OF FEDROLA

3.1 Overview

In this work, we propose FedRoLa, a cutting-edge defense method for FL, which employs a similarity analysis based on current round data to detect malicious clients and bolster the global model's resilience. Diverging from most existing defense methods, FedRoLa features an adaptive layer-based detection system, meticulously designed to lower false positives and increase true positives, thus offering a more precise and effective defense approach.

Figure 1 illustrates the architecture of our proposed FedRoLa framework, which commences with individual client nodes executing local computations to generate updates. These updates are then rigorously vetted through a series of strategic steps within the parameter server framework, forming the backbone of our comprehensive defense mechanism:

① **Layer Selection**. This step selects critical model layers for in-depth analysis, based on their potential for revealing anomalies indicative of malicious behavior.
② **Layer-based Detection**. Each selected layer undergoes specific detection to spot unusual patterns in local updates, suggesting potential maliciousness. We introduce two new similarity-based algorithms to achieve this goal.
③ **Layer-based Vote**. Post-detection, a collective voting mechanism assesses each layer's findings, to accurately pinpoint potential threats and thereby reducing false positive rates.
④ **Discount Factor**. This factor, integral to the aggregation process, adapts according to the current round. It applies reduced weights to updates flagged as potentially malicious, with its value converging towards 1 over time from 0.
⑤ **Global Aggregation**. The final phase involves aggregating updates for the global model, incorporating insights from the detection phase and discount factor.

Our advocated *Layer-based Detection* module is pivotal in the design of FedRoLa, serving as the cornerstone of its effectiveness. The index value, central to this module, is instrumental in executing accurate detection. We delve into this design and assess its efficiency through extensive experiments in the subsequent section.

3.2 Similarity-based Index Value

Building on the insights from previous research [13, 31] that highlight the need for a certain degree of similarity in malicious updates to significantly impact FL, we propose a novel approach inspired by [11, 27]. This method focuses on detecting malicious clients through a layer-based analysis of neural network updates, diverging from traditional methods that analyze the entire network collectively.

Layer Alignment Similarity Index (*LASI*). We define the set of local updates from clients in any given round t as $\{\mathrm{g}_i^{(t)}\}_{i\in\mathcal{N}^{(t)}}$, with $\mathcal{N}^{(t)}$ representing the participants. We calculate the estimated global update at round t, denoted as $\tilde{\mathrm{g}}^{(t)}$, by averaging these updates. To identify anomalies and potential malicious behaviors at a more granular level within the neural network, we define the index value $LASI_l$ for each layer $l \in [L]$, which is based on the similarity between local and global updates on layer l. Specifically, the index

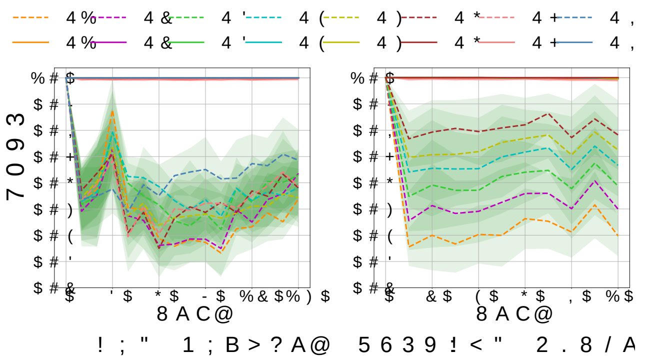

Figure 3: Peer Consensus Similarity Index (*PCSI*). Dashed lines depict the similarity indices for benign updates, whereas solid lines indicate those for malicious updates.

value is defined as

$$LASI_{i,l} := \frac{\langle \mathbf{g}_{i,l}^{(t)}, \tilde{\mathbf{g}}_l^{(t)} \rangle}{||\mathbf{g}_{i,l}^{(t)}|| \cdot ||\tilde{\mathbf{g}}_l^{(t)}||}. \tag{4}$$

Leveraging the layer-specific similarity calculation in (4), our goal is to effectively distinguish benign updates from malicious ones within each layer of a DNN model. To evaluate this approach, we leverage two distinct datasets: Fashion MNIST [21] for image classification and HARBox [8] for human activity recognition. The experiments, designed to detect Min-Max attacks [31], also incorporate the Multi-krum [5], a Byzantine-robust aggregation rule, to test the layer-specific detection in adversarial settings. The experimental setup involves applying the detection method to each individual model layer, starting with the first layer (denoted as $L1$). The distinct neural networks are used for Fashion MNIST and HARBox, due to their different input features and complexities. This approach ensures a thorough and fair evaluation of the method's performance across various data types and model architectures.

The results shown in Figure 2 indicate that specific layers within a neural network are more effective at distinguishing between malicious and benign updates, although this effectiveness varies between different datasets. Identifying these key layers is vital for enhancing the precision of our detection methodology. However, the observations reveal that overlaps in similarity values between benign and malicious updates can potentially lead to errors.

Peer Consensus Similarity Index (*PCSI*). To address challenges in *LASI*, we propose a refined approach for calculating similarity values, aimed at reducing such errors. Specifically, we calculate the similarities among all clients for each layer $l \in [L]$. Once these similarity values are obtained, we select the top three values for each client and compute their average. This average then serves as the index value for client i on layer l:

$$PCSI_{i,l} := \text{Top-K}\left\{ \frac{\langle \mathbf{g}_{i,l}^{(t)}, \mathbf{g}_{j,l}^{(t)} \rangle}{||\mathbf{g}_{i,l}^{(t)}|| \cdot ||\mathbf{g}_{j,l}^{(t)}||} \right\}_{j \in \mathcal{N}^{(t)}, j \neq i}. \tag{5}$$

To evaluate the effectiveness of the proposed approach *PCSI* in (5), we conduct experiments under the same settings. From Figure 3, we observe that most layers within the neural network are capable of distinguishing between malicious and benign updates. However,

Algorithm 1 FedRoLa Algorithm

```
Input: Initial global model $\mathbf{w}_0$
1: Initialize $\alpha_l(0), \beta_l(0) = 1, \forall l \in [L]$; $\alpha_i(0), \beta_i(0) = 1, \forall i \in [N]$
2: for $t = 1, \cdots, T$ do
3:   Based on $\mathbf{w}^{(t-1)}$, perform local model training, and get local update as $\mathbf{g}_i^{(t)} = \mathbf{w}_i^{(t)} - \mathbf{w}_i^{(t-1)}, \forall i \in \mathcal{N}^{(t)}$
4:   Select $\lceil \frac{L}{2} \rceil$ layers denoted as $\mathcal{S}_L$ based on the probability defined as $p_l(t) = \frac{\alpha_l(t)}{\alpha_l(t)+\beta_l(t)}$, and let $\mathcal{B}'_t = \emptyset$
5:   Based on (4) or (5), calculate similarity-based index values $\{I_{i,l}\}_{i \in \mathcal{N}^{(t)}, l \in \mathcal{S}_L}$, where $I \in \{LASI, PCSI\}$
6:   for $l \in \mathcal{S}_L$ do
7:     Implement Algorithm 2, and get layer-based detection results $\mathcal{B}_{t,l}$ for layer $l$
8:     if $\mathcal{B}_{t,l} = \emptyset$ then
9:       Update $\beta_l = \beta_l + 1$
10:    else
11:      Update $\alpha_l = \alpha_l + 1$, and $\mathcal{B}'_t = \mathcal{B}'_t \bigoplus \mathcal{B}_{t,l}$
12:    end if
13:  end for
14:  Implement Algorithm 3 based on $\mathcal{B}'_t$, and get the final detected malicious clients $\mathcal{B}_t$
15:  for $i \in \mathcal{N}^{(t)}$ do
16:    if $i \in \mathcal{B}_t$ then
17:      Update $\beta_i(t) = \beta_i(t) + 1$
18:      Let $D_i^{fac} = \frac{2}{1+e^{-\delta \times t}} - 1$
19:    else
20:      Update $\alpha_i(t) = \alpha_i(t) + 1$
21:      Let $D_i^{fac} = 1$
22:    end if
23:  end for
24:  Use discount factor and $p_i(t) = \frac{\alpha_i(t)}{\alpha_i(t)+\beta_i(t)}$ to update
         $\mathbf{w}^{(t)} \leftarrow \sum_{i \in \mathcal{N}^{(t)}} \frac{|\mathcal{D}_i| \cdot p_i(t) \cdot D_i^{fac}}{\sum_{j \in \mathcal{N}^{(t)}} |\mathcal{D}_j| \cdot p_j(t) \cdot D_j^{fac}} \cdot \mathbf{w}_i^{(t)}$.   (6)
25: end for
```

it is crucial to recognize the computational complexities involved. The *PCSI* exhibits quadratic complexity, $O(n^2)$, compared to the linear complexity, $O(n)$, of *LASI*, with $n = |\mathcal{N}^{(t)}|$ representing the number of clients in round t. Simplifying notation, $I_{i,l}$ is used to denote the similarity index value for client i on layer l, obtained via either method. This index is crucial for anomaly detection, highlighting the need to balance computational load with detection accuracy in real-world scenarios.

An important aspect of the analysis focuses on whether these trends are consistent across varying data distributions, particularly considering the non-IID nature common in FL. We extend these analyses to include scenarios ranging from extremely non-IID data to IID data. These observations, as illustrated in Figures 10 and 11 in the Appendix A.2, demonstrate that specific layers continue to effectively identify malicious clients across these diverse data distributions. This consistency reinforces the robustness of our proposed approach and validates the design choices.

3.3 Design Details

The workflow of FEDROLA is outlined in Algorithm 1, which utilizes the principle of similarity to safeguard FL models against malicious clients. The FL model architecture is typically composed of L layers, and instead of directly removing detected possible malicious updates, FEDROLA operates through a two-phase process within these layers to mitigate malicious impact.

Initially, FEDROLA employs a discount factor, labeled as D^{fac}, to mitigate the influence of potentially malicious updates (lines 18 and 21). This discount factor is designed to gradually converge to 1 as FEDROLA progresses. This factor is crucial because it reduces the impact of malicious updates at an early stage when the probability-based method may not be fully optimized for accuracy.

As FEDROLA evolves, it shifts its focus to rely more on probability measures, defined by α_i and β_i, for identifying and managing potentially malicious clients. The discount factor, defined as $D^{fac} = \frac{2}{1+e^{-\delta\times t}} - 1$, plays an instrumental role in the process of getting suitable $\{\alpha_i, \beta_i\}_{i\in[N]}$. Notably, this formula resembles the Tanh activation function. In our experiments, we set $\delta = 0.1$. The impact of this discount factor is meticulously evaluated in Section 4. This analysis aims to validate the efficiency of our defense method in real-world scenarios, emphasizing its capability to detect and neutralize malicious activities within FL systems.

Since we develop *LASI* in (4) and *PCSI* in (5) for computing similarity-based indices, we call the corresponding FEDROLA defense methods as FEDROLA-LASI and FEDROLA-PCSI, respectively. As discussed above regarding the nature of LASI and PCSI indices, FEDROLA-LASI prioritizes computational efficiency but offers modest performance, whereas FEDROLA-PCSI, though requiring more computational resources, yields significantly better performance .

Here we also evaluate the importance of each layer $l \in [L]$, operating under the common assumption that less than half of the total participants are malicious [1, 3, 5, 6, 13, 27, 31, 40, 45, 46]. If a layer flags more than 50% of participants as malicious, or none at all, we modify the values of α_l and β_l (lines 8-11). This recalibration of selection probabilities is crucial for identifying layers that are particularly robust against model poisoning attacks, a strategy we refer to as *Layer Selection* (line 4), depicted in Figure 1.

Based on the similarity-based index value, *Layer-based Detection* is implemented as shown in Algorithm 2, based on the presumption that malicious clients comprise less than 50% of the total. A higher percentage would likely lead to an increase in false positive rates, undermining the reliability of the detection process. The criterion $D^{thr} \leq 0$ is used to indicate the likely absence of further malicious clients. This is based on the observed trend where malicious entities tend to demonstrate consistently higher and positive similarity index values. The initial detection threshold, set at $D^{thr} = 0.9$, is determined based on the patterns observed in Figures 2 and 3.

For each client $i \in [N]$, increment β_i following a detection of malicious, or α_i otherwise (lines 17 and 20). The probabilities derived from α_i and β_i (line 24) aid in mitigating the effect of malicious updates. This ensures that malicious clients are assigned lower probabilities than benign ones, based on a lower false positive rate. In scenarios devoid of attacks, all clients will possess comparable probabilities, thereby preventing performance decline due to the absence of enough updates.

Algorithm 2 Layer-based Detection

```
Input: Index values $\{I_{i,l}\}_{i\in\mathcal{N}_t, l\in\mathcal{S}_L}$
1: Initialize the detection threshold $D^{thr} = 0.9$ which is based on
   previous observations, and $\mathcal{B}_{t,l} = \emptyset$
2: while $\mathcal{B}_{t,l} = \emptyset$ and $D^{thr} > 0$ do
3:     for $i \in \mathcal{N}^{(t)}$ do
4:         if $I_{i,l} \geq D^{thr}$ then
5:             $\mathcal{B}_{t,l} = \mathcal{B}_{t,l} \cup \{i\}$
6:         end if
7:     end for
8:     if $|\mathcal{B}_{t,l}|/|\mathcal{N}^{(t)}| \geq 0.5$ then
9:         $\mathcal{B}_{t,l} = \emptyset$
10:    end if
11:    if $\mathcal{B}_{t,l} = \emptyset$ then
12:        $D^{thr} = D^{thr} - 0.1$
13:    end if
14: end while
15: Return layer malicious clients $\mathcal{B}_{t,l}$
```

Algorithm 3 Layer-based Vote

```
Input: Detection results $\mathcal{B}'_t$
1: Initialize vote threshold $V^{thr} = |\mathcal{S}_L|$ and $\mathcal{B}_t = \emptyset$, note that
   $\mathcal{B}'_t = \{\mathcal{B}_{t,l}\}_{l\in\mathcal{S}_L}$
2: for $i \in \mathcal{N}_t$ do
3:     Initialize $V_i^{thr} = 0$
4:     for $l \in \mathcal{S}_L$ do
5:         if $i \in \mathcal{B}_{t,l}$ then
6:             $V_i^{thr} = V_i^{thr} + 1$
7:         end if
8:     end for
9: end for
10: while $\mathcal{B}_t = \emptyset$ and $V^{thr} > 0$ do
11:     for $i \in \mathcal{N}_t$ do
12:         if $V_i^{thr} \geq V^{thr}$ then
13:             $\mathcal{B}_t = \mathcal{B}_t \cup \{i\}$
14:         end if
15:     end for
16:     if $\mathcal{B}_t = \emptyset$ then
17:         $V^{thr} = V^{thr} - 1$
18:     end if
19: end while
20: Return final malicious clients $\mathcal{B}_t$
```

The *Layer-based Vote*, outlined in Algorithm 3, employs a methodical approach to identify malicious updates. Initially, an update is deemed malicious if flagged in every layer. If no malicious updates are detected for a client i (indicated by an empty $\mathcal{B}_t$ set), the algorithm reduces the detection threshold by one unit and recommences the voting process. This iterative approach helps refine the detection accuracy. Should the detection threshold reach zero, it implies that, as per this algorithm, there are no obvious malicious updates in the current evaluation round, thereby ensuring a careful balance between accuracy and false detection.

4 EXPERIMENTS

4.1 Experimental Setup

Tasks and datasets. In this work, we address three distinct tasks: (i) image classification using CIFAR-10, CIFAR-100 [19] and Fashion-MNIST [21] datasets; (ii) NLP for next-character prediction on the dataset of *The Complete Works of William Shakespeare* (Shakespeare) [25]; and (iii) human activity recognition of HARBox [8], which is collected from the smartphones of 121 users using a crowdsourcing approach. To facilitate our experiments, we simulate a heterogeneous partition of N clients by randomly sampling $\boldsymbol{p}_i \sim \text{Dir}_N(\alpha)$, with α representing the parameter of the Dirichlet distribution. We set $\alpha = 0.5$ as the default parameter in our experiments, in line with prior works [7, 9, 10, 13, 17, 28, 37, 38].

Machine learning models. We consider several representative models: VGG-11 [33], ResNet-18 [14], AlexNet [20], LSTM [18, 25], and fully connected DNN [22, 23]. In particular, we use VGG-11 as the global model architecture for CIFAR-10, ResNet-18 for CIFAR-100, AlexNet for Fashion-MNIST, LSTM for Shakespeare, DNN for HARBox, respectively.

Baselines. To comprehensively evaluate our FedRoLa, we benchmark against several state-of-the-art defense mechanisms: FLAIR [1], cosDefense [11], FLDetector [46], FLTrust [6], AFA [27], Multi-krum [5] and Trimmed-mean [40, 45]. Interestingly, Median [40, 45] often yields similar performance to Trimmed-mean, consistent with the experimental results in FLTrust [6]. Therefore, in this work, we focus solely on utilizing Trimmed-mean. Additionally, we also assess our approach against five of the most robust model poisoning attacks: Fang [13], LIE [3], Min-Sum and Min-Max [31], MPHM [32]. In line with previous research [3, 13, 31], we consider two adversary knowledge scenarios: (a) ***Full***, where the adversary has access to benign clients' gradients, and (b) ***Partial***, where the adversary lacks knowledge of gradient updates from benign clients.

Parameter settings. We utilize PyTorch [30] on Python 3, leveraging three NVIDIA RTX A6000 GPUs to implement defenses and poisoning attacks in a FL context. Each experiment is conducted across four independent trials, employing distinct random seeds to ensure variability and robustness in our findings.

The FL experiments simulate a network of $N = 128$ clients in total. In each training round, the central server randomly selects $n = 32$ clients to participate in updating the global model. Out of these, $M = 32$ clients are under adversarial control. The selection of malicious clients within these controlled clients is randomized by the server in each round, aligning with realistic scenarios of adversarial behavior in FL environments. Detailed configurations and hyperparameters for these experiments are outlined in Table 1.

Parameters	CIFAR-10	CIFAR-100	Fashion MNIST	Shakespeare	HARBox
Model	VGG-11	ResNet-18	AlexNet	LSTM	DNN
Local Epochs	3	3	3	2	2
Batch Size	16	128	16	128	16
Learning Rate	0.01	0.01	0.01	0.1	0.0001
Momentum	0.9	0.9	0.9	0.9	0.9
Weight Decay	1e-5	1e-5	1e-5	0	1e-5

Table 1: Details of the hyperparameters.

Dataset (Model)	Aggregation Algorithm	LIE	Fang	Min-Max	Min-Sum	MHPM
CIFAR-10 (VGG-11)	Multi-krum	58.07±0.36	57.40±0.58	56.17±0.58	56.70±0.61	56.94±0.64
	Trimmed-mean	56.94±0.46	55.18±0.62	54.14±0.68	54.79±0.63	54.24±0.65
	AFA	58.81±0.35	59.26±0.48	59.32±0.52	58.95±0.37	58.39±0.33
	FLTrust	59.88±0.49	59.95±0.59	59.75±0.68	59.57±0.35	59.10±0.64
	FLDetector	59.95±0.39	59.04±0.32	58.16±0.49	57.86±0.46	58.61±0.48
	FLAIR	54.84±0.70	55.54±0.52	53.13±0.63	54.28±0.57	54.89±0.63
	cosDefense	55.73±0.31	53.56±0.67	54.55±0.59	55.66±0.66	53.28±0.27
	FedRoLa-LASI	**60.67**±0.24	59.80±0.27	60.64±0.24	60.55±0.27	60.09±0.30
	FedRoLa-PCSI	60.57±0.33	**60.39**±0.42	**61.02**±0.36	**61.01**±0.23	**60.64**±0.25
CIFAR-100 (ResNet-18)	Multi-krum	27.59±0.42	26.74±0.64	27.13±0.70	26.31±0.70	25.65±0.59
	Trimmed-mean	28.95±0.19	28.91±0.53	29.14±0.68	28.33±0.54	27.39±0.69
	AFA	28.32±0.27	28.46±0.27	28.60±0.34	27.75±0.26	27.57±0.35
	FLTrust	28.81±0.30	28.08±0.29	28.66±0.32	28.63±0.29	28.25±0.41
	FLDetector	28.27±0.17	28.15±0.37	28.24±0.54	27.59±0.28	27.35±0.41
	FLAIR	28.11±0.14	27.95±0.15	27.45±0.15	26.20±0.17	25.54±0.18
	cosDefense	27.65±0.24	24.99±0.60	26.36±0.38	25.89±0.53	24.25±0.66
	FedRoLa-LASI	29.69±0.17	**30.21**±0.33	30.38±0.34	29.81±0.23	29.34±0.23
	FedRoLa-PCSI	**29.72**±0.18	30.01±0.23	**30.53**±0.26	**29.82**±0.26	**29.60**±0.15
Fashion MNIST (AlexNet)	Multi-krum	83.28±0.32	82.34±0.24	81.52±0.34	80.75±0.64	81.07±0.40
	Trimmed-mean	83.07±0.24	81.77±0.24	81.91±0.36	83.65±0.24	81.37±0.26
	AFA	85.23±0.33	85.19±0.40	85.57±0.33	86.01±0.39	84.93±0.71
	FLTrust	86.01±0.67	86.55±0.49	86.14±0.57	85.98±0.67	86.55±0.53
	FLDetector	86.20±0.37	85.81±0.29	86.18±0.17	85.30±0.30	84.80±0.31
	FLAIR	84.25±0.44	84.27±0.68	84.55±0.39	84.36±0.38	83.35±0.64
	cosDefense	82.20±0.22	81.84±0.74	83.31±0.50	82.39±0.23	81.27±0.61
	FedRoLa-LASI	**87.50**±0.15	86.39±0.22	87.27±0.21	86.66±0.20	86.36±0.31
	FedRoLa-PCSI	87.29±0.30	**87.22**±0.26	**87.64**±0.26	**86.74**±0.21	**87.64**±0.22
Shakespeare (LSTM)	Multi-krum	44.24±0.61	43.55±0.62	42.52±0.61	42.59±0.63	42.85±0.61
	Trimmed-mean	44.38±0.62	43.49±0.59	43.23±0.56	43.24±0.66	43.51±0.73
	AFA	46.95±0.70	46.90±0.60	46.38±0.65	46.65±0.65	47.44±0.70
	FLTrust	46.51±0.61	46.80±0.67	45.27±0.62	45.62±0.69	45.97±0.66
	FLDetector	46.42±0.68	45.73±0.59	46.12±0.62	46.45±0.63	46.30±0.69
	FLAIR	44.69±0.76	44.60±0.62	44.82±0.60	44.76±0.69	44.99±0.68
	cosDefense	44.76±0.72	42.54±0.68	44.07±0.72	44.75±0.68	42.77±0.64
	FedRoLa-LASI	48.03±0.70	46.38±0.68	47.78±0.58	47.91±0.66	46.93±0.70
	FedRoLa-PCSI	**48.06**±0.68	**47.42**±0.70	**47.80**±0.61	**47.98**±0.61	**47.92**±0.65
HARBox (DNN)	Multi-krum	39.76±0.26	38.26±0.48	35.44±0.58	35.18±0.41	36.84±0.28
	Trimmed-mean	33.43±0.20	32.32±0.29	30.61±0.38	31.40±0.23	31.57±0.30
	AFA	39.48±0.63	40.76±0.58	39.87±0.59	39.01±0.68	39.14±0.19
	FLTrust	43.12±0.70	42.47±0.59	41.59±0.67	42.14±0.62	41.78±0.52
	FLDetector	40.83±0.53	39.99±0.33	39.96±0.17	39.65±0.64	40.47±0.37
	FLAIR	39.24±0.15	38.57±0.24	37.47±0.24	36.50±0.14	36.17±0.50
	cosDefense	37.20±0.19	36.11±0.72	36.76±0.42	37.78±0.61	35.63±0.19
	FedRoLa-LASI	**43.34**±0.76	41.85±0.18	41.54±0.55	41.10±0.45	42.78±0.53
	FedRoLa-PCSI	42.99±0.60	**43.26**±0.30	**44.29**±0.44	**44.01**±0.56	**45.01**±0.60

Table 2: Comparisons of final test accuracy.

4.2 Main Results

In this section, we primarily report the results under the *Partial Knowledge* scenario, as it represents a more realistic setting for adversarial attacks in FL. We relegate the results under *Full Knowledge* to Appendix A.3 due to space constraints.

Final Test Accuracy. Table 2 summarizes the final test accuracy obtained from various combinations of defense mechanisms and attack strategies. Notably, FedRoLa demonstrates a remarkable capability to enhance final accuracy by up to 4% compared to the best existing state-of-the-art defense, particularly evident in the HARBox dataset. The analysis reveals that both variants of FedRoLa, namely FedRoLa-LASI and FedRoLa-PCSI, generally outperform other defenses, effectively mitigating the impact of malicious client updates. Intriguingly, in scenarios where the LIE attack targets datasets like CIFAR-10, Fashion MNIST, and HARBox, FedRoLa-LASI shows superior efficacy compared to FedRoLa-PCSI. This observation suggests that FedRoLa-LASI may be more advantageous in certain attack contexts. However, the performance of FedRoLa-LASI varies depending on the dataset and attack type. For instance, under Fang, Min-Max, and Min-Sum attacks on the HARBox dataset, FLTrust outperforms FedRoLa-LASI. While FedRoLa-PCSI consistently demonstrates robust defense capabilities

Dataset (Model)	Aggregation Algorithm	LIE	Fang	Min-Max	Min-Sum	MHPM
CIFAR-10 (VGG-11)	Multi-krum	0.40/0.66	0.43/0.63	0.43/0.44	0.40/0.45	0.43/0.57
	AFA	0.24/0.70	0.09/0.78	0.16/0.63	0.16/0.87	0.16/0.50
	FLTrust	0.22/0.88	0.23/0.89	0.22/0.88	0.22/0.66	0.21/0.61
	FLDetector	0.07/0.79	0.03/0.69	0.10/0.54	0.16/0.65	0.04/0.59
	FLAIR	0.38/0.88	0.38/0.67	0.47/0.89	0.45/0.68	0.45/0.70
	cosDefense	0.47/0.89	0.85/0.62	0.45/0.82	0.48/0.88	0.81/0.50
	FedRoLa-LASI	**0.08/0.95**	**0.14/0.92**	**0.04/0.90**	**0.04/0.90**	**0.05/0.95**
	FedRoLa-PCSI	**0.03/0.98**	**0.03/0.98**	**0.03/0.97**	**0.03/0.98**	**0.03/0.98**
CIFAR-100 (ResNet-18)	Multi-krum	0.51/0.52	0.54/0.46	0.54/0.51	0.51/0.44	0.48/0.24
	AFA	0.14/0.60	0.04/0.65	0.15/0.89	0.17/0.59	0.16/0.55
	FLTrust	0.17/0.87	0.12/0.69	0.12/0.63	0.15/0.59	0.15/0.86
	FLDetector	0.13/0.55	0.12/0.51	0.12/0.52	0.10/0.51	0.08/0.51
	FLAIR	0.03/0.52	0.03/0.51	0.02/0.51	0.03/0.50	0.03/0.50
	cosDefense	0.43/0.73	0.76/0.60	0.45/0.51	0.49/0.62	0.87/0.50
	FedRoLa-LASI	**0.04/0.91**	**0.19/0.92**	**0.12/0.90**	**0.12/0.95**	**0.10/0.90**
	FedRoLa-PCSI	**0.03/0.92**	**0.03/0.94**	**0.03/0.92**	**0.05/0.97**	**0.04/0.91**
Fashion MNIST (AlexNet)	Multi-krum	0.40/0.61	0.43/0.60	0.43/0.56	0.40/0.41	0.43/0.43
	AFA	0.24/0.50	0.20/0.59	0.16/0.68	0.17/0.65	0.21/0.50
	FLTrust	0.21/0.79	0.21/0.51	0.19/0.68	0.22/0.89	0.20/0.67
	FLDetector	0.18/0.77	0.09/0.64	0.09/0.72	0.12/0.63	0.09/0.53
	FLAIR	0.38/0.69	0.38/0.68	0.49/0.88	0.45/0.89	0.45/0.59
	cosDefense	0.58/0.88	0.87/0.78	0.52/0.89	0.55/0.88	0.80/0.52
	FedRoLa-LASI	**0.08/0.96**	**0.15/0.93**	**0.08/0.93**	**0.08/0.93**	**0.06/0.92**
	FedRoLa-PCSI	**0.03/0.96**	**0.04/0.98**	**0.03/0.98**	**0.03/0.98**	**0.02/0.98**
Shakespeare (LSTM)	Multi-krum	0.43/0.66	0.43/0.63	0.43/0.40	0.43/0.55	0.41/0.59
	AFA	0.22/0.80	0.10/0.89	0.20/0.89	0.15/0.82	0.15/0.88
	FLTrust	0.21/0.67	0.18/0.89	0.18/0.51	0.20/0.65	0.22/0.66
	FLDetector	0.25/0.51	0.16/0.79	0.10/0.51	0.10/0.61	0.05/0.78
	FLAIR	0.37/0.86	0.37/0.88	0.39/0.88	0.37/0.88	0.39/0.88
	cosDefense	0.44/0.79	0.70/0.54	0.45/0.55	0.38/0.51	0.80/0.79
	FedRoLa-LASI	**0.05/0.95**	**0.17/0.92**	**0.03/0.93**	**0.05/0.92**	**0.06/0.94**
	FedRoLa-PCSI	**0.04/0.98**	**0.03/0.98**	**0.04/0.93**	**0.02/0.98**	**0.03/0.98**
HARBox (DNN)	Multi-krum	0.03/0.70	0.03/0.66	0.03/0.46	0.03/0.49	0.37/0.48
	AFA	0.27/0.69	0.25/0.60	0.20/0.74	0.22/0.50	0.20/0.51
	FLTrust	0.35/0.89	0.33/0.88	0.33/0.71	0.34/0.57	0.32/0.80
	FLDetector	0.15/0.89	0.14/0.53	0.04/0.84	0.03/0.83	0.09/0.73
	FLAIR	0.47/0.73	0.45/0.71	0.46/0.63	0.50/0.61	0.49/0.53
	cosDefense	0.49/0.52	0.81/0.89	0.47/0.88	0.46/0.89	0.76/0.52
	FedRoLa-LASI	**0.13/0.92**	**0.13/0.90**	**0.06/0.91**	**0.07/0.91**	**0.06/0.91**
	FedRoLa-PCSI	**0.04/0.92**	**0.04/0.94**	**0.03/0.94**	**0.05/0.97**	**0.04/0.98**

Table 3: Comparisons of FPR and TPR.

against advanced attacks, its computational complexity is higher than that of FedRoLa-LASI. This aspect might make FedRoLa-LASI a more practical choice in real-world applications where efficiency is crucial. Overall, both FedRoLa-LASI and FedRoLa-PCSI maintain consistent performance across most scenarios, underlining their reliability as defense solutions in FL.

True and False Positive Rate. The False Positive Rate (FPR) is a crucial metric that quantifies the frequency of benign clients being erroneously flagged as malicious. In contrast, the True Positive Rate (TPR) is an indicator of the effectiveness in correctly identifying genuine malicious clients. Table 3 provides a detailed evaluation of the FPR and TPR achieved by FedRoLa in detecting malicious updates. The results demonstrate that FedRoLa significantly lowers the FPR to approximately 6.4%, marking a notable improvement over the average FPR of 33.4% observed in existing defense methods. Additionally, FedRoLa attains a TPR of around 92.8%, substantially surpassing the average TPR of 67.3% typically found in current defense strategies.

Our comparative analysis underscores that both FedRoLa-LASI and FedRoLa-PCSI outperform other available defense mechanisms, achieving lower FPRs and higher TPRs. It is observed that FedRoLa-LASI exhibits a slightly increased FPR and a marginally reduced TPR when compared to FedRoLa-PCSI across most evaluated scenarios. This distinction highlights the efficacy of the FedRoLa framework, particularly in its ability to accurately discern between benign and malicious updates in federated learning environments.

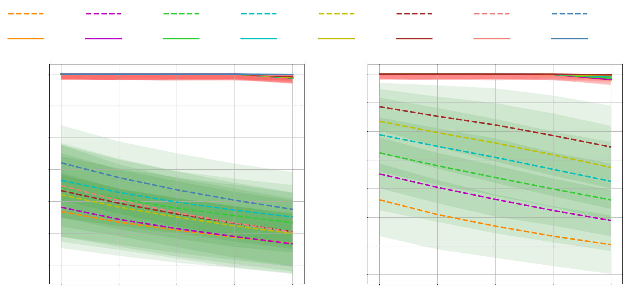

Figure 4: The impact of the value K in Top-K.

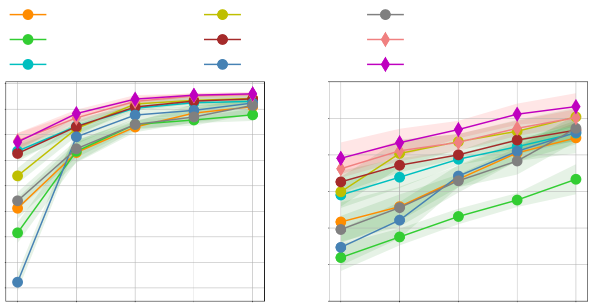

Figure 5: The impact of the Non-IID degree α.

4.3 Ablation Study

This section presents results focusing on the choice of K in the Top-K method (Equation (5)), variations in Non-IID data distribution, different values of δ for the discount factor, the total number of clients (N), the count of adversary-controlled clients (M) under two cases, and the application of FedRoLa to the entire model.

Different K values. The main results primarily utilize the top-3 similarity scores as defined in *PCSI*, demonstrating enhanced efficacy in detecting malicious clients compared to other methods. To further assess the impact of varying the K values in Top-K, we conduct extended experiments. The results, as shown in Figure 4, reveal a notable pattern: increasing the top-K value leads to a decrease in the average similarity score. However, this decline does not obscure the distinction between benign and malicious updates. Choosing a top-3 setup for our main experiments strikes an effective balance, ensuring both accuracy in identifying malicious clients and maintaining manageable computational requirements.

Non-IID degree. In our experiments, we distribute data heterogeneously across N clients using a Dirichlet distribution with parameter α. For our main analysis, we set α at 0.5 to represent a middle ground between non-IID and independent and IID data scenarios. To test the resilient performance of FedRoLa against the non-IID degree, we vary the value of α between 0.1 and 0.9. As illustrated in Figure 5, FedRoLa consistently outperforms other methods across all α values. This robust performance is further supported by results shown in Figures 10 and 11 in Appendix A.2, in which the difference between the malicious and the benign is also obvious.

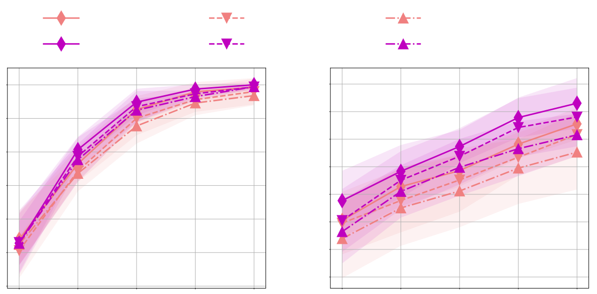
Figure 6: The impact of the value of δ.

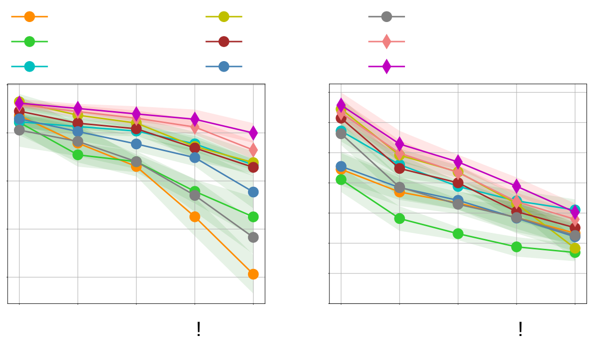
Figure 7: The impact of total number of clients N.

Evaluation of δ in discount factor. Our experimental results recommend setting the δ value to 0.1 based on careful evaluation. This offers a balanced approach: with higher δ, our FedRoLa quickly reduces the impact of potential malicious updates, at the risk of including more of these updates with significant weights in the global model. Conversely, a lower δ prolongs the period for mitigating malicious influence, which may result in the exclusion of more data samples, especially when malicious clients are few. Therefore, we further evaluate the impact of the value of δ. Figure 6 showcases that a δ of 0.1 strikes an optimal balance. This value effectively counters malicious updates while ensuring adequate data inclusion for training. This equilibrium is pivotal, as it navigates the fine line between efficiently neutralizing malicious activities and maintaining a robust training dataset.

Number of clients. In our experiments, we operate within a FL setting with a total number of $N = 128$ clients. Building upon these conditions, we proceed to exploring its impact by adjusting the total client count while maintaining the same experimental setups, using AlexNet for the Fashion-MNIST dataset and a DNN for HARBox dataset, with the Dirichlet distribution parameter α set to 0.5. Figure 7 demonstrates that, irrespective of the client population size, FedRoLa consistently surpasses the performance of existing state-of-the-art methods.

Number of controlled clients by one adversary. We initially assume that the total number of compromised clients is $M = 32$. In this setting, the parameter server randomly selects participating clients in each training round t. Now we further explore the impact

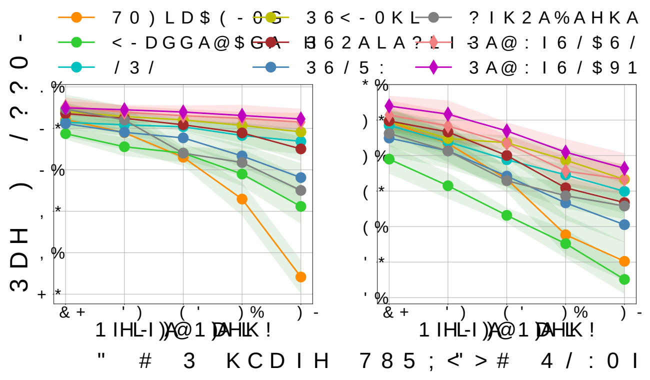
Figure 8: The impact of total controlled clients M by one adversary.

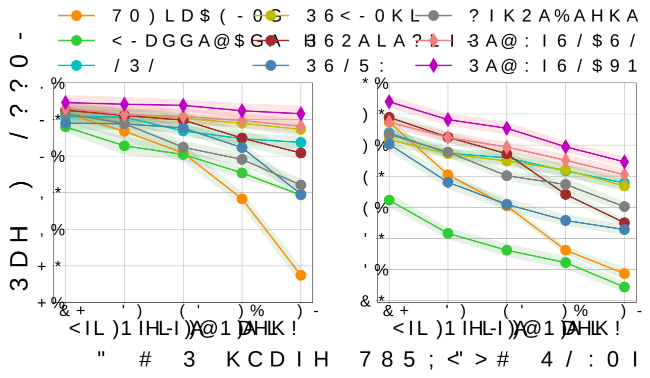
Figure 9: The impact of total controlled clients M by two adversaries.

of varying M within the range of 16 to 48, ensuring it remains below the 50% threshold. As illustrated in Figure 8, FedRoLa exhibits optimal performance, particularly when the value of M decreases. This result highlights FedRoLa's effectiveness in scenarios where attacks are absent with high probability, a condition that typically corresponds to a higher False Positive Rate (FPR) in state-of-the-art defense methods. In many scenarios, while FedRoLa-LASI may not outperform FedRoLa-PCSI in effectiveness, its lower operational demands render it more suitable for a wide range of real-world applications.

Number of controlled clients by two adversaries. State-of-the-art methods such as FLTrust and FLDetector predominantly address threats from a single group of malicious clients. However, we further investigate scenarios involving multiple groups of malicious clients. To tackle this, we utilized PCSI to prioritize top-K similarity for each client. Our observations show that malicious updates from the same group exhibit higher similarity values, resulting in elevated PCSI scores. Inspired by this, FedRoLa is designed to identify and distinguish between different groups of malicious clients effectively. This capability is particularly valuable as it not only enhances the model's defense against single-group attacks but also equips it with the necessary tools to navigate the complexities introduced by multiple adversary groups. Results can be found in Figure 9.

Applying FedRoLa to the entire model. In this scenario, FedRoLa is applied to the entire model. The corresponding outcomes

Dataset (Model)	Type	Index	LIE	Fang	Min-Max	Min-Sum	MPHM
CIFAR-10 (VGG-11)	Original	LASI	**60.67**±0.24	59.80±0.27	60.64±0.24	60.55±0.27	60.09±0.30
		PCSI	60.57±0.33	**60.39**±0.42	**61.02**±0.36	**61.01**±0.23	**60.64**±0.25
	Variation	LASI	60.23±0.33	59.48±0.31	59.27±0.31	60.09±0.28	57.85±0.28
		PCSI	60.52±0.28	60.28±0.28	61.13±0.28	60.20±0.28	60.62±0.28
CIFAR-100 (ResNet-18)	Original	LASI	29.69±0.17	**30.21**±0.33	30.38±0.34	29.81±0.23	29.34±0.23
		PCSI	**29.72**±0.18	30.01±0.23	**30.53**±0.26	**29.82**±0.26	**29.60**±0.15
	Variation	LASI	28.22±0.27	28.88±0.28	29.33±0.29	29.11±0.29	28.72±0.27
		PCSI	29.87±0.32	30.24±0.27	30.17±0.25	29.95±0.25	29.49±0.20
Fashion MNIST (AlexNet)	Original	LASI	**87.50**±0.15	86.39±0.22	87.27±0.21	86.66±0.20	86.36±0.31
		PCSI	87.29±0.30	**87.22**±0.26	**87.64**±0.26	**86.74**±0.21	**87.64**±0.22
	Variation	LASI	87.05±0.34	86.02±0.30	86.58±0.32	86.12±0.32	85.33±0.35
		PCSI	87.38±0.34	86.70±0.31	87.34±0.29	86.27±0.33	86.37±0.30
Shakespeare (LSTM)	Original	LASI	48.03±0.70	46.38±0.68	47.78±0.58	47.91±0.66	46.93±0.70
		PCSI	**48.06**±0.68	**47.42**±0.70	**47.80**±0.61	**47.98**±0.61	**47.92**±0.65
	Variation	LASI	47.75±0.57	44.80±0.54	43.83±0.64	45.51±0.58	44.04±0.60
		PCSI	48.19±0.65	46.46±0.59	47.32±0.66	47.73±0.63	47.60±0.67
HARBox (DNN)	Original	LASI	**43.34**±0.76	41.85±0.18	41.54±0.55	41.10±0.45	42.78±0.53
		PCSI	42.99±0.60	**43.26**±0.30	**44.29**±0.44	**44.01**±0.56	**45.01**±0.60
	Variation	LASI	41.45±0.35	39.60±0.28	39.73±0.40	41.06±0.39	40.01±0.30
		PCSI	43.59±0.39	41.02±0.27	42.24±0.31	42.50±0.47	43.08±0.28

Table 4: Comparisons of final test accuracy between original FedRoLa and variation FedRoLa.

are presented in Table 4. We observe that FedRoLa maintains its effectiveness when treating the entire model as a single layer. However, we note that there are additional challenges, particularly with FedRoLa-LASI. These challenges arise due to inefficiencies in specific layers and an increased risk of false positives from an inactive layer-based voting mechanism.

4.4 Similarity-based Attack

Our proposed FedRoLa marks a notable progression in detecting malicious updates, surpassing other leading methods. To thoroughly test FedRoLa's robustness and motivated by the design of FedRoLa using similarity-based indices, we further develop a heuristic attack method named SimAttack by leveraging similarity. Most existing attack methods, easily identified by Byzantine-robust algorithms due to their reliance on abrupt changes to the update vector, are less effective, as observed from our experimental results discussed above. In contrast, our advanced attack leverages cosine similarity, subtly manipulating updates to mimic legitimate ones. This approach adaptively alters critical aspects of the update, maximizing damage while maintaining a benign appearance to avoid detection.

Besides accuracy, the effectiveness of an attack could be gauged by FPR and TPR. The data in Table 5 highlight that SimAttack, while having a lower FPR compared to benchmark standards, significantly reduces the TPR across multiple scenarios. This indicates that while SimAttack is stealthier, it is less effective in incorrectly being identified. Despite this, SimAttack surpasses other attack methods in effectiveness, underscoring the necessity for more sophisticated attack strategies to rigorously test and challenge the resilience of FedRoLa. A comprehensive analysis and detailed methodology of SimAttack provided in Appendix B. Future research will focus on developing more potent attack strategies, with SimAttack serving as a key starting point for this exploration.

5 CONCLUSION

In this work, we introduced FedRoLa, a novel and efficient algorithm designed for the robust defense of FL systems against model poisoning attacks. By eschewing reliance on historical data and validation datasets, FedRoLa innovatively employs global model parameters and client updates, focusing on the unique attributes of deep neural network architectures. This results in significant enhancements in FL security, evidenced by substantial improvements in model accuracy, a marked reduction in false positives, and an impressive true positive rate. Our comprehensive evaluations across diverse models and datasets validate FedRoLa's effectiveness. The future direction includes exploring more sophisticated attacks like SimAttack, further strengthening FL attacks.

Dataset (Model)	**Aggregation Algorithm**	**Accuracy**		**FPR/TPR**	
		SOTA	SimAttack	SOTA	SimAttack
CIFAR-10 (VGG-11)	Multi-krum	56.17±0.58	55.60±0.63	0.43/0.44	0.31/0.41
	Trimmed-mean	54.14±0.68	53.30±0.69	-	-
	AFA	58.39±0.33	58.12±0.44	0.24/0.50	0.17/0.46
	FLTrust	59.10±0.64	58.30±0.62	0.23/0.61	0.23/0.48
	FLDetector	57.86±0.46	57.43±0.64	0.16/0.54	0.05/0.51
	FLAIR	53.13±0.63	52.42±0.72	0.47/0.67	0.42/0.48
	cosDefense	53.28±0.27	52.47±0.71	0.85/0.50	0.62/0.39
	FedRoLa-LASI	59.80±0.27	58.42±0.56	0.14/0.90	**0.06/0.59**
	FedRoLa-PCSI	**60.39**±0.42	**59.17**±0.35	0.03/0.97	**0.04/0.71**
CIFAR-100 (ResNet-18)	Multi-krum	25.65±0.59	25.15±0.59	0.54/0.24	0.54/0.22
	Trimmed-mean	27.39±0.69	26.59±0.52	-	-
	AFA	27.57±0.35	26.96±0.39	0.17/0.55	0.18/0.26
	FLTrust	28.08±0.29	28.03±0.22	0.17/0.59	0.14/0.30
	FLDetector	27.35±0.41	26.56±0.39	0.13/0.51	0.11/0.24
	FLAIR	25.54±0.18	25.06±0.15	0.03/0.50	0.03/0.24
	cosDefense	24.25±0.66	23.76±0.31	0.87/0.50	0.63/0.38
	FedRoLa-LASI	29.34±0.23	28.66±0.30	0.19/0.90	**0.15/0.47**
	FedRoLa-PCSI	**29.60**±0.15	**28.92**±0.26	0.05/0.91	**0.04/0.52**
Fashion MNIST (AlexNet)	Multi-krum	80.75±0.64	80.11±0.34	0.43/0.41	0.27/0.35
	Trimmed-mean	81.37±0.26	80.28±0.43	-	-
	AFA	84.93±0.71	84.21±0.40	0.24/0.50	0.15/0.43
	FLTrust	85.98±0.67	84.92±0.62	0.22/0.51	0.20/0.49
	FLDetector	84.80±0.31	84.32±0.23	0.18/0.53	0.07/0.52
	FLAIR	83.35±0.64	82.33±0.42	0.49/0.59	0.44/0.44
	cosDefense	81.27±0.61	81.02±0.34	0.87/0.52	0.73/0.46
	FedRoLa-LASI	86.36±0.31	85.02±0.44	0.15/0.92	**0.09/0.69**
	FedRoLa-PCSI	**86.74**±0.21	**85.32**±0.41	0.03/0.96	**0.03/0.71**
Shakespeare (LSTM)	Multi-krum	42.52±0.61	41.75±0.65	0.43/0.40	0.40/0.39
	Trimmed-mean	43.23±0.56	42.85±0.66	-	-
	AFA	46.38±0.65	45.48±0.76	0.22/0.80	0.19/0.46
	FLTrust	45.27±0.62	44.72±0.67	0.22/0.51	0.20/0.43
	FLDetector	45.73±0.59	44.67±0.72	0.25/0.51	0.09/0.43
	FLAIR	44.60±0.62	44.32±0.69	0.39/0.86	0.42/0.48
	cosDefense	42.54±0.68	41.35±0.57	0.80/0.51	0.64/0.42
	FedRoLa-LASI	46.38±0.68	45.94±0.64	0.17/0.92	**0.08/0.63**
	FedRoLa-PCSI	**47.42**±0.70	**46.53**±0.69	0.03/0.93	**0.05/0.78**
HARBox (DNN)	Multi-krum	35.18±0.41	34.62±0.53	0.37/0.46	0.03/0.45
	Trimmed-mean	30.61±0.38	30.18±0.30	-	-
	AFA	39.01±0.68	38.19±0.64	0.27/0.50	0.18/0.48
	FLTrust	41.59±0.67	39.78±0.30	0.35/0.57	0.33/0.52
	FLDetector	39.65±0.64	39.18±0.24	0.15/0.53	0.03/0.52
	FLAIR	36.17±0.50	34.70±0.38	0.50/0.53	0.45/0.49
	cosDefense	35.63±0.19	35.27±0.43	0.81/0.52	0.59/0.50
	FedRoLa-LASI	41.10±0.45	39.70±0.42	0.13/0.90	**0.07/0.67**
	FedRoLa-PCSI	**42.99**±0.60	**41.24**±0.62	0.05/0.92	**0.06/0.69**

Table 5: The performance of SimAttack.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) grants 2315612, 2315613, 2315614, 2327480 and 2348452. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Joshua Zhao Qiang Qiu Saurabh Bagchi Atul Sharma, Wei Chen and Somali Chaterji. 2023. FLAIR: Defense against Model Poisoning Attack in Federated Learning. In *Proc. of ASIA CCS*.

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

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

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

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

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

[7] Xiaoyu Cao and Neil Zhenqiang Gong. 2022. MPAF: Model Poisoning Attacks to Federated Learning based on Fake Clients. *arXiv preprint arXiv:2203.08669* (2022).

[8] Sashank Reddiand Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecny, Sanjiv Kumar, and H Brendan McMahan. 2021. Adaptive federated optimization.. In *Proc. of ICLR*.

[9] Hong-You Chen and Wei-Lun Chao. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *Proc. of ICLR*.

[10] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. 2022. DisPFL: Towards Communication-Efficient Personalized Federated Learning via Decentralized Sparse Training. In *Proc. of ICML*. 4587–4604.

[11] Tuo Zhang Duygu Nur Yaldiz and Salman Avestimehr. 2023. Secure Federated Learning against Model Poisoning Attacks via Client Filtering. In *Proc of ICLR Workshop*.

[12] Mahdi El El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The hidden vulnerability of distributed learning in byzantium. In *Proc. of ICML*.

[13] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *Proc. of USENIX Security*.

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

[15] Ahmed Imteaj, Khandaker Mamun Ahmed, Urmish Thakker, Shiqiang Wang, Jian Li, and M Hadi Amini. 2022. Federated learning for resource-constrained IoT devices: panoramas and state of the art. *Federated and Transfer Learning* (2022), 7–27.

[16] Matthew Jagielski, Aline Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *Proc of IEEE S&P*. 19–35.

[17] Zhifeng Jiang, Wei Wang, Baochun Li, and Bo Li. 2022. Pisces: Efficient federated learning via guided asynchronous training. In *SoCC*. 370–385.

[18] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proc. of AAAI*.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning Multiple Layers of Features from Tiny Images. (2009).

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Proc. of NIPS*.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[22] Ang Li, Jingwei Sun, Pengcheng Li, Yu Pu, Hai Li, and Yiran Chen. 2021. Hermes: an efficient federated learning framework for heterogeneous mobile clients. In *Proc. of ACM MobiCom*. 420–437.

[23] Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. 2022. PyramidFL: A fine-grained client selection framework for efficient federated learning. In *Proc. of ACM MobiCom*. 158–171.

[24] Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. 2019. Universal multi-party poisoning attacks. In *Proc. of ICML*.

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proc. of AISTATS*. 1273–1282.

[26] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proc of ACM AISec*. 27–38.

[27] 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).

[28] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. 2022. Fedbabu: Towards enhanced representation for federated image classification. In *Proc. of ICLR*.

[29] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. 2021. Clusterfl: a similarity-aware federated learning system for human activity recognition. In *Proc. of ACM MobiSys*. 54–66.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *Proc. of NIPS*.

[31] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Proc. of NDSS*.

[32] Lei Shi, Zhen Chen, Yucheng Shi, Lin Wei, Yongcai Tao, Mengyang He, Qingxian Wang, Yuan Zhou, and Yufei Gao. 2023. MPHM: Model poisoning attacks on federal learning using historical information momentum. *Security and Safety* (2023).

[33] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-scale Image Recognition. In *Proc. of ICLR*.

[34] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated Multi-Task Learning. In *Proc. of NeurIPS*.

[35] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963* (2019).

[36] Paul Voigt and Axel von dem Bussche. 2017. *The EU General Data Protection Regulation (GDPR)* (1st ed.). Springer International Publishing.

[37] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *Proc. of ICLR*.

[38] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Proc. of NeurIPS*.

[39] Chulin Xie, Keli Huang, Pin Yu Chen, and Bo Li. 2020. Dba: Distributed backdoor attacks against federated learning. In *Proc. of ICLR*.

[40] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2018. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116* (2018).

[41] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2020. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Proc. of UAI*.

[42] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. 2023. Criticalfl: A critical learning periods augmented client selection framework for efficient federated learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2898–2907.

[43] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. 2023. DeFL: defending against model poisoning attacks in federated learning via critical learning periods awareness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10711–10719.

[44] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2019), 1–19.

[45] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proc. of ICML*.

[46] Zaixi Zhang, Xiaoyu Cao, Jinayuan Jia, and Neil Zhenqiang Gong. 2022. FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. In *Proc. of ACM SIGKDD*.

[47] Haomin Zhuang, Mingxian Yu, Hao Wang, Yang Hua, Jian Li, and Xu Yuan. 2023. Backdoor Federated Learning by Poisoning Backdoor-Critical Layers. *arXiv preprint arXiv:2308.04466* (2023).

A APPENDIX: ADDITIONAL RESULTS

A.1 Datasets

We explore three tasks across various datasets and models:

- **Image Classification**: CIFAR-10 and CIFAR-100 [19], each with 60,000 color images (32×32 pixels) in 10 and 100 classes, split into 50,000 training and 10,000 test images. The Fashion-MNIST dataset [21] is also used, comprising 60,000 training and 10,000 test 28×28 grayscale images in 10 classes.
- **Natural Language Processing (NLP)**: For next-character prediction, utilize "The Complete Works of William Shakespeare" dataset [25], containing 734,057 training and 70,657 test data points over 74 characters.
- **Human Activity Recognition (HARBox)**: This task uses the HARBox dataset [29], featuring 9-axis IMU data from smartphones of 121 users.

For classification tasks, we deploy several models accordingly: ResNet-18 [14], VGG-11 [33], and AlexNet [20]. For NLP's next-character prediction, use an LSTM language model, following the

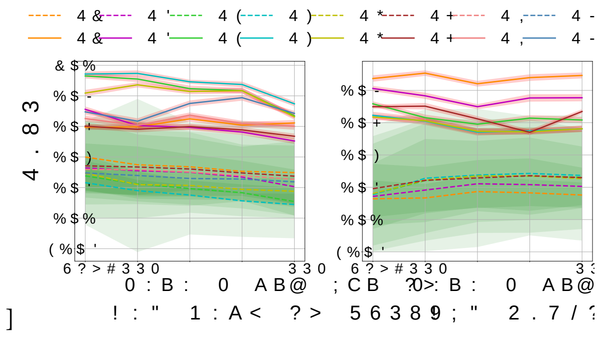
Figure 10: From Non-IID to IID: *LASI* by Multi-krum.

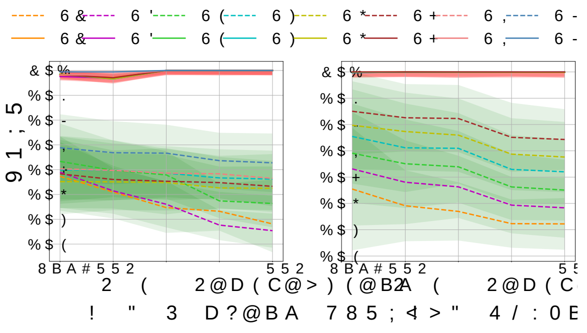
Figure 11: From Non-IID to IID: *PCSI* by Multi-krum.

configuration in [18]. The DNN model used for the HARBox dataset. This diverse array of datasets and models provides a robust platform for evaluating our federated learning strategies.

In relation to the data split discussed in Section 4.1, benign clients possessing similar data samples might generate gradients that exhibit a degree of similarity. However, the extent of similarity among these benign gradients generally falls short of the level observed within malicious updates. This discrepancy arises because, in practice it is unlikely for any two clients to possess identical data samples. For instance, while the data distribution of client i may align closely with that of client j, their data points are not exactly the same. Thus, though gradients from benign updates may bear resemblance to one another, their similarities are typically less pronounced than those observed among malicious updates. As shown in Figure 5, we distribute data samples across $N = 128$ clients using a Dirichlet distribution with parameter α. Even under nearly IID (with $\alpha = 0.9$), FedRoLa surpasses the performance of existing state-of-the-art methods.

A.2 Design of FedRoLa

In the main part of this work, the motivation of our design is based on the Multi-krum, because Multi-krum is one of the widely used algorithms and it can be used to find malicious clients, which is in line with the proposed design.

From Non-IID to IID. In alignment with the previous experimental setup, we adjust the value of α in the Dirichlet distribution from 0.1 to 0.9 to mimic various data distribution scenarios. Consistent with the previous observations, find similar trends in Figures 10 and 11, which greatly validate the proposed design of FedRoLa.

Dataset (Model)	Aggregation Algorithm	LIE	Fang	Min-Max	Min-Sum	MHPM
CIFAR-10 (VGG-11)	Multi-krum	55.81±0.46	56.46±0.48	54.93±0.61	55.26±0.72	56.44±0.66
	Trimmed-mean	54.18±0.69	53.70±0.60	52.04±0.63	52.50±0.70	54.22±0.71
	AFA	58.38±0.36	58.99±0.35	58.54±0.48	57.81±0.34	58.79±0.51
	FLTrust	59.56±0.47	59.82±0.66	59.16±0.61	59.93±0.42	59.06±0.69
	FLDetector	58.42±0.57	59.16±0.37	57.62±0.65	58.53±0.68	57.97±0.47
	FLAIR	53.04±0.74	54.50±0.64	51.37±0.62	51.58±0.70	54.74±0.71
	cosDefense	55.52±0.61	53.72±0.67	54.94±0.47	54.24±0.63	53.13±0.67
	FedRoLa-LASI	60.16±0.47	59.12±0.47	60.51±0.56	59.72±0.28	60.15±0.39
	FedRoLa-PCSI	60.69±0.31	59.88±0.26	60.82±0.48	60.93±0.27	60.71±0.32
CIFAR-100 (ResNet-18)	Multi-krum	25.86±0.54	25.31±0.75	25.52±0.60	24.21±0.59	25.06±0.63
	Trimmed-mean	26.98±0.36	27.01±0.46	26.96±0.70	25.60±0.63	26.13±0.63
	AFA	27.49±0.16	27.58±0.30	27.42±0.21	27.71±0.22	26.93±0.30
	FLTrust	28.43±0.29	28.51±0.35	28.31±0.28	28.99±0.26	29.21±0.18
	FLDetector	28.08±0.32	28.18±0.38	27.42±0.70	27.33±0.45	27.35±0.59
	FLAIR	27.98±0.15	27.69±0.20	25.56±0.16	25.27±0.18	26.97±0.15
	cosDefense	26.37±0.41	25.40±0.56	26.84±0.58	26.20±0.63	24.56±0.70
	FedRoLa-LASI	28.96±0.19	29.18±0.56	29.48±0.28	29.44±0.33	29.08±0.22
	FedRoLa-PCSI	29.35±0.16	29.48±0.18	30.26±0.22	29.54±0.23	29.39±0.16
Fashion MNIST (AlexNet)	Multi-krum	82.59±0.34	82.26±0.31	81.09±0.34	80.21±0.64	80.70±0.30
	Trimmed-mean	80.94±0.34	80.73±0.32	80.41±0.36	80.46±0.38	80.54±0.37
	AFA	84.32±0.28	84.73±0.40	84.58±0.30	84.54±0.32	84.87±0.32
	FLTrust	85.36±0.67	85.02±0.60	84.98±0.57	85.02±0.66	85.42±0.57
	FLDetector	84.88±0.61	85.66±0.34	85.79±0.19	84.66±0.28	85.01±0.33
	FLAIR	84.79±0.43	84.50±0.39	83.52±0.69	84.78±0.35	82.52±0.63
	cosDefense	81.61±0.21	80.96±0.71	81.11±0.27	80.50±0.42	81.01±0.64
	FedRoLa-LASI	86.57±0.15	86.09±0.32	85.72±0.41	86.46±0.27	86.04±0.51
	FedRoLa-PCSI	87.44±0.27	86.97±0.25	86.54±0.32	86.96±0.36	87.22±0.32
Shakespeare (LSTM)	Multi-krum	43.98±0.69	43.54±0.61	41.69±0.64	41.79±0.69	42.40±0.70
	Trimmed-mean	43.85±0.69	43.49±0.59	42.11±0.64	42.18±0.64	42.65±0.74
	AFA	46.32±0.64	46.93±0.61	45.50±0.61	45.75±0.69	47.08±0.69
	FLTrust	46.21±0.74	46.85±0.69	45.82±0.64	45.34±0.67	46.27±0.72
	FLDetector	45.41±0.67	45.33±0.58	45.60±0.67	46.07±0.65	45.94±0.69
	FLAIR	44.31±0.57	43.60±0.67	44.60±0.69	44.05±0.72	43.52±0.64
	cosDefense	45.23±0.68	42.26±0.59	46.52±0.66	45.64±0.69	42.57±0.66
	FedRoLa-LASI	47.70±0.59	46.40±0.61	47.15±0.58	47.81±0.74	46.78±0.71
	FedRoLa-PCSI	48.45±0.61	47.25±0.66	47.86±0.58	48.04±0.66	48.01±0.57
HARBox (DNN)	Multi-krum	36.21±0.59	35.97±0.46	35.10±0.14	35.18±0.17	36.16±0.32
	Trimmed-mean	30.27±0.16	29.88±0.15	29.48±0.29	29.48±0.60	30.66±0.13
	AFA	38.89±0.42	40.30±0.24	38.12±0.43	37.97±0.42	38.49±0.65
	FLTrust	41.13±0.67	40.75±0.70	40.97±0.51	40.66±0.72	40.60±0.76
	FLDetector	40.54±0.72	40.86±0.48	39.37±0.16	40.06±0.33	39.49±0.39
	FLAIR	36.85±0.26	36.99±0.67	34.24±0.21	35.46±0.38	35.34±0.69
	cosDefense	36.89±0.32	35.53±0.21	36.20±0.24	36.42±0.67	35.49±0.16
	FedRoLa-LASI	42.28±0.59	41.88±0.28	41.36±0.62	40.94±0.68	41.21±0.58
	FedRoLa-PCSI	42.87±0.60	42.84±0.67	42.33±0.66	42.35±0.73	43.76±0.70

Table 6: Comparisons of final accuracy with full knowledge.

A.3 Main Results

Final Test Accuracy. The earlier section presented final accuracy results under partial knowledge conditions. This part focuses on outcomes under full knowledge scenarios, detailed in Table 6. The findings here align closely with those previously discussed, offering similar conclusions regarding model performance and effectiveness.

True and False Positive Rate. For a comprehensive view of the True Positive Rate (TPR) and False Positive Rate (FPR) under both partial (Table 3) and full (Table 7) knowledge scenarios. These tables provide detailed insights into the performance metrics across different conditions.

B APPENDIX: DESIGN OF SIMATTACK

The FedRoLa algorithm represents a significant advancement in the detection of malicious updates within the field, outperforming other state-of-the-art methods. Nevertheless, to rigorously validate the robustness of FedRoLa, we aim to construct advanced attack methods tailored to bypass its detection capabilities. Traditional attack algorithms typically alter the direction of the entire update vector, a blatant action that current Byzantine-robust algorithms can readily identify. This ease of detection stems from these algorithms' dependency on distance-based mechanisms, which operate

Dataset (Model)	**Aggregation Algorithm**	**LIE**	**Fang**	**Min-Max**	**Min-Sum**	**MHPM**
CIFAR-10 (VGG-11)	Multi-krum	0.40/0.64	0.43/0.35	0.43/0.72	0.40/0.54	0.41/0.34
	AFA	0.23/0.80	0.04/0.38	0.15/0.68	0.16/0.88	0.15/0.38
	FLTrust	0.21/0.87	0.22/0.89	0.23/0.89	0.21/0.88	0.22/0.48
	FLDetector	0.05/0.76	0.03/0.72	0.13/0.87	0.14/0.85	0.04/0.34
	FLAIR	0.38/0.37	0.37/0.88	0.46/0.88	0.46/0.69	0.48/0.37
	cosDefense	0.43/0.88	0.83/0.85	0.40/0.55	0.43/0.88	0.71/0.35
	FedRoLa-LASI	**0.15/0.93**	**0.13/0.96**	**0.06/0.87**	**0.08/0.90**	**0.04/0.83**
	FedRoLa-PCSI	**0.03/0.98**	**0.05/0.98**	**0.04/0.92**	**0.02/0.98**	**0.03/0.88**
CIFAR-100 (ResNet-18)	Multi-krum	0.51/0.60	0.54/0.56	0.54/0.58	0.51/0.19	0.48/0.19
	AFA	0.05/0.71	0.04/0.89	0.05/0.89	0.16/0.88	0.05/0.57
	FLTrust	0.18/0.59	0.13/0.83	0.13/0.89	0.14/0.89	0.12/0.59
	FLDetector	0.12/0.67	0.12/0.81	0.13/0.64	0.10/0.83	0.07/0.62
	FLAIR	0.03/0.60	0.03/0.58	0.02/0.32	0.03/0.56	0.02/0.31
	cosDefense	0.42/0.72	0.78/0.83	0.47/0.50	0.43/0.75	0.84/0.33
	FedRoLa-LASI	**0.03/0.91**	**0.19/0.90**	**0.15/0.92**	**0.15/0.92**	**0.06/0.81**
	FedRoLa-PCSI	**0.03/0.95**	**0.05/0.94**	**0.04/0.96**	**0.03/0.95**	**0.03/0.85**
Fashion MNIST (AlexNet)	Multi-krum	0.40/0.60	0.43/0.57	0.42/0.49	0.40/0.33	0.41/0.32
	AFA	0.23/0.38	0.14/0.62	0.17/0.59	0.20/0.50	0.17/0.38
	FLTrust	0.21/0.40	0.20/0.88	0.20/0.67	0.21/0.88	0.21/0.39
	FLDetector	0.23/0.61	0.12/0.72	0.08/0.36	0.11/0.89	0.03/0.36
	FLAIR	0.38/0.89	0.37/0.61	0.47/0.88	0.47/0.63	0.46/0.39
	cosDefense	0.53/0.89	0.89/0.85	0.57/0.39	0.43/0.89	0.85/0.39
	FedRoLa-LASI	**0.13/0.93**	**0.12/0.90**	**0.08/0.92**	**0.07/0.91**	**0.05/0.81**
	FedRoLa-PCSI	**0.03/0.98**	**0.05/0.91**	**0.03/0.98**	**0.04/0.95**	**0.05/0.88**
Shakespeare (LSTM)	Multi-krum	0.43/0.72	0.43/0.69	0.43/0.31	0.43/0.44	0.40/0.30
	AFA	0.25/0.51	0.09/0.79	0.19/0.88	0.17/0.53	0.17/0.51
	FLTrust	0.20/0.83	0.18/0.51	0.18/0.89	0.18/0.77	0.16/0.50
	FLDetector	0.28/0.51	0.16/0.44	0.09/0.51	0.10/0.85	0.03/0.43
	FLAIR	0.37/0.87	0.37/0.89	0.39/0.86	0.39/0.88	0.41/0.49
	cosDefense	0.49/0.55	0.79/0.51	0.37/0.88	0.39/0.51	0.80/0.51
	FedRoLa-LASI	**0.14/0.91**	**0.16/0.93**	**0.07/0.92**	**0.05/0.87**	**0.06/0.77**
	FedRoLa-PCSI	**0.03/0.98**	**0.05/0.96**	**0.03/0.93**	**0.03/0.98**	**0.04/0.86**
HARBox (DNN)	Multi-krum	0.38/0.80	0.38/0.27	0.30/0.78	0.28/0.40	0.36/0.27
	AFA	0.23/0.79	0.27/0.80	0.19/0.29	0.23/0.52	0.24/0.29
	FLTrust	0.35/0.89	0.33/0.87	0.33/0.89	0.35/0.63	0.35/0.40
	FLDetector	0.15/0.83	0.16/0.84	0.03/0.55	0.03/0.31	0.03/0.31
	FLAIR	0.46/0.80	0.45/0.81	0.45/0.61	0.48/0.33	0.48/0.33
	cosDefense	0.57/0.89	0.85/0.51	0.47/0.67	0.47/0.88	0.83/0.27
	FedRoLa-LASI	**0.11/0.95**	**0.15/0.94**	**0.06/0.92**	**0.04/0.85**	**0.03/0.74**
	FedRoLa-PCSI	**0.06/0.94**	**0.05/0.96**	**0.04/0.96**	**0.05/0.93**	**0.05/0.84**

Table 7: Comparisons of FPR and TPR with full knowledge.

by establishing a safe zone indicative of benign activity and then excluding any substantial deviations from this zone as threats.

B.1 Design Details

To effectively breach these security measures, we propose an attack method that closely imitates the properties of legitimate updates. This advanced attack leverages the nuances of cosine similarity, manipulating the updates in a discreet and selective manner. By doing so, it preserves a close resemblance to benign updates while strategically altering key components to inflict maximum damage without raising suspicion. Algorithm 4 delineates our innovative SimAttack, a meticulously crafted strategy targeting FL systems. In this context, $\mathcal{K}^{(t)}$ represents the knowledge about updates collected by the attacker. The optimization of this method begins with determining the initial value of the parameter λ in Algorithm 4. For this purpose, we introduce the following lemma:

Lemma 1. *Suppose that λ is the changing direction to craft gradients of m malicious clients based on the cosine similarity. For any given attack threshold τ, the value of λ satisfies*

$$\lambda = \frac{-z - \sqrt{z^2 - 4xy}}{2x}, \tag{9}$$

where $x = (\tilde{\mathbf{g}}^{(t)\intercal} \cdot \tilde{\mathbf{s}}(t))^2 - \tau^2 \|\tilde{\mathbf{g}}^{(t)}\|^2 \|\tilde{\mathbf{s}}(t)\|^2$, $y = (1-\tau^2)\|\tilde{\mathbf{g}}^{(t)}\|^4$, and $z = 2(\tau^2 - 1)\|\tilde{\mathbf{g}}^{(t)}\|^2 \tilde{\mathbf{g}}^{(t)\intercal} \cdot \tilde{\mathbf{s}}(t)$.

This lemma ensures that our modifications are subtle enough to avoid detection while challenging the defenses. Implementing

Algorithm 4 SimAttack Algorithm

```
Input: Knowledge $\mathbf{w}_i^{(t)}, \forall i \in \mathcal{K}^{(t)}$
1: for $t = 1, \cdots, T$ do
2:     Based on the received real global updates $\mathbf{g}^{(t-1)}$ and estimated global update $\hat{\mathbf{g}}^{(t-1)}$ at last round, implement algorithm 5 to dynamically adjust attack ratio $q \in (0, 1]$
3:     Perform local model updates $\mathbf{w}_i^{(t)}$ on all participants and get local update as $\mathbf{w}_i^{(t)} - \mathbf{w}^{(t-1)}$
4:     Use $\mathbf{g}_i^{(t)} = \mathbf{w}_i^{(t)} - \mathbf{w}^{(t-1)}, \forall i \in \mathcal{K}^{(t)}$ to estimate global update $\tilde{\mathbf{g}}^{(t)}$ and update direction $\mathbf{s}(t) = \text{sgn}(\tilde{\mathbf{g}}^{(t)})$
5:     Based on updates $\tilde{\mathbf{g}}^{(t)}$, selectively choose only a portion $q$ of elements from $\mathbf{s}(t)$ to form $\tilde{\mathbf{s}}(t)$
6:     Modify the global update as $\hat{\mathbf{g}}^{(t)} = \tilde{\mathbf{g}}^{(t)} - \lambda \tilde{\mathbf{s}}(t)$
7:     if Find suitable $\lambda$ then
8:         Utilize Lemma 1 to find the maximum value of $\lambda_0$ that guarantees cosine$(\hat{\mathbf{g}}(t), \tilde{\mathbf{g}}(t)) > 0$
9:         Based on the gathered information, calculate the cosine similarities $\{s_{i,j}\}_{i,j \in \mathcal{K}^{(t)}}$
10:        Determine the first 5% percentile value of $\{s_{i,j}\}_{i,j \in \mathcal{K}^{(t)}}$ and denote it as $Q_5$.
11:        Use $\lambda_0$ as initial value and employ the grid search method, find a suitable $\lambda$ that satisfies:
               $\text{argmax}_{\lambda} \min_{i \in \mathcal{K}^{(t)}} \text{cosine}(\hat{\mathbf{g}}^{(t)}, \mathbf{g}_i^{(t)}) \geq Q_5, \quad (7)$
           where $\hat{\mathbf{g}}^{(t)} = \tilde{\mathbf{g}}^{(t)} - \lambda \tilde{\mathbf{s}}(t)$.
12:    end if
13:    Get the global malicious as $\tilde{\mathbf{w}}^{(t)} := \mathbf{w}^{(t-1)} + \hat{\mathbf{g}}^{(t)}$. Then clients' local models are modified as follows:
           $\tilde{\mathbf{w}}_i^{(t)} = \mathbf{w}_i^{(t)} \odot [\tilde{\mathbf{s}}(t) = 0] + \tilde{\mathbf{w}}^{(t)} \odot [\tilde{\mathbf{s}}(t) \neq 0]. \quad (8)$
14: end for
```

Algorithm 5 Adjustment of q in SimAttack

```
Input: $\bar{\mathbf{s}}(t-1), \hat{\mathbf{g}}^{(t-1)}, \mathbf{g}^{(t-1)}$
1: Get similarity $s$ between $\hat{\mathbf{g}}^{(t-1)}$ and $\mathbf{g}^{(t-1)}$
2: if $s > 0$ then
3:     $q = q + 0.1$
4: else
5:     $q = q - 0.1$
6: end if
7: Limitation: $q \in (0, 1]$
8: Return $q$
```

Lemma 1 is crucial for determining an initial value for λ, a key parameter in our attack algorithm. Setting $\tau = 0.01$ provides a precise threshold, ensuring that deceptive updates, though minimal, are significant enough to subtly steer the global model in the adversary's favor. Our strategy, summarized in Algorithm 5, introduces how adjust attack ratio q dynamically. If malicious updates oppose real global updates, indicating easy detection, we decrease q. If they are not inverse and harder to detect, we increase q to compromise more parameters without raising suspicion.