

Article

FLGQM: Robust Federated Learning Based on Geometric and Qualitative Metrics

Shangdong Liu ¹ , Xi Xu ¹, Musen Wang ¹, Fei Wu ^{2,*}, Yimu Ji ^{1,*}, Chenxi Zhu ¹ and Qurui Zhang ¹

¹ School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; lsd@njupt.edu.cn (S.L.); 1021041208@njupt.edu.cn (X.X.); 1221045916@njupt.edu.cn (M.W.); b21030430@njupt.edu.cn (C.Z.); b20032121@njupt.edu.cn (Q.Z.)

² College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

* Correspondence: feiw@njupt.edu.cn (F.W.); jiym@njupt.edu.cn (Y.J.)

Abstract: Federated learning is a distributed learning method that seeks to train a shared global model by aggregating contributions from multiple clients. This method ensures that each client's local data are not shared with others. However, research has revealed that federated learning is vulnerable to poisoning attacks launched by compromised or malicious clients. Many defense mechanisms have been proposed to mitigate the impact of poisoning attacks, but there are still some limitations and challenges. The defense methods are either performing malicious model removal from the geometric perspective to measure the geometric direction of the model or adding an additional dataset to the server for verifying local models. The former is prone to failure when facing advanced poisoning attacks, while the latter goes against the original intention of federated learning as it requires an independent dataset; thus, both of these defense methods have some limitations. To solve the above problems, we propose a robust federated learning method based on geometric and qualitative metrics (FLGQM). Specifically, FLGQM aims to metricize local models in both geometric and qualitative aspects for comprehensive defense. Firstly, FLGQM evaluates all local models from both direction and size aspects based on similarity calculated by cosine and the Euclidean distance, which we refer to as geometric metrics. Next, we introduce a union client set to assess the quality of all local models by utilizing the union client's local dataset, referred to as quality metrics. By combining the results of these two metrics, FLGQM is able to use information from multiple views for accurate poisoning attack identification. We conducted experimental evaluations of FLGQM using the MNIST and CIFAR-10 datasets. The experimental results demonstrate that, under different kinds of poisoning attacks, FLGQM can achieve similar performance to FedAvg in non-adversarial environments. Therefore, FLGQM has better robustness and poisoning attack defense performance.

Keywords: federated learning; poisoning attack; robust defense



Citation: Liu, S.; Xu, X.; Wang, M.; Wu, F.; Ji, Y.; Zhu, C.; Zhang, Q. FLGQM: Robust Federated Learning Based on Geometric and Qualitative Metrics. *Appl. Sci.* **2024**, *14*, 351. <https://doi.org/10.3390/app14010351>

Academic Editor: Nuno Silva

Received: 13 November 2023

Revised: 10 December 2023

Accepted: 19 December 2023

Published: 30 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous increase in data scale and the growing importance of data privacy, new machine learning paradigms have been proposed, such as federated learning (FL) [1]. Federated learning allows multiple distributed clients to jointly train a global model without sharing their privacy data. Federated learning can summarize the three main steps of model distribution, local training, and aggregation. Compared to traditional machine learning, federated learning offers a compelling advantage by effectively safeguarding user privacy while maintaining the high accuracy of the global model. Consequently, it has garnered significant research interest in recent years.

However, the inherent distributed nature of federated learning renders it particularly susceptible to attacks orchestrated by malicious clients. Attackers can exploit this vulnerability by launching poisoning attacks through client control, thereby impeding the convergence of the global model. Finally, it will destroy the performance of federated

learning. Based on the attacker's objectives, poisoning attacks in federated learning can be categorized into two distinct types: untargeted poisoning attacks and targeted poisoning attacks. In an untargeted poisoning attack, the goal is to diminish the performance of the global model, resulting in low accuracy across all classes of test inputs. For example, in data poisoning attacks [2–5], malicious clients can compromise the integrity of the global model by poisoning local data. Another example is the local model poisoning attack [6–13], where the attacker compromises the integrity of the global model by uploading a locally corrupted model. The objective of a target poisoning attack [7,11,14] is to diminish the global model's accuracy, particularly for specific test inputs. For example, in backdoor attacks [14], attackers can compromise a subset of clients and exploit the compromised client to embed a backdoor within the global model, thereby posing a threat to the integrity of the entire global model.

Researchers have proposed a large number of robust federated learning methods [6,15–20] to address the above problems. Most of the works assume that malicious local models are geometrically distant from other benign models. Therefore, they would exclude malicious local models by detecting anomalies through statistical analysis before aggregation. Another approach is to maintain a clean dataset on the server and utilize it to perform anomaly detection on all local models. It is evident that the existing methods possess certain limitations. Firstly, in the first scheme, some advanced poisoning attacks have the capability to construct malicious local model updates that exhibit geometric similarity to benign local model updates, enabling them to evade existing defense mechanisms. Secondly, the alternative approach necessitates directly collecting clean datasets from individual clients, a practice that contradicts the privacy principles of federated learning.

To tackle the aforementioned challenges, we propose a novel and robust federated learning method called FLGQM. It measures local models from two aspects: quality and geometry, and assigns metric scores to different local models. Table 1 lists the main differences between the current mainstream defense approach and the approach we propose. As evident from Table 1, the current mainstream defense methods either detect malicious attacks geometrically or utilize a small, clean dataset to evaluate the quality of a local model. It is evident that certain methods fail to effectively resist all attacks using only Euclidean distance or the median method, and they are also prone to failure under non-IID data [21]. Moreover, the use of a centralized dataset contradicts the principle of federated learning, which aims to avoid data collection. In contrast, our approach leverages cosine similarity and Euclidean distance to defend against malicious attacks from a geometric perspective. Additionally, we introduce a distributed set of union clients to assess the quality of the local model, eliminating the need for a central dataset. Thus, our method demonstrates significant advantages in defending against malicious attacks from multiple angles. The main innovations of this work are as follows:

- (1) We propose a novel and robust federated learning method that considers each uploaded local model from both geometric and quality aspects. It can use information from multiple views to perform a robust defense.
- (2) We propose a geometric metric mechanism that considers each local model from a geometric perspective. The similarity is calculated by cosine and the Euclidean distance to constrain the direction and size of all local models.
- (3) We propose a quality metric mechanism that introduces a union client set to send all local models to the union client for evaluation and finally measures the quality of all local models based on the trimmed mean.
- (4) We conducted a comprehensive experimental evaluation of FLGQM on MNIST and CIFAR-10. The results indicate that the algorithm outperforms existing robust federated learning methods in terms of its effectiveness against state-of-the-art poisoning attacks.

Table 1. The current defenses against poisoning attacks.

Defense	Technique	Geometric	Qualitative	Central Dataset	Non-IID Data
Krum/Multi-Krum	Euclidean distance	✓	✗	Not needed	✗
Bulyan	Krum + trimmed median	✓	✗	Not needed	✗
RFA	Geometric median	✓	✗	Not needed	✗
FoolsGold	Contribution similarity	✓	✗	Not needed	✓
Zeno	Calculates score by clean small dataset	✗	✓	Needed	✓
Fltrust	Cosine similarity + Clean small dataset	✓	✓	Needed	✓
MAB-RFL	Similarity + Graph theory	✓	✗	Not needed	✓
FLGQM (ours)	Cosine similarity + Euclidean distance (+ Distributed score calculation)	✓	✓	Not needed	✓

2. Materials and Methods

2.1. Related Work

2.1.1. Federated Learning

There is a set of nodes/clients $C = c_1, \dots, c_k, \dots, c_K$ in federated learning, where each client has its own data $\mathbb{D}_k = \{(a_1, b_1), \dots, (a_i, b_i), \dots, (a_{n_k}, b_{n_k})\}$, where a_i is the i th data, and b_i represents the true label. $|\mathbb{D}_k| = n_k$. n_k is the number of training data owned by c_k . $n = \sum_{k \in K} n_k$ represents the cumulative number of samples owned by all clients, while the server is denoted as S . Specifically, all clients train a classifier $f(w)$, where w is the global model obtained by aggregation, and this model is expected to have good results on the test set D_{test} . The objective of the server is as follows.

$$\min_w l(w) = \sum_{k=1}^K \frac{n_k}{n} l_k(w) \quad (1)$$

where $l_k(w) = \frac{1}{n_k} \sum_{i \in \mathbb{D}_k} l_i(w)$. $l_i(\theta) = l(b_i, f_w(a_i))$ is the loss predicted with model parameter w for data (a_i, b_i) .

2.1.2. Poisoning Attacks in Federated Learning

In recent studies [15], it has been revealed that non-adversarial federated learning is highly susceptible to poisoning attacks initiated by malicious clients. These attacks can result in significant performance degradation of the global model. In summarizing, we categorize these poisoning attacks based on the objectives and capability of the malicious poisoning adversary. Firstly, we can categorize poisoning attacks into targeted [7,11,14] and untargeted attacks [6,8,9,12,22] based on the objectives of the adversary. The untargeted attack seeks to maximize the global model's inaccuracy across any test input, whereas the targeted attack seeks to minimize the accuracy specifically on certain test inputs without significantly impacting the accuracy of other test inputs.

Secondly, we can categorize poisoning attacks into two main types: data poisoning attacks [2–5] and model poisoning attacks [6–13], depending on the adversary's capabilities. Data poisoning attacks leverage corrupted data of the client to indirectly manipulate the gradient or model parameters uploaded by the client. Model poisoning attacks directly change the gradient or model parameters of a malicious client to manipulate the global model. For example, Fang et al. [8] formulated an attack to change the local model of the malignant client. Its objective is to maximize the divergence between the global model after the attack and the pre-attack global model. Shejwalkar et al. [13] introduced a comprehensive framework for model poisoning attacks in federated learning, which enables a more comprehensive model poisoning attack.

2.1.3. Existing Robust Federated Learning

Many defense mechanisms have been proposed to relieve the effects of poisoning attacks. They are mainly divided into two ideas. One type of defense method is to assume that all malicious models are geometrically far from benign models. For example, In Krum [15], the Euclidean distance is employed to score the local models, and subsequently, the highest-rated model is chosen as the global model. In Fast Aggregation algorithm against Byzantine Attacks (FABA) [23], local gradients that deviate significantly from the average gradient are discarded. However, both of these methods necessitate prior knowledge of malicious clients' number, which imposes certain limitations on their applicability. To overcome this limitation, FoolsGold [24] utilize cosine similarity to assign a low score to malicious clients, thereby minimizing malicious clients' impact. Wan et al. [25] propose MAB-RFL, which employs graph theory to eliminate models that exhibit excessive similarity and then extracts key parameters from the models using principal component analysis as a way to distinguish benign and malicious clients. The median and trimmed mean [18] compute the trimmed mean and median of each dimension of the local model to circumvent models from malicious clients. Robust Federated Aggregation (RFA) [26] employs alternating minimization methods to calculate the geometric median. Bulyan [6] trims the mean on the basis of Multi-Krum. Adaptive Federated Aggregation (AFA) [27] first calculates the similarity between the local model and the global model. It then identifies and removes underperforming local models according to the mean and median of these similarities. Xu et al. [28] proposed SignGuard, which first processes the received gradients to generate relevant statistics related to similarity, sign, and magnitude. The malicious gradients are then eliminated by the filters utilized collaboratively before final aggregation. Geng et al. [29] proposed FLEST. It proposes a trust synthesis mechanism that aggregates confidence scores and trust scores into a composite trust score (STS) through dynamic trust ratios, and uses the score as a weight for updating the aggregated local parameters. Wang et al. [30] proposed FLForest. Its primary focus is to assess the occurrence of an attack according to the magnitude of change in the global model update, accomplished by incorporating Isolation Forest.

Another defense method involves maintaining a clean, small dataset on the server to filter out poorly performing local models. For example, Zeno [20] has a small dataset on the server and calculates a score for local models. Then, these scores are aggregated into a global model. Cao et al. [31] proposed a distributed gradient algorithm that reduces the effect of malicious clients by computing noise gradients using a clean dataset. FLTrust [19] maintains a small training dataset on the server, trains a bootstrap model based on this dataset, and then utilizes cosine similarity between this model and the local model as a score for the local model. Rodr guez-Barroso et al. [32] proposed Dynamic Defense Against Byzantine Attacks (DDaBA). The main idea is to judge the quality of individual clients by the clean dataset on the server and then score each client with a linguistic quantifier. Cao et al. [33] proposed FLCert, which first groups all the clients and trains a global model separately. The method also needs to keep a clean dataset on the server to validate all the local models, and then an optimal global model is selected by a voting mechanism.

Existing robust federated learning methods have a significant limitation: some poisoning attacks can simulate malicious models that are geometrically similar to benign models, thereby breaking this defense mechanism. At the same time, the method of keeping small datasets on the server is problematic as it goes against the idea of federated learning. In contrast, this paper proposes a robust federated learning method based on geometric and qualitative metrics to defend against advanced poisoning attacks and avoid a centralized dataset.

2.2. FLGQM

2.2.1. Overview

In this paper, we design a new federated learning method: robust federated learning based on geometric and qualitative metrics. The adversary can control the malicious client's

model in an arbitrary manner. When we calculate the global model, we consider two aspects, the intuitive model quality and the geometric direction and size of the uploaded models, and use them to assign a metric score (MS) to the local model. A high metric score represents better local models.

Specifically, we first consider a scenario where the adversary has arbitrary control over the geometric direction and size of the local model and therefore performs geometric metrics on all uploaded local models from this perspective. Formally, we use similarity based on cosine and the Euclidean distance. To guide the global model's direction, we ensure that each global iteration aligns with the trajectory of the previous round's global model. First, the server retains a collection of models uploaded by the clients who participated in the previous round of training. Considering that the uploaded model in the current round of benign clients cannot deviate from the uploaded model in the previous round, we quantify the similarity between the uploaded model in the current round and the model uploaded in the previous round using cosine similarity. This process allows us to identify and eliminate some of the malicious clients. Next, we utilize cosine similarity to quantify the alignment in direction between the local model and the global model from the previous round. We then identify and exclude malicious clients whose behavior deviates significantly from the global model using the ReLU function. Subsequently, we quantify the similarity in size between the local model and the global model from the previous round using Euclidean distance.

The similarity based on Euclidean distance can identify the difference in malicious models and then assign geometric metric scores (GMSs) to different clients. Also, considering that certain advanced poisoning attacks can geometrically make malicious updates similar to benign ones, we perform quality metrics on all uploaded local models from this perspective. Specifically, we randomly select some of the clients that do not participate in this round of federated learning as union clients. They verify the models uploaded by all local clients separately using their own local data, evaluate the models for different clients based on the trimmed mean, and then evaluate the model quality for all uploaded models as a way to assign quality metric scores (QMSs) to different clients. Finally, the global model is updated by taking a weighted average of the combined quality and geometric metric scores.

2.2.2. Aggregation Rule

Our proposed FLGQM is shown in Figure 1. It is considered in two aspects simultaneously: model quality and model geometry aspects. Firstly, the server receives all the local models. It will use cosine similarity and Euclidean distance to perform the geometric metric on each local model to obtain the geometric metric score (GMS). Then, the server will send the received local models down to the union client set for quality metrics to obtain the quality metric score (QMS), and finally, a new round of global models is obtained by weighted average of the two scores. Next, we describe each step of the algorithm in detail.

Geometric Metric Score (GMS). The adversary can arbitrarily control the direction and size of the malicious client's local model. Therefore, we consider constraining the behavior of all uploaded local models prior to aggregation. The whole process is shown in Figure 2. For each uploaded model weight, we vectorize it to the vector space, and it is easy to know that each model corresponds to a vector in the vector space, i.e., it has a size and direction. Obviously, the more similar two vectors are, the more similar their corresponding sizes and directions should be. Based on this, we distinguish benign local models from malicious local models. A malicious local model should be far from a benign model in terms of direction and size. And we know that cosine similarity can be used to determine whether two vectors are similar in direction and Euclidean distance can be used to determine whether two vectors are similar in size, so we use cosine similarity and Euclidean distance for geometric metrics. The specific steps of the algorithm will be described in detail below. Under the federal learning framework, we assume that the current set of clients selected to participate in this round of training is $C^t = c_1, \dots, c_p, \dots, c_P \subseteq C$.

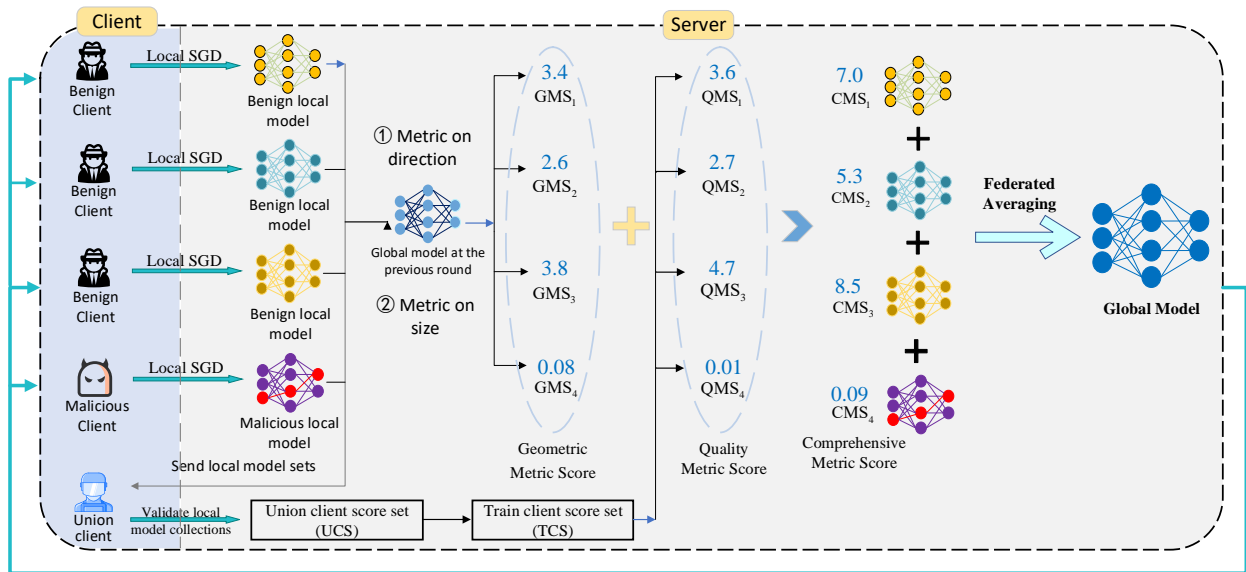


Figure 1. Robust federated learning method based on geometric and qualitative metrics.

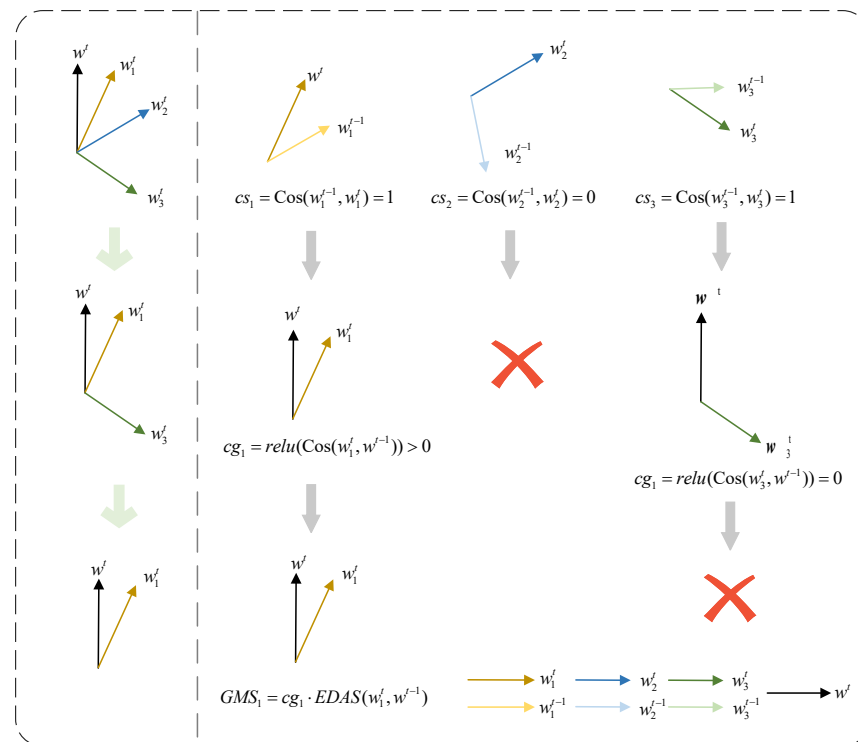


Figure 2. Geometric metrics score calculation process. The right is the detailed GMS calculation process, and the left is the spatial state of the model after each calculation.

First, we constrain the direction of the local model using cosine similarity, which is reflected in two similarity measures. Here, we use the p th client, i.e., $c_p \in C^t$ as an example. The first measure is a cosine similarity measure between the local model uploaded in this round w_p^t and its local model uploaded in the previous round w_p^{t-1} , and the second measure is a cosine similarity measure between the local model uploaded in this round w_p^t and the global model in the previous round w^{t-1} . Based on the results obtained from the first measure, we will reject the local models with negative similarity, i.e., rejecting their participation in this aggregation because their models in this upload are too different from their models in the last upload. Based on the results obtained from the second measure, we

further eliminate the local models with negative similarity because their uploaded model in this current round w_p^t is too different from the global model w^{t-1} . Then, we score the remaining local models using the second measure.

Specifically, the server stores a model set $\mathcal{V} = \{w_1^{t-1}, \dots, w_p^{t-1}, \dots, w_P^{t-1}\}$ containing the local models for round $t - 1$ of the clients that participate in training. Considering the arbitrary nature of the malicious local model, we first use the cosine similarity to detect the clients that have the model w_p^t far removed from the model w_p^{t-1} in the direction, and then eliminate the detected clients. This process is formally as follows.

$$cs_p = \begin{cases} 0, & \text{Cos}(w_p^{t-1}, w_p^t) < 0, \\ 1, & \text{Cos}(w_p^{t-1}, w_p^t) \geq 0 \end{cases} \quad (2)$$

where cs_p indicates whether to reject the model uploaded by the c_p to participate in aggregation, with 0 representing rejection and 1 representing no rejection. $\text{Cos}(w_p^{t-1}, w_p^t) = \frac{\langle w_p^{t-1}, w_p^t \rangle}{\|w_p^{t-1}\| \|w_p^t\|}$. The set of local models after this round of exclusion is formalized as $C^{t_1} = c_1, \dots, c_l, \dots, c_L \subseteq C^t, L \leq P$.

Based on the set C^{t_1} , we further use cosine similarity to quantify the similarity of direction between the local models $w_1^t, \dots, w_l^t, \dots, w_L^t$ and the global model w^{t-1} , where a bigger similarity represents a better local model. It is evident that, when the cosine similarity between the local model and the global model is negative, indicating a contrary direction, the corresponding client can be identified as a malicious client. In such cases, the client should be excluded from the aggregation process. Formally, we use the ReLU function commonly used in deep learning to eliminate malicious clients with negative cosine similarity between the local model and the global model. Then, we use the cosine similarity to score the remaining local models with positive similarity, thus constraining the local model direction as follows.

$$cg_l = \text{ReLU}(\text{Cos}(w_l^t, w^{t-1})) \quad (3)$$

where cg_l denotes the metric of w_l^t from the aspect of direction by the c_l . For $\text{ReLU}(x)$, ReLU is equal to 0 when $x < 0$, and ReLU is equal to x when $x \geq 0$. The set of local models after this round of elimination is formally $C^{t_2} = \{c_1, \dots, c_m, \dots, c_M\} \subseteq C^{t_1}, M \leq L$.

Since an attacker can also control the global model by uploading damaged local models from a malicious client, it is insufficient to solely constrain the local model from a directional perspective. The Euclidean distance can be used as a measure of the distance between two models. The Euclidean distance is defined as $ED = \|\mathbf{a} - \mathbf{b}\|^{1/2}$. Therefore, we propose a difference amplifying similarity based on Euclidean distance, i.e., $EDAS(\mathbf{a}, \mathbf{b}) = \frac{1}{1 + [\|\mathbf{a} - \mathbf{b}\|^\mu - \omega \|\mathbf{a} - \mathbf{b}\|^\mu]^\gamma}$. We introduce three parameters μ, ω, γ and make the similarity range from 0 to 1; μ, ω are used to amplify the more subtle differences between the local model and the global model. γ is used to control the weight of the difference. The geometric metric scores that specifically constrain the local model direction and size are as follows.

$$GMS_m = cg_m \cdot EDAS(w_m^t, w^{t-1}) \quad (4)$$

where GMS_m denotes the geometric metric score of the model uploaded by the c_m . Our proposed method constrains the local models in terms of direction, eliminating the local models with reverse backwardness while giving higher weights to the local models that are close to the global model. And we constrain the local model in terms of size. The farther the local model is from the global model in size, the lower the score it receives, thus ensuring that more participants are from non-malicious clients when aggregating.

By doing the above, our method has a better geometric advantage over the existing geometry-based defense methods. Through the introduction and related work, it is easy to see that the current geometry-based defense methods are not comprehensive in identifying

poisoning attacks from one perspective alone. In contrast, in our approach, we first eliminate all the directionally anomalous malicious clients from the direction using cosine similarity. Then, we propose a difference amplifying similarity based on Euclidean distance to eliminate all malicious clients that are principle benign clients in terms of size. Thus, our geometric metric approach has a greater advantage.

Quality Metric Score (QMS). Here, we consider that an attacker can exploit more advanced attacks such that the model uploaded by the malicious client has the same geometric direction and size as the benign client's updated model to affect the global model. Thus, defending from a geometric perspective alone is not enough. Therefore, we consider a quality metric for the uploaded local model so as to remove the effect of this kind of malicious client on the global model. Existing defense methods typically maintain a small, clean dataset on the server for quality assessment. The datasets under this approach are usually particularly small and not yet fully representative of the entire data distribution, while also violating the highly problematic principle that federated learning does not collect data. Therefore, we consider delegating server privileges to each client in order to ensure a reliable measure of the quality of the local model, eliminating the influence of the central dataset for distributed quality metrics. Specifically, a certain number of clients are selected to form a union set to participate in the evaluation of the uploaded local models in each round. Then, the generated quality measure score is assigned to the uploaded local models based on the trimmed mean. Figure 3 shows an example of how to calculate the quality metric score. The approximate process is that there are a total of eight training clients and four union clients. After each training client uploads the model weight, the server will send the weights to the union clients, and each union client will use its own local dataset to score the model weights for all the training clients. Then, the server will obtain the final score by trimming the mean for each model weight after collecting all the scores.

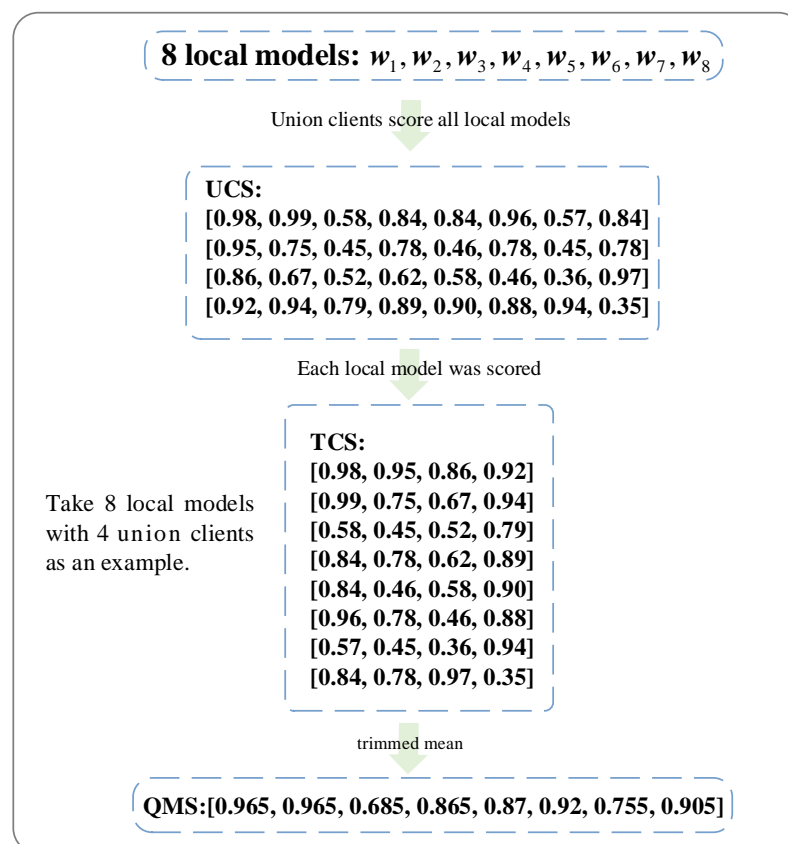


Figure 3. An example of calculating the quality metric score.

In particular, we randomly elect a union client set $U^t = u_1^t, \dots, u_h^t, \dots, u_H^t$ with $H \leq K - P$ local clients from the set $C^{no in} = C - C^t$, which do not participate in round t . After the geometric metric, the server sends the local models $W_{goodInGeo}^t = w_1^t, \dots, w_m^t, \dots, w_M^t$ uploaded by the client in C^{t2} where some malicious clients have been excluded to the union set U^t for quality evaluation. After a union client u_h^t receives the local models $W_{goodInGeo}^t$, it uses its local dataset \mathbb{D}_h to score that local model set for evaluation. Then, the union client u_h^t will form a set, called the union client score set (UCS), which is described as follows.

$$UCS_h = [sv_h^1, \dots, sv_h^m, \dots, sv_h^M] \quad (5)$$

where sv_h^m represents the evaluation score of the union client u_h^t for the local model w_m^t . After the union client finishes evaluating all local models, it sends the evaluation results UCS_h back to the server.

After receiving the evaluation results from all union clients, the server evaluates the quality of the local models. Specifically, each model w_m^t will have a set called a train client score set (TCS).

$$TCS_m = [sv_1^m, \dots, sv_h^m, \dots, sv_H^m] \quad (6)$$

After the above process, each local model will have H scores, and then each local model will have a single score based on the trimmed mean.

$$sv^m = \frac{1}{(1 - 2\beta)H} \sum_{x \in TCS_m^\beta} x \quad (7)$$

where sv^m is the final score of the local model w_m^t by the H union clients. β is the percentage factor that will be eliminated. $TCS_m^\beta \subseteq TCS_m$ denotes the set that eliminates the largest scores with the number of βm and the smallest scores with the number of βm .

Then, according to [34], we quantified the local model quality, and the quality metric score of the i th local model is described as follows.

$$QMS_m = \frac{\psi}{\log\left(\frac{1}{sv^m}\right)} \quad (8)$$

where ψ is the number of local model iterations. Our quality metrics have a good advantage over current defense approaches that directly utilize small datasets from the server for quality metrics of local models. Firstly, we abandon the practice of centralizing the data in the server and delegate the quality metrics to the clients, which better satisfies the principle of federated learning. Secondly, our quality metrics are distributed across multiple clients, which provides better fault tolerance and accuracy than utilizing a central dataset for evaluation.

Aggregation. We first define the composite metric score (CMS) of the local model w_m^t , which is a combination of the geometry and quality metric scores.

$$CMS_m = GMS_m + QMS_m \quad (9)$$

Then, we calculate the global model by weighted averaging the local model, where the weights are determined based on the composite metric scores.

$$w^t = \frac{1}{\sum_{i=1}^M CMS_i} \sum_{i=1}^M CMS_i \cdot w_i^t \quad (10)$$

Algorithm 1 describes the complete FLGQM algorithm. Through the above process, we can obtain a more robust global model.

Algorithm 1 FLGQM.

```

1: Input: Clients' total number  $K$ , the client set  $C$ , train clients' total number  $P$ , union
   clients' total number  $H$ , communication rounds' total number  $T$ , local epochs' total
   number  $E$ , the minibatch size  $b$ .
2: Onput: Global model  $w$ 
3: Server executes:
4: for  $t \in \{1, 2, \dots, T\}$  do
5:   // The server randomly selects clients and distributes the gobal model.
6:   client( $C^t$ ) executes:
7:   for  $p \in \{1, 2, \dots, P\}$  in paralld do
8:      $w_p^t \leftarrow \text{ClientUpdate}(p, w^{t-1})$ 
9:     send  $w_p^t$  to server
10:  end
11:  for  $p \in \{1, 2, \dots, P\}$  do
12:     $cs_p = \begin{cases} 0, & \text{Cos}(w_p^{t-1}, w_p^t) < 0, \\ 1, & \text{Cos}(w_p^{t-1}, w_p^t) \geq 0 \end{cases}$ 
13:  end
14:  // Reject models with a value of 0. The remaining client set is  $C^{t_1} =$ 
    $c_1, \dots, c_l, \dots, c_L \subseteq C^t, L \leq P$ .
15:  for  $l \in \{1, 2, \dots, L\}$  do
16:     $cg_l = \text{ReLU}(\text{Cos}(w_l^t, w^{t-1}))$ 
17:  end
18:  // Reject models with a value of 0. The remaining client set is  $C^{t_2} =$ 
    $\{c_1, \dots, c_m, \dots, c_M\} \subseteq C^{t_1}, M \leq L$ .
19:  for  $m \in \{1, 2, \dots, M\}$  do
20:     $GMS_m^t = cg_m \cdot \text{EDAS}(w_m^t, w^{t-1})$ 
21:  end
22:  // The server randomly samples  $H$  clients from  $C^{\text{noin}} = C - C^t$ , and forms a union
   set  $U^t$ .
23:  // The server sends the set  $W_{\text{goodInGeo}} = w_1^t, \dots, w_m^t, \dots, w_M^t$  to union clients.
24:  client( $U^t$ ) executes:
25:  for  $h \in \{1, 2, \dots, H\}$  in paralld do
26:    for  $m \in \{1, 2, \dots, M\}$  do
27:       $sv_h^m \leftarrow \text{accuracy in dataset } \mathbb{D}_h$ 
28:    end
29:     $UCS_h^t \leftarrow [sv_h^1, \dots, sv_h^m, \dots, sv_h^M]$ 
30:    send  $UCS_h^t$  to server
31:  end
32:  for  $m \in \{1, 2, \dots, M\}$  do
33:     $TCS_m^t = [sv_1^m, \dots, sv_h^m, \dots, sv_H^m]$ 
34:     $sv^m = \frac{1}{(1-2\beta)H} \sum_{x \in TCS_m^t} x$ 
35:     $QMS_m^t = \frac{\psi}{\log(\frac{1}{sv^m})}$ 
36:     $MS_m^t = MMS_m^t + QMS_m^t$ 
37:  end
38:  aggregation  $w^t = \frac{1}{\sum_{i=1}^M MS_i} \sum_{i=1}^M MS_i \cdot w_i^t$ 
39: end

```

3. Results and Discussion

3.1. Experimental Setup

3.1.1. Datasets

In our experiments, we utilize two publicly available classification datasets.

MINST [35]: A widely used dataset in machine learning, it serves as the standard benchmark for digit recognition tasks. It comprises 60,000 training images and 10,000 test images, with each grayscale image representing a handwritten digit (ranging from 0 to 9) in a 28×28 pixel format.

CIFAR-10 [36]: Curated by the Canadian Institute for Advanced Research (CIFAR), it is a widely used dataset in machine learning. It consists of 60,000 32×32 pixel color images categorized into 10 distinct classes. These classes represent objects such as airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

To simulate the distribution of non-IID data among clients, we follow previous work [5] and assign a random dominant label l to each client, i.e., the number of labels l dominates within the client and the number of other labels is uniformly distributed. Specifically, we group all clients according to the number of classes within the dataset, i.e., if there are Y classes in the dataset, the clients are divided into Y groups. Then, for label l , it is assigned to group l with a probability of q ; otherwise, it is assigned to other groups.

Each client within the same group receives an equal and balanced distribution of the data. q controls differences in the distribution of training data across clients. When $q = 1/Y$, it indicates that the client's local training data follow an independent and identically distributed (IID) pattern. Conversely, if q takes any other value, the client's local training data are considered non-IID. Moreover, as the value of q increases, the degree of non-IIDness in the client's local training data also escalates. In our experiments, we deliberately set q to 0.5 for both MNIST and CIFAR-10, resulting in a higher degree of non-IID characteristics in the data.

3.1.2. Poisoning Attack

In our experiment, for data poisoning attacks, we focus on the label flipping attack. Regarding local model poisoning attacks, we focus on the LIE attack, Fang attack, and AGRT attack.

Label flipping attack [37]: For each sample of each malicious client, we cyclically shift the corresponding label l to $(l + 1) \bmod Y$, where Y represents the total number of labels (e.g., $Y = 10$ in the MNIST dataset) $l \in 1, 2, \dots, Y$.

LIE attack [9]: The Little Is Enough (LIE) attack adds a little noise on the model that is the average of all uploaded local models in the non-adversarial setting. The perturbation used in the attack is carefully chosen to be large enough to significantly impact the global model but small enough to avoid detection by the Byzantine Robust Aggregation algorithm.

Fang attack [8]: The malicious client crafts poisoned local model updates that are intentionally opposite to the benign updates, allowing them to bypass the defense mechanism of Krum.

AGRT attack [13]: The goal of the aggregation algorithm tailored (AGRT) attack is to maximize the effect on the global model in an optimization problem during each round of FL, i.e., to find the best malicious update that is not easily detected by the defense mechanism, while also achieving the maximum attack effect. Formally, the attacker builds a malicious update using the following optimization problem.

$$\begin{aligned} \arg \min_{\gamma} & \|\nabla^b - A(\nabla_{i \in [m]}^m \cup \nabla_{i \in [m+1, n]})\|_2 \\ \nabla_{i \in [m]}^m &= \nabla^b + \gamma \nabla^p; \nabla^b = \text{FedAvg}(\nabla_{i \in [n]}) \end{aligned} \quad (11)$$

where A is the defense method deployed by the server (e.g., Krum, etc.). $\nabla_{i \in [n]}$ is the benign update known to the attacker. FedAvg is a benign aggregation method that averages the updates from all clients. ∇^p is a perturbation variable. γ is a scaling factor and the

AGRT attack, i.e., solving for the optimal γ value. $\nabla_{i \in [m]}^m$ is then the final malicious update obtained.

3.1.3. Baseline Aggregation Rules

We consider the following aggregation rules as our baseline for comparison.

FedAvg [1]: The new global model is obtained by calculating the average of all local models.

Median [18]: The median directly takes the median of the coordinates of each dimension of all local models to obtain the new global model.

Krum [15]: In Krum, the Euclidean distance is employed to score the local models, and subsequently, the highest-rated model is chosen as the global model. For the i th client, assuming the existence of f malicious clients, the local model's score can be determined as follows.

$$s_i = \sum_{w_j \in \Gamma_{i,K-f-2}} \|w_j - w_i\|_2^2 \quad (12)$$

where $\Gamma_{i,K-f-2}$ is the set of $K - f - 2$ local models, and this $K - f - 2$ local model is the $K - f - 2$ model with the smallest Euclidean distance from w_i .

Trimmed Mean [18]: The trimmed mean is an aggregation rule that takes into account each model parameter and operates based on coordinates. For every model parameter, the server gathers all values from local models and arranges them in ascending order. Subsequently, it excludes the largest and smallest \mathcal{K} values, computes their average, and utilizes this average as the corresponding parameter value in the global model.

3.1.4. Performance Metrics and Federated Learning System Setup

Given that this work primarily centers around countering data poisoning attacks and model poisoning attacks, both of which intend to undermine the global classification accuracy of the model, we have chosen the model's accuracy as our evaluation metric. A higher accuracy for the global model signifies the effectiveness of the corresponding defense method. By default, in the case of MNIST, we consider a federated learning (FL) system comprising 100 clients. For CIFAR-10, the FL system consists of 40 clients. Unless specifically stated, we assume that 20% of the clients in each dataset are malicious.

3.1.5. Global Model

In the case of MNIST, we employ a straightforward CNN architecture comprising two convolutional layers and two fully connected layers as the global model. For CIFAR-10, our choice for the global model is ResNet18.

3.1.6. Federated Learning Method Parameter Setting

We compared FLGQM with FedAvg, median, Krum, and trimmed mean. Among them, FedAvg is the most popular method in non-adversarial environments. The median, Krum, and trimmed mean are three Byzantine robust federated learning methods. These methods all satisfy the general framework. Hence, all of them share the following parameters: K , τ , R_l , R_g , b , and β . The specific definitions and explanations of these parameters can be found in Table 2. Specifically, based on previous work [16], we set $\tau = K$, i.e., all clients participate in the federated learning process in each round. Also, we set the local iteration round $R_l = 5$ and the global iteration round $R_g = 300$. We set $b = 16$ for MNIST and $b = 64$ for CIFAR-10.

In addition to the common parameters mentioned above, different methods have different additional parameters. Krum has the parameter f and the trimmed mean has the parameter \mathcal{K} , which represents the upper limit of malicious clients. We set $f = \mathcal{K} = \mathcal{M}$, i.e., giving them enough advantage in both algorithms to let them know the number of all malicious clients. For the FLGQM method proposed in this paper, we set $\tau_a = 0.8$ and $\tau_b = 0.2$. In the geometric metric score stage, for MNIST, we set the hyperparameters to $\mu = 1$, $\omega = 0$, $\gamma = 2$, and for CIFAR-10, we set the hyperparameters to $\mu = 2$, $\omega = 1$, $\gamma = 4$.

Table 2. Federated learning parameter settings in our experiment.

	Explanation	MNIST	CIFAR-10
K	Number of clients	100	40
τ	Number of clients selected per global round	K	
τ_d	Proportion of clients selected as training clients each global round	0.8	
τ_u	Proportion of clients selected as union clients each global round	0.2	
E_l	Local epoch	5	
E_g	Global epoch	300	
b	Batch size	16	64
β	Learning rate	0.01	0.001
\mathcal{M}	Number of malicious clients	20	8
f	Parameter of Krum [11]	\mathcal{M}	
\mathcal{K}	Parameter of trimmed mean [15]	\mathcal{M}	

3.2. Experimental Results

3.2.1. FLGQM Can Achieve Two Defensive Goals.

Table 3 shows the accuracy of different federated learning algorithms on two datasets under different attacks. The results show that FLGQM can achieve two defense goals: fidelity and robustness.

Fidelity. As evident from Table 3, FLGQM achieves fidelity as the accuracy in the non-adversarial setting is comparable to the baseline (FedAvg) on both datasets.

Table 3. Accuracy of compared federated learning algorithms on MNIST and CIFAR-10 with different attacks.

Dataset	Attacks	FedAvg	Median	Trim_mean	Krum	FLTrust	MAB-RFL	FLGQM
MNIST	No attack	98.51	98.25	98.22	95.01	97.84	97.95	98.39
	LF attack	-	97.79	97.41	93.51	97.40	96.60	98.03
	LIE attack	-	96.25	88.62	94.49	97.14	97.44	98.05
	Fang attack	-	96.17	95.64	94.75	97.10	97.45	97.97
	AGRT attack	-	95.18	95.84	74.19	97.03	97.14	97.63
CIFAR-10	No attack	73.68	72.83	72.58	66.36	73.29	73.70	73.55
	LF attack	-	69.59	69.35	57.63	73.21	73.66	73.53
	LIE attack	-	48.38	35.77	41.68	72.95	72.62	73.30
	Fang attack	-	58.73	66.57	42.43	73.19	72.18	73.39
	AGRT attack	-	23.06	53.77	10.00	72.52	70.55	72.63

However, the existing robust federated learning methods are less accurate in the absence of attacks. For example, the accuracy of median, Trim_mean, Krum, FLTrust, and MAB-RFL on MNIST differed from FedAvg by 0.26%, 0.29%, 3.50%, 0.67%, and 0.56%, respectively. However, the accuracy of FLGQM is close to that of FedAvg, with only 0.12% difference. Compared with FedAvg, the accuracy of median, Trim_mean, and Krum decreased by 0.20%, 0.45%, and 6.67%, with Krum possessing a lower accuracy. And the accuracy of FLGQM differs from FedAvg by only 0.13%. The results show that FLGQM has high accuracy, i.e., has better fidelity, than the existing robust federated learning method in a non-adversarial setting. This is due to the fact that the current FL method discards some of the parameter updates while aggregating local parameter updates and considers a single perspective, so the test accuracy will be lower. In contrast, our approach takes into account model updates from two perspectives and is able to amplify the more subtle

differences between benign clients and malicious clients, allowing us to effectively leverage all available local model information.

Robustness. As can be seen from Table 3, FLGQM achieves robustness as its accuracy under all attacks has very little change compared to the baseline (FedAvg) on both datasets.

Specifically, FLGQM can have comparable accuracy compared to FedAvg on both datasets, with a maximum difference of only 0.88%. Meanwhile, other existing robust federated learning methods possess worse accuracy. Especially in the case of untargeted attacks, their performance drops sharply. For example, on the MNIST dataset, Trim_mean's accuracy decreases by 9.6% under LIE attack and Krum's accuracy decreases by 22.02% under AGRT attack. Median and Krum fail under AGRT attack on the CIFAR-10 dataset. On the CIFAR-10 dataset, median, Trim_mean, and Krum all show varying degrees of performance degradation under other attacks. In comparison to the latest defense methods, FLTrust and MAB-FL, our method exhibits slightly lower accuracy than MAB-FL solely on CIFAR-10 under LF attack. However, in all other scenarios, our method demonstrates higher accuracy than both FLTrust and MAB-FL. This highlights the relative stability of our defense method and underscores its advantages over existing defense methods. Therefore, it can be seen that FLGQM achieves the goal of robustness because FLGQM takes more factors into account and can minimize the impact caused by malicious clients from different dimensions.

3.2.2. The Variant of FLGQM

FLGQM has two key features: it uses cosine similarity and Euclidean similarity to constrain the size and direction of the local model, and it uses union clients to perform quality metrics on the local model. Considering that the union client cannot run alone, we consider a variant of FLGQM called FLGQM-NoQMS. In this variant, we remove the union client and only consider the effect of geometric metric scores.

Table 4 shows the accuracy of this variant with median, Trim_mean, Krum, and FLGQM under different attack methods. It is evident that the accuracy of FLGQM-NoQMS decreases relative to FLGQM, indicating that the union client has a role in improving the robustness of federated learning. By incorporating quality metrics alongside geometric metrics, we can effectively counteract poisoning attacks that may go undetected if relying solely on geometric metrics. Through distributed quality assessment, we gain additional insights into the local models, enabling us to implement a comprehensive defense strategy against poisoning attacks and achieve higher accuracy rates. However, FLGQM-NoQMS is able to achieve similar results relative to methods such as the median. This shows that robustness with certain effects can be achieved by using only geometric metric scores.

Table 4. Accuracy of compared federated learning algorithms on MNIST and CIFAR-10 with different attacks.

Dataset	Attacks	FedAvg	Median	Trim_mean	Krum	FLTrust	MAB-RFL	FLGQM-NoQMS	FLGQM
MNIST	No attack	98.51	98.25	98.22	95.01	97.84	97.95	97.98	98.39
	LF attack	-	97.79	97.41	93.51	97.40	96.60	95.94	98.03
	LIE attack	-	96.25	88.62	94.49	97.14	97.44	96.83	98.05
	Fang attack	-	96.17	95.64	94.75	97.10	97.45	95.53	97.97
	AGRT attack	-	95.18	95.84	74.19	97.03	97.14	87.64	97.63
CIFAR-10	No attack	73.68	72.83	72.58	66.36	73.29	73.70	73.15	73.55
	LF attack	-	69.59	69.35	57.63	73.21	73.66	65.59	73.53
	LIE attack	-	48.38	35.77	41.68	72.95	72.62	69.28	73.30
	Fang attack	-	58.73	66.57	42.43	73.19	72.18	67.30	73.39
	AGRT attack	-	23.06	53.77	10.00	72.52	70.55	56.34	72.63

3.2.3. Impact of Malicious Clients' Number

Table 5 presents the accuracy results of various robust federated learning methods on MNIST, considering different attacks and varying proportions of malicious clients (ranging from 0% to 50%). Since Trim_mean works by eliminating local model updates with twice the number of malicious clients, only 0–40% of the results are displayed for Trim_mean.

Table 5. Accuracy of compared federated learning algorithms on MNIST with different attacks.

Attacks	Percentage of Malicious Clients	FedAvg	Median	Trim_mean	Krum	FLTrust	MAB-RFL	FLGQM
LF attack	0%	98.51	98.25	98.22	95.01	97.84	97.95	98.39
	10%	-	98.04	98.01	94.81	97.69	97.42	98.22
	20%	-	97.79	97.41	93.51	97.40	96.60	98.03
	30%	-	96.96	96.54	93.27	97.42	97.28	98.01
	40%	-	94.14	93.71	93.14	95.78	97.12	97.87
	50%	-	17.44	-	23.55	95.23	95.32	97.86
LIE attack	0%	98.51	98.25	98.22	95.01	97.84	97.95	98.39
	10%	-	98.01	97.95	94.63	97.57	97.42	98.26
	20%	-	96.25	88.62	94.49	97.14	97.44	98.05
	30%	-	96.06	84.77	94.46	97.25	97.22	98.02
	40%	-	95.96	79.67	30.43	97.04	96.94	97.89
	50%	-	16.80	-	11.22	97.10	96.86	97.64
Fang attack	0%	98.51	98.25	98.22	95.01	97.84	97.95	98.39
	10%	-	96.35	96.69	94.79	97.38	97.86	98.33
	20%	-	96.25	96.54	94.75	97.10	97.45	97.97
	30%	-	96.17	95.36	94.19	96.27	97.57	97.77
	40%	-	95.31	9.80	10.00	94.58	97.42	97.68
	50%	-	10.30	-	9.90	94.20	97.37	97.59
AGRT attack	0%	98.51	98.25	98.22	95.01	97.84	97.95	98.39
	10%	-	96.98	96.77	94.88	97.21	97.23	98.09
	20%	-	95.18	95.84	74.19	97.03	97.14	97.63
	30%	-	95.09	90.03	66.84	96.99	96.12	97.45
	40%	-	94.98	9.80	10.00	96.37	96.10	97.41
	50%	-	9.80	-	9.80	95.49	96.00	97.20

It can be observed through Table 5 and Figure 4 that FLGQM is able to defend against 0–50% malicious clients in the presence of untargeted and targeted attacks. When there are 50% malicious clients, FLGQM is still able to have almost comparable performance to FedAvg, with a maximum error of only 1.19%. Nevertheless, alternative robust federated learning methods exhibit lower tolerance toward malicious clients. For example, under the LF attack, median and Krum almost fail when the malicious client reaches 50%, and Trim_mean's performance drops by 4.51% when the malicious client reaches 40%. Under the LIE attack, Krum fails when the malicious client reaches 40%; median fails when the malicious client reaches 50%; and the performance of Trim_mean drops by 18.55% when the malicious client reaches 40%. Under model poisoning attacks, such as Fang and AGRT, Trim_mean and Krum fail when the malicious clients reach 40%, and median fails when it reaches 50%. The most recent defense method, FLTrust, exhibits a decrease in accuracy under LF attack, Fang attack, and AGRT attack, especially when the malicious clients' proportion reaches 40%. Similarly, the latest defense method, MAB-FL, experiences a certain degree of accuracy reduction under LF attack when the malicious clients' proportion reaches 50% and under AGRT attack when the malicious clients' proportion reaches 30%. It can be seen that FLGQM is able to tolerate a much larger number of malicious clients.

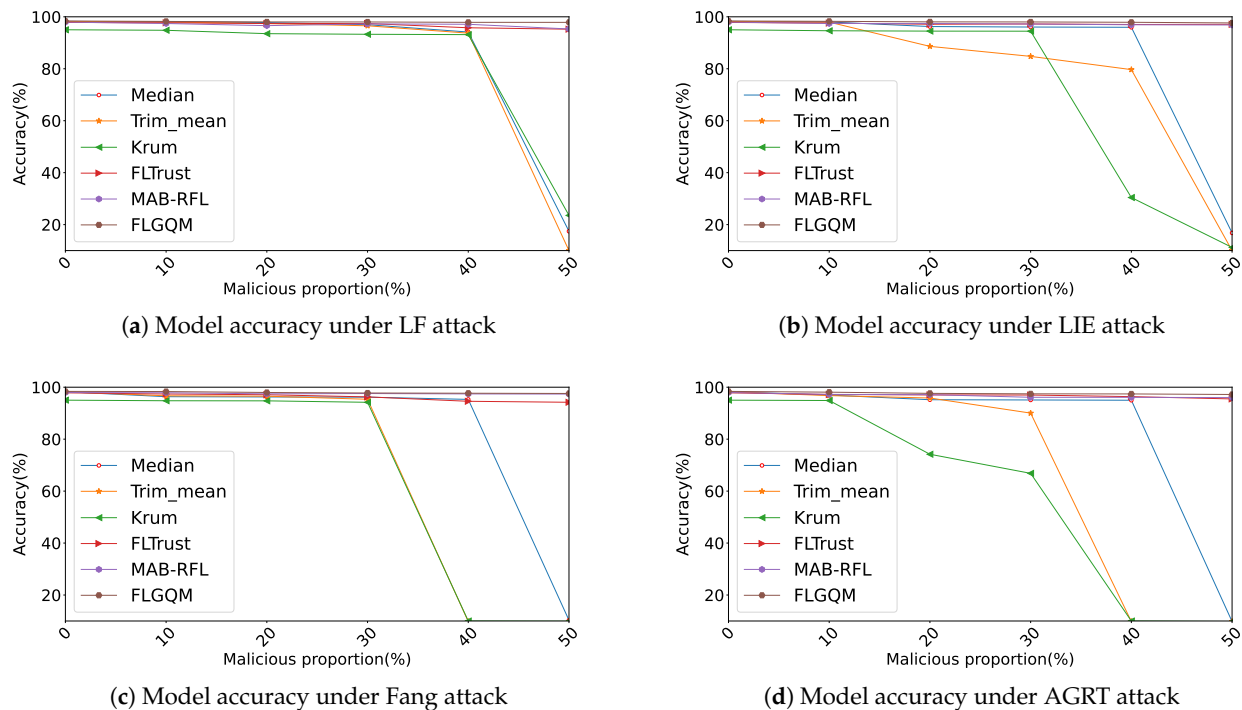


Figure 4. Accuracy of compared federated learning algorithms on MNIST with different attacks.

4. Conclusions

We propose a novel federated learning method called FLGQM for defense against malicious clients. By designing a defense mechanism from geometric metrics, we can geometrically exclude malicious clients. By incorporating quality metrics in addition to geometric metrics, we can effectively detect and mitigate poisoning attacks that may go unnoticed when relying solely on geometric metrics. This comprehensive approach enhances the robustness of our defense mechanisms. Our approach combines both geometric and qualitative aspects, thus it can ensure the robustness of federation learning.

Our evaluation on two datasets shows that FLGQM can achieve robustness against most malicious clients. Moreover, even in scenarios involving a substantial number of malicious clients, FLGQM can successfully train a global model that performs comparably to the global model learned by FedAvg in the absence of attacks.

There are several potential avenues for improvement in future work. One possibility is to explore the development of an optimal method, potentially utilizing other geometric statistics, to enhance the accuracy of geometric metrics. We will also explore the design of a local parameter poisoning attack against FLGQM and subsequently optimize FLGQM to enhance its resilience against such attacks. In addition, we consider the introduction of a reputation mechanism to enable a better approach to union and to coalition client selection.

Author Contributions: Conceptualization, S.L.; methodology, S.L. and X.X.; validation, F.W.; formal analysis, S.L. and X.X.; investigation, S.L.; data curation, X.X. and Q.Z.; writing—original draft preparation, S.L. and X.X.; writing—review and editing, S.L. and C.Z.; supervision, Y.J.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62076139), Jiangsu Key Development Planning Project (BE2023004-2), Natural Science Foundation of Jiangsu Province (Higher Education Institutions) (20KJA520001), Jiangsu Hongxin Information Technology Co., Ltd Project (JSSGS2301022EGN00), Future Network Scientific Research Fund Project (No. FNSRFP-2021-YB-15).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/geektutu/tensorflow-tutorial-samples/tree/master/mnist/data_set, <https://www.cs.toronto.edu/~kriz/cifar.html>.

Conflicts of Interest: The authors declare that this study received funding from Jiangsu Hongxin Information Technology Co., Ltd. Project. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

References

- McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
- Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against support vector machines. *arXiv* **2012**, arXiv:1206.6389.
- Nelson, B.; Barreno, M.; Chi, F.J.; Joseph, A.D.; Rubinstein, B.I.; Saini, U.; Xia, K. Exploiting machine learning to subvert your spam filter. In Proceedings of the LEET '08: USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, CA, USA, 15 April 2008; pp. 16–17.
- Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; Li, B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–23 May 2018; pp. 19–35.
- Muoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.C.; Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Redmond, WA, USA, 30 October–3 November 2017; pp. 27–38.
- Guerraoui, R.; Rouault, S. The hidden vulnerability of distributed learning in byzantium. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3521–3530.
- Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 634–643.
- Fang, M.; Cao, X.; Jia, J.; Gong, N.Z. Local model poisoning attacks to byzantine-robust federated learning. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USA, 12–14 August 2020; pp. 1623–1640.
- Baruch, G.; Baruch, M.; Goldberg, Y. A little is enough: Circumventing defenses for distributed learning. In Proceedings of the NeurIPS 2019: Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8632–8642.
- Xie, C.; Koyejo, O.; Gupta, I. Generalized byzantine-tolerant sgd. *arXiv* **2012**, arXiv:1802.10116.
- Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, Online, 26–28 August 2020; pp. 2938–2948.
- Xie, C.; Koyejo, O.; Gupta, I. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In Proceedings of the 36th Uncertainty in Artificial Intelligence, Online, 3–6 August 2020; pp. 261–270.
- Shejwalkar, V.; Houmansadr, A. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In Proceedings of the 28th Network and Distributed System Security Symposium, Online, 21–25 February 2021.
- Xie, C.; Huang, K.; Chen, P.Y.; Li, B. Dba: Distributed backdoor attacks against federated learning. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
- Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Proceedings of the NIPS 2017: Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
- Chen, Y.; Su, L.; Xu, J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc. Acm Meas. Anal. Comput. Syst.* **2017**, *1*, 1–25. [\[CrossRef\]](#)
- Yang, H.; Zhang, X.; Fang, M.; Liu, J. Byzantine-Resilient stochastic gradient descent for distributed learning: A Lipschitz-Inspired coordinate-wise median approach. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 5832–5837.
- Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
- Cao, X.; Fang, M.; Liu, J.; Gong, N.Z. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv* **2020**, arXiv:2012.13995.
- Xie, C.; Koyejo, S.; Gupta, I. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6893–6901.
- Zhao, B.; Sun, P.; Wang, T.; Jiang, K. Fedinv: Byzantine-robust federated learning by inverting local model updates. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2022; pp. 9171–9179.
- Mahloujifar, S.; Mahmood, M. Mohammed, A. Universal multi-party poisoning attacks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 4274–4283.
- Xia, Q.; Tao, Z.; Hao, Z.; Li, Q. FABA: An algorithm for fast aggregation against byzantine attacks in distributed neural networks. In Proceedings of the International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4824–4830.

24. Fung, C.; Yoon, C.J.; Beschastnikh, I. Mitigating sybils in federated learning poisoning. *arXiv* **2018**, arXiv:1808.04866.
25. Wan, W.; Hu, S.; Lu, J.; Zhang, L.Y.; Jin, H.; He, Y. Shielding Federated Learning: Robust Aggregation with Adaptive Client Selection. *arXiv* **2022** arXiv:2204.13256.
26. Pillutla, K.; Kakade, S.M.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [[CrossRef](#)]
27. Muñoz-González, L.K.; Co, T.; Lupu, E.C. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv* **2019**, arXiv:1909.05125.
28. Xu, J.; Huang, S.L.; Song, L.; Lan, T. Byzantine-robust federated learning through collaborative malicious gradient filtering. In Proceedings of the 42nd International Conference on Distributed Computing Systems (ICDCS 2022), Bologna, Italy, 10–13 July 2022; pp. 1223–1235.
29. Geng, G.; Cai, T.; Yang, Z. Better safe than sorry: Constructing byzantine-robust federated learning with synthesized trust. *Electronics* **2023**, *12*, 2926. [[CrossRef](#)]
30. Wang, T.; Zhao, B.; Fang, L. FLForest: Byzantine-robust Federated Learning through Isolated Forest. In Proceedings of the 28th International Conference on Parallel and Distributed Systems (ICPADS), Nanjing, China, 10–12 January 2023; pp. 296–303.
31. Cao, X.; Lai, L. Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Trans. Signal Process.* **2019**, *67*, 5850–5864. [[CrossRef](#)]
32. Rodríguez-Barroso, N.; Martínez-Cámara, E.; Luzón, M.V.; Herrera, F. Dynamic defense against byzantine poisoning attacks in federated learning. *Future Gener. Comput. Syst.* **2022**, *133*, 1–9. [[CrossRef](#)]
33. Cao, X.; Zhang, Z.; Jia, J.; Gong, N.Z. Flcert: Provably secure federated learning against poisoning attacks. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3691–3705. [[CrossRef](#)]
34. Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; Zhang, J. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet Things J.* **2019**, *6*, 10700–10714. [[CrossRef](#)]
35. Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [[CrossRef](#)]
36. Krizhevsky, A.; Hinton, G. Chen, C. F. R.; Fan, Q.; Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 357–366.
37. Tolpegin, V.; Truex Gursoy, S.M.E.; Liu, L. Data poisoning attacks against federated learning systems. In Proceedings of the Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, Guildford, UK, 14–18 September 2020; pp. 480–501.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.