# Enhancing Federated Learning Robustness Through Clustering Non-IID Features

Yanli Li[1(✉)], Abubakar Sadiq Sani[2], Dong Yuan[1], and Wei Bao[3]

[1] School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia
{yanli.li,dong.yuan}@sydney.edu.au

[2] School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London SE10 9LS, UK
S.Sani@greenwich.ac.uk

[3] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
wei.bao@sydney.edu.au

**Abstract.** Federated learning (FL) enables many clients to train a joint model without sharing the raw data. While many byzantine-robust FL methods have been proposed, FL remains vulnerable to security attacks (such as poisoning attacks and evasion attacks) because of its distributed nature. Additionally, real-world training data used in FL are usually Non-Independent and Identically Distributed (Non-IID), which further weakens the robustness of the existing FL methods (such as Krum, Median, Trimmed-Mean, etc.), thereby making it possible for a global model in FL to be broken in extreme Non-IID scenarios.

In this work, we mitigate the vulnerability of existing FL methods in Non-IID scenarios by proposing a new FL framework called Mini-Federated Learning (Mini-FL). Mini-FL follows the general FL approach but considers the Non-IID sources of FL and aggregates the gradients by groups. Specifically, Mini-FL first performs unsupervised learning for the gradients received to define the grouping policy. Then, the server divides the gradients received into different groups according to the grouping policy defined and performs byzantine-robust aggregation. Finally, the server calculates the weighted mean of gradients from each group to update the global model. Owning the strong generality, Mini-FL can utilize the most existing byzantine-robust method. We demonstrate that Mini-FL effectively enhances FL robustness and achieves greater global accuracy than existing FL methods when against the security attacks and in Non-IID settings.

**Keywords:** Federated Learning (FL) · Byzantine-robust aggregation · Untargeted model attack

## 1 Introduction

Federated Learning (FL) is an emerging distributed learning paradigm that enables many clients to train a machine learning model collaboratively while


Y. Zheng et al. (Eds.): ACCV 2022, LNCS 13848, pp. 45–59, 2023.
https://doi.org/10.1007/978-3-031-27066-6_4

keeping the training data decentralized and users' privacy protected [13]. Generally speaking, FL contains three steps: 1) a server broadcasts the current global model to selected clients; 2) each client locally trains the model (called local model) and sends back the local model updates[1]; and 3) the server updates the global model by aggregating the local model updates received through a particular aggregation algorithm (AGR).

However, the distributed nature of training data makes FL vulnerable to various attacks (such as poisoning attacks) by malicious attackers and untrusted clients. Poisoning attack, which seeks to damage the model and generate misbehaviour, draws the most important threats to FL security. Through poisoning in different training stages, poisoning attacks can lead the global model to show an indiscriminate accuracy reduction (called untargeted attack) or attacker-chosen behaviour on a minority of examples (called targeted attack) [13]. One popular defence solution against the untargeted attack is introducing the byzantine-robust aggregation rule [3,4,11,20] on the server to update the global model. By comparing the client's model updates, these aggregation rules can find and discard the statistical outliers and prevent the suspected model uploaded from poisoning the global model. Although most of the studies [3,20] are designed and evaluated in an Independent and Identically Distributed (IID) setting. Assuming each client's data follows the same probability distribution, the training data in real-world FL applications are usually Non-IID due to location, time, and user clusters reasons, which make the existing byzantine-robust FL methods show little effectiveness and even fully break when facing the state-of-the-art attack [9].

The most common sources of Non-IID are a client corresponding to a particular location, a particular time window, and/or a particular user cluster [13,15]. In terms of location, various kinds of locations factors drive the most impact on the Non-IID of a dataset. For instance, the mammal's distributions are different due to the geographic location [12], customer profiles are different due to various city locations [18], and emoji usage patterns are different due to the demographic locations [13]. In terms of a time window, people's behaviour and objects' features can be very different at different times. For instance, the images of the parked cars sometimes are snow-covered due to the seasonal effects, and people's shopping patterns are different due to the fashion and design trends. In terms of a particular user, different personal preferences can result in a dataset Non-IID. For instance, [5] shows students from different disciplines have very different library usage patterns.

In this paper, we design a new FL framework, namely Mini-FL framework, to mitigate the research gap. Mini-FL considers the main source of Non-IID and identifies Geo-feature, Time-feature, and User feature as the alternative grouping features. Based on the grouping feature selected, the server defines the grouping principle through performing unsupervised learning. In each iteration, the server first assigns the received gradients to different groups and then performs byzantine-robust aggregation, respectively. Finally, the server aggregates the aggregation outcomes (called group gradient) from each group to update the

[1] In this work, we combined use "model update" and "gradient" with same meaning.

global model in each iteration. We use Krum [3], Median [20], and Trimmed-mean [20] as the byzantine-robust aggregation rule to evaluate our Mini-FL on the various dataset from different Non-IID levels. Our results show that Mini-FL effectively enhances the security of existing byzantine-robust aggregation rules and also reaches a high level of accuracy (without attack) in the extreme Non-IID setting. We also provide a case study to further demonstrate the effectiveness of Mini-FL in the real world.

Our contributions are summarized as follows:

- We propose the group-based aggregation method and identify three features (i.e., Geo-feature, Time-feature, and User-feature) as the grouping principles.
- We propose the Mini-FL framework to enhance the robustness of existing FL methods. Our results show these methods can achieve byzantine robustness through the Mini-FL framework even in an extreme Non-IID setting.

## 2 Related Work

### 2.1 Poisoning Attacks on Federated Learning

Poisoning attacks generally indicate the attack type that crafts and injects the model during training time. These attacks include data poisoning attacks [2] and model poisoning attacks [6,8,9,12,18] which are performed by poisoning the training data owned and gradients, respectively. The model poisoning attack directly manipulates gradients, which can bring higher attack impacts to FL.

Based on the adversary's goals, the attacks can be further classified into untargeted attacks [6,8,9,12,18] (model downgrade attacks) and targeted attacks [10,16] (backdoor attacks). In untargeted attacks, the adversary aims to reduce the global model's accuracy and entirely 'break' the model by participating in the learning task. In contrast, target attacks maintain the global model's overall accuracy but insert 'back door' in minority examples. These back-doors can result in a wrong reaction when the attacker-chosen action event occurs. For instance, [10] can force GoogLeNet to classify a panda as a gibbon by adding an imperceptibly small vector on the panda image; the Faster RCNN can not detect the 'stop' sign that added small perturbations [16]. As the untargeted draws lead to security threats for FL, we consider the setting of **untargeted model poisoning attacks** in this study which shows as follows:

"Reverse attack" [6] and "Random attack" [8] poison the global model by uploading a reverse gradient and a random gradient. "Partial drop attack" [8] replaces the gradient parameter as a 0 with a given probability and subsequently uploads the crafted gradient to poison the global model. "Little is enough attack" [1] and "Fall of empires attack" [19] leverage the dimension curse of machine learning and upload the crafted gradient by adding perturbation on the mean of the gradient owned (based on the capability). "Local model poisoning attack" [9] is a state of art attack. It infers the convergence direction of the gradients and uploads the scaled, reverse gradient to poison the global model.

### 2.2 Byzantine-Robust Aggregation Rules for FL

The FL server can effectively average and aggregate the local models received in non-adversarial settings [17]. However, linear combination rules, including averaging, are not byzantine resilient. In particular, a single malicious worker can corrupt the global model and even prevent global model convergence [3]. Therefore, the existing byzantine-robust aggregation rules have been designed to replace the averaging aggregation and address byzantine failures. Next, we discuss the popular byzantine-robust aggregation rules.

**Krum** [3]**:** Krum discards the gradients that are too far away from benign gradients. In particular, for each gradient received, Krum calculates the sum Euclidean distance of a number of the closest neighbours as the score. The gradient with the lowest score is the aggregation outcome and becomes the new global model in this iteration. As the number of the closest neighbors selected influences the score, Krum requires the number of attackers.

**Trimmed-Mean and Median** [20]**:** Trimmed-mean is a coordinate-wise aggregation rule which aggregates each model parameter, respectively. Specifically, for a given parameter, the server firstly sorts the parameter from all gradients received. Then, the server discards a part of the largest and smallest values and finally averages the remaining gradients as the corresponding parameter of the new global model in this iteration. The Median method is another coordinate-wise aggregation rule. In the Median method, the server firstly sorts the parameter from all gradients received and selects the median as the corresponding parameter of the new global model in this iteration.

**Bulyan** [11]**:** Bulyan can be regarded as a combination of Krum and Trimmed-mean. Specifically, Bulyan first selects a number of gradients by performing Krum (the gradient is then removed from the candidate pool once selected). Then Bulyan performs Trimmed-mean in the gradients selected to update the global model.

**FLTrust** [4]**:** FLTrust considers both the directions and magnitudes of the gradients. Particularly, the server collects a clean dataset and owns a corresponding model; in each iteration, FLTrust first calculates the cosine similarity between the gradient received and owned. The higher cosine similarity gradient gains a higher trust score and consequently participates in the weighted average with a higher proportion. Instead of directly participating in the aggregation, each gradient is normalized by the gradient server owned before the weighted average.

Table 1 illustrates the robustness of the existing FL methods/proposed Mini-FL methods against different attacks under the IID/Non-IID settings. Since these attacks (i.e., untargeted attack) aim to reduce the model's global accuracy indiscriminately, we use the global testing accuracy to evaluate the robustness of FL methods.

**Table 1.** The robustness of the existing FL methods against poisoning attacks

| | "Reverse","Random" | | "Partial" | | "Little","Fall" | | "Local" | |
|---|---|---|---|---|---|---|---|---|
| | IID | Non-IID | IID | Non-IID | IID | Non-IID | IID | Non-IID |
| Average | × | × | × | × | × | × | × | × |
| Krum | ✓ | × | ✓ | × | O | × | × | × |
| Trimmed-Mean | O | × | × | × | O | × | O | O |
| Median | ✓ | O | ✓ | O | ✓ | O | O | × |
| Bulyan | ✓ | O | ✓ | O | ✓ | O | O | O |
| FL-Trust | ✓ | O | ✓ | O | ✓ | O | ✓ | O |
| Mini Krum | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mini T-Mean | ✓ | O | ✓ | × | ✓ | ✓ | ✓ | × |
| Mini Median | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Non: Non-IID, ✓: effective, O: partially effective, ×: ineffective.

## 3 Problem Setup

### 3.1 Adversary's Objective and Capability

Adversary aims to reduce the model's global accuracy or 'fully break' the global by uploading the malicious gradients; this is also known as untargeted model poisoning attacks or model downgrade attacks [10,13,16]. We consider the adversary's capability and knowledge from three dimensions: the adversary amount, the malicious client's distribution, and the knowledge of aggregation rule. We assume the adversary controls some clients, called malicious clients, and we keep the setting of the adversary number of each existing FL method that Krum: $2f + 2 < n$, Trimmed-Mean: $2f < n$ and Trimmed-Median: $2f < n$, where $f$ is the number of attackers, $n$ denotes the number of all clients. The adversary knows the local training data on malicious clients and can arbitrarily send crafted local model updates to the server in each iteration. To guarantee the generality, we assume the distribution of malicious clients and benign clients are similar. Furthermore, we assume the adversary knows the aggregation rules but does not know the grouping principle.

### 3.2 Defense Objective and Capability

We aim to develop the FL framework to achieve byzantine robustness against untargeted attacks and embody the data minimization principle. Specifically, the new framework does not need clients to upload further information beyond local model updates. The server plays the defender's role and has access to the information naturally brought with the gradients uploaded (e.g., IP, Timestamp, etc.). We notice some byzantine-robust aggregation rules need to know the upper bound of the malicious clients [3,20]; we follow these settings but don't leak further information of malicious clients; specifically, the defender does not know the distribution of malicious clients.

# 4 Mini-FL Design and Analysis

## 4.1 Overview of Mini-FL

In our Mini-FL, the server assigns the model updates received into different groups and executes byzantine-robust aggregation accordingly. Specifically, Mini-FL follows the general FL framework but adds a new step (i.e., Grouped model aggregation) before the Global model update. Furthermore, a prepossessing step: Grouping principle definition is introduced before the training task starts. Figure 1 illustrates the Mini-FL framework.

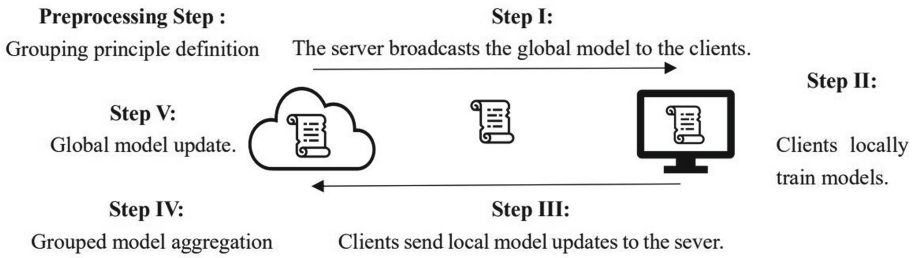


**Fig. 1.** Illustration of the Mini-FL framework.

To craft the malicious gradient and avoid being excluded by byzantine-robust aggregation rules, the adversary commonly statistically analyzes the gradient owned and calculates (or infers) the range of the benign gradients. By restricting the crafted gradient under this range, the attackers can effectively hide their gradients in benign gradients and subsequently attack the global model. However, because most federated learning models are trained through Non-IID data, the gradients uploaded naturally tend to be clustered due to location, time and user clusters reason. Thus, Mini-FL firstly defined the groups and then execute byzantine-robust aggregation accordingly. The similar behaviour of each group brings a smaller gradient range and therefore results in a smaller attack space. Finally, the server aggregates the outcome from each group and updates the global model to finish the current iteration.

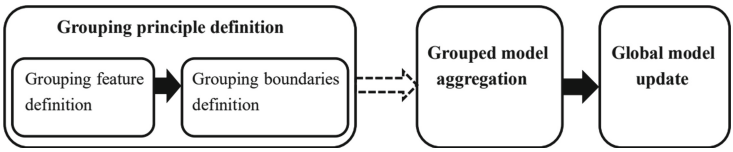


**Fig. 2.** Illustration of the Mini-FL aggregation rule.

### 4.2 Mini-FL Framework

Our Mini-FL considers leveraging the Non-IID nature of federated learning to define groups and execute byzantine-robust aggregation accordingly. Figure 2 illustrates the Mini-FL aggregation rule.

*Grouping Principle Definition.* Before the learning task starts, the server defines the grouping principle (i.e., prepossessing step), which includes "grouping feature definition" and "grouping boundaries definition"; the grouping principle could only be defined before the learning task starts or is required to be updated.

- **Grouping feature definition**: The existing research [13] believes the major sources of Non-IID are due to each client corresponding to a particular geographic location, a particular time window, and/or a particular user. For instance, [12] demonstrates the real-world example of skewed label partitions: geographical distribution of mammal pictures on Flickr, [13] illustrates the same label can also look very different at different times(e.g., seasonal effects, fashion trends, etc.).
  Considering the major source of Non-IID and the features naturally carried in server-client communication, we identify textbfGeo-feature (e.g., IP address), Time-feature (e.g., Timestamp), and User-feature (e.g., User ID) of the local model update as the based grouping feature to maintain the principle of focused collection and guarantee the effectiveness of clustering.
  When defining the grouping feature, the server firstly regroups the gradient collection $C$ by Geo-feature; the collection $C$ should accumulate the gradients received in a few iterations to maintain the generality. Then, we execute the 'elbow method' [14] to detect the number for clustering and subsequently get the SSE (i.e., Sum of the Squared Errors, which reflected the grouping effectiveness). By repeating the first two steps through replacing the Geo-feature with Time-feature and User-feature, we can find the feature $F$ with the lowest SSE. Finally, we select that feature $F$ acts as the grouping feature and the corresponding elbow point as the number of groups.
- **Grouping boundaries definition**: Once the grouping feature has been defined, we cluster the collection regrouped through unsupervised learning. In this research, we use the K-means algorithm to execute the unsupervised learning; the "elbow" point is assigned to the algorithm as the number of groups. By analyzing the gradient's feature value in different groups, the grouping boundaries could be defined.

*Grouped Model Aggregation.* According to the grouping principle, the server divides the gradients received into different groups and executes byzantine-robust aggregation respectively. The mini-FL framework has strong generality and can utilize most existing byzantine-robust aggregation rules. In this research, we used 'Krum,' 'Trimmed-mean,' and 'Median' for aggregation in this research, and the detail of the experiments are studied in Sect. 5.

*Global Model Update.* The server calculates the weighted mean of grouped gradients (i.e., outcome from each group) and updates the global model to finish this iteration.

### 4.3 Security Enhancement Analysis

In this section, we analyze the security enhancement of Mini-FL from 'information asymmetry' perspective.

As discussed in Sect. 2, most existing byzantine-robust aggregation rules can effectively detect and discard the malicious gradient if it is far (based on Euclidean distance) from benign gradients. To guarantee the attack effectiveness and avoid being excluded by the byzantine-robust aggregation rules, a common perturbation strategy is determining the attack direction and then scaling the crafted gradient to stay close with benign gradients. Depending on different knowledge, the adversary can precisely or generally infer the statistics (e.g., max, min, mean, and Std (Standard Deviation)) of the benign gradients and subsequently scale the crafted gradient; Table 2 illustrates the scalier of gradient crafted in different attacks.

**Table 2.** Illustration of the scalier of gradient crafted in different attacks.

| Attack | Crafted gradients range |
|---|---|
| "Little" [1] | $(\mu - z\sigma, \mu + z\sigma)$<br>$\mu$: mean, z: scalar (set 0∼1.5 in research), $\sigma$: Std. |
| "Fall" [19] | $(-z\mu, -z\mu)$<br>$\mu$: mean, z: scalar (0∼10 in research), $\sigma$: Std. |
| "Local" [9] | $(\mu + 3\sigma, \mu + 4\sigma)$ when the adversary has partial knowledge. or ($\mu$-4$\sigma$, $\mu$-3$\sigma$) depends on the gradient direction<br>$(Wmax, z * Wmax)$ when the adversary has full knowledge. or $(z * Wmin, Wmin)$ depends on the gradient direction<br>$\mu$: mean, z: scalar(set 2 in research), $\sigma$: Std, $Wmax/Wmin$: the max/min gradient value at that iteration |

However, Mini-FL defines the grouping principles and clusters the gradients received **only** on the server-side. The information asymmetry makes the adversary hardly infer the members of different groups, much less calculate the relevant statistical parameters to scale the crafted gradients and bypass the defense of Mini-FL.

## 5 Evaluation

### 5.1 Experimental Setup

*Dataset.* We evaluate our Mini-FL framework on the MNIST [7]. To simulate the dataset pattern in the real world, we set different Non-IID degrees when

distributing training data. Suppose we have $m$ groups of clients and $l$ different data labels; we set training data size as $s$ and assign $p*s$ training examples with label $l$ to the client group $m$ with probability $p$, then we randomly select and assign other $s-(p*s)$ training data to $m$ groups. As the parameter $p$ controls the distribution of training data on clients, we call $p$ the Non-IID degree. To further embody the source of each Non-IID distribution, we assign a feature (i.e., Geo, Time, or User feature) for each item of local model updates.

**MNIST-1.0:** The MNIST [7] (Modified National Institute of Standards and Technology) database is an extensive database of handwritten digits that includes 60,000 training images and 10,000 testing images. To simulate people's different handwriting habits in different countries [13], we divide clients into five groups; each group owns one unique IP range (reflect different countries) and training examples with two different labels (reflect different handwriting habits). We use MNIST-1.0 ($p = 1.0$) to simulate the extreme Non-IID situation (Non-IID degree = 1.0). In other words, each group only has two different unique labels of training examples in MNIST-1.0.

**MNIST-0.75 and MNIST-0.5:** We use MNIST-0.75 and MNIST-0.5 to evaluate the effectiveness of Mini-FL in different Non-IID degrees. MNIST-0.75 and MNIST-0.5 have similar settings as MNIST-1.0, but the Non-IID degree $p$ is 0.75 and 0.5, respectively.

*Evaluated Poisoning Attacks.* Mini-FL provides a new framework to enhance the security of FL and the excellent generalization enables Mini-FL can introduce most existing byzantine-robust aggregation rules. We introduce Krum [3], Trimmed-mean [20], Median [20] in experiments, respectively, and select the following poisoning attacks to evaluate the effectiveness of Mini-FL; we have not introduced FL-Trust in Mini-FL as FL-Trust does not fit extreme Non-IID scenarios - Krum adapted attacks can achieve 90% attack success rate when the root dataset's bias probability is over 0.6 [4].

**"Reverse Attack" [6]:** "Reverse attack" poisons the global model through uploading the reverse gradient. We follow the setting in [6] and set the attack multiple as 100.

**"Random Attack" [8]:** "Random attack" poisons the global model through uploading a random gradient.

**"Partial Drop Attack" [8]:** "Partial drop attack" masks the gradient parameter as 0 with probability p. As the parameter naturally carries a few 0 in our training tasks, we enhance the attack strength by replacing the mask 0 as -1 and setting p as 0.8 in experiments.

**"Little is Enough Attack" [1]:** "Little is enough attack" leverages the dimension curse of ML and upload the crafted gradient where gradient $= \mu + z * \sigma$; here, $\mu$ and $\sigma$ are the mean and standard deviation of the gradients respectively. z is the attack multiple, and we set z as 1.035, 1.535, and 2.035.

**"Fall of Empires Attack"** [19]**:** "Fall of empires attack" uploads the crafted gradient where gradient $= -z * \mu$. Here, $\mu$ is the mean of gradients and z is the attack multiple; we set z as 1 and 10.

**"Local Model Poisoning Attack"** [9]**:** "Local model poisoning attack" is a state of art attack. It infers the convergence direction of the gradients and uploads the scaled, reverse gradient to poison the global model. We follow the default setting in [9] for the local model poisoning attack.

*Evaluation Metrics.* Since these attacks (i.e., untargeted attack) aim to reduce the model's global accuracy indiscriminately, we use the testing accuracy to evaluate the effectiveness of our Mini-FL. In particular, we use a part of data owned as testing examples and test the model's global accuracy each iteration. The testing accuracy reflects the model's robustness against byzantine attacks; in other words, it is more robust if the model has a higher testing accuracy. We further use the existing FL methods with the original framework as the baseline to compare against.

*FL System Setting.* Without other specific notifications, we use the setting as follows.

Global model setting: As this study does not aim to improve the model accuracy through crafting the model, we use a general model for training MNIST. This model consists of a dense layer (28 * 28) and a softmax layer (10).

Learning parameters: We set the learning rate as 0.01, the batch size as 128, and the epoch as 50. We set the global iterations as 300. As some byzantine-robust methods (Krum in this study) require the parameter $M$ for the upper bound of the number of malicious clients, we follow the setting in [3] that the server knows the exact number of all malicious clients. However, since Mini-FL defines groups and performs aggregation accordingly, Mini-FL further requires the malicious clients m of each group when introducing Krum. To maintain the generality, We set $m$ belong with the group size:

$$m = \frac{n_{group}}{N_{global}} M$$

Here, $n_{group}$ is the client number of the group (i.e., group size) and $N_{global}$ is the total number of clients. In other words, we do not give any privilege to Mini-FL, and Mini-FL can only use the proportion to infer the number of malicious clients in each group.

Clients & data setting: We assume 20 clients participate in the learning task in each iteration, and 25% of clients are malicious. In Mini-FL, gradients are assigned in different groups as they carry different features. To simulate the Non-IID setting in the real world, we assign different numbers of clients to different groups; subsequently, the larger group has more malicious clients. Table 3 illustrates the setting detail for the MNIST (Non-IID degree = 1.0).

**Table 3.** Illustration of the setting (Client & Data) for the MNIST.

| | Group1 | Group2 | Group3 | Group4 | Group5 |
|---|---|---|---|---|---|
| Training labels | 1, 2 | 3, 4 | 5, 6 | 7, 8 | 9, 0 |
| Client ID | C1, C6<br>C11, C16, C19 | C2, C7<br>C12, C17, C20 | C3, C8<br>C13, C18 | C4, C9<br>C14 | C5, C10<br>C15 |
| Attackers | C1, C11 | C2, C12 | C3 | None | None |

### 5.2 Experimental Results

The results show Mini-FL achieves better robustness than the existing FL methods. Figures 3 and 4 illustrate the global accuracy of the existing FL methods/Mini-FL methods under different Non-IID degrees. When increasing the Non-IID degree, the results show that most Mini-FL methods can maintain a similar global accuracy under the same attack, while the existing FL methods witness decreasing global accuracy. For instance, Mini-median stably maintains around 90% global accuracy against various attacks and Non-IID settings. In contrast, Median achieves around 85% global accuracy against various attacks in MNIST-0.5 but drops global accuracy to 64.62%, 26.51%, 55.79%, and 53.68% in MNIST-1.0 under "little attack", "fall attack", "random attack", and "drop attack", respectively.

**Mini-FL Achieves the Defense Objectives:** Recall that the defense objectives include two parts (see Sect. 3): **achieving byzantine robustness against untargeted attacks** and **maintaining the data minimization principle of FL.** The experimental results show our Mini-FL framework achieves these goals.

**Table 4.** The global accuracy of different FL/Mini-FL methods under different Non-IID degrees and non-attack setting

| | MNIST-1.0 | MNIST-0.75 | MNIST-0.5 |
|---|---|---|---|
| Avg | 88.89% | 90.04% | 90.38% |
| Krum | 88.25% | 77.27% | 87.47% |
| Mini Krum | 89.51% | 90.03% | 90.09% |
| Median | 53.60% | 80.45% | 86.30% |
| Mini median | 90.34% | 90.26% | 90.47% |
| Trimmed-mean | 86.88% | 88.29% | 89.24% |
| Mini trimmed-mean | 90.38% | 90.47% | 90.51% |

First, Mini-FL achieves similar global accuracy as FedAvg (average aggregation rule) in the non-attack setting, but most existing byzantine robust FL methods have a decreased accuracy. For instance, FedAvg and all Mini-FL methods (i.e., Mini-Krum, Mini-Median, Mini-Trimmed mean) achieve over 90% global accuracy on MNIST-0.75 while Krum, Median, Trimmed mean get 77.27%, 80.45%, 88.29%, respectively. Table 4 illustrates the global accuracy of different FL/Mini-FL methods under different Non-IID degrees and non-attack settings. The result shows the Mini-FL framework increases the accuracy for existing

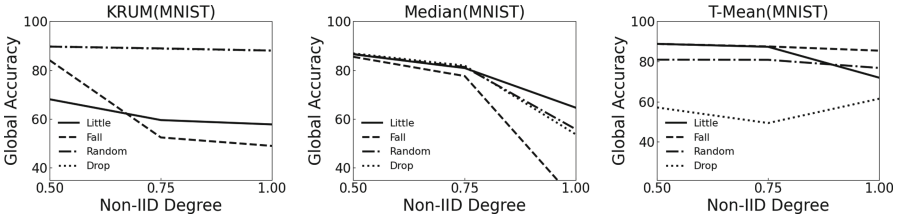


**Fig. 3.** The robustness of existing FL-methods under different Non-IID levels.

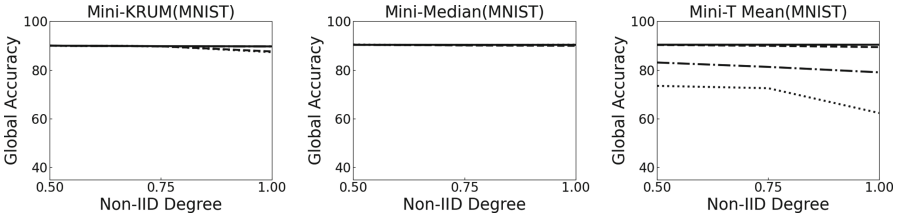


**Fig. 4.** The robustness of Mini FL-methods under different Non-IID levels.

**Table 5.** The global accuracy of FL/Mini-FL methods under different Non-IID degrees and non-attack setting

| | Average | Krum | Mini Krum | T-Median | Mini T-Median | T-Mean | Mini T-Mean |
|---|---|---|---|---|---|---|---|
| **(a) MNIST-1.0** | | | | | | | |
| **Little (2.035)** | 74.71% | 74.92% | 89.62% | 53.76% | 90.37% | 52.83% | 89.76% |
| Little (1.035) | 84.42% | 57.78% | 89.70% | 64.62% | 90.34% | 71.95% | 90.40% |
| **Fall (10)** | 23.73% | 77.34% | 88.38% | 54.98% | 90.10% | 61.54% | 90.18% |
| Fall (1) | 78.23% | 48.97% | 87.61% | 26.51% | 89.95% | 85.41% | 89.41% |
| **Random** | 80.19% | 88.03% | 89.68% | 55.79% | 90.37% | 76.80% | 79.04% |
| **Partial Drop** | 61.65% | 88.05% | 87.33% | 53.68% | 90.42% | 61.47% | 62.33% |
| **Local** | 78.62% | n/d | n/d | 2.85% | 89.77% | 64.51% | 87.32% |
| **(b) MNIST-0.75** | | | | | | | |
| **Little (2.035)** | 66.89% | 83.84% | 89.74% | 81.25% | 90.22% | 61.05% | 89.94% |
| Little (1.035) | 89.49% | 59.55% | 89.81% | 80.65% | 90.34% | 87.33% | 90.41% |
| **Fall (10)** | 85.33% | 77.27% | 89.77% | 79.15% | 89.98% | 64.09% | 90.23% |
| Fall (1) | 89.63% | 52.43% | 89.74% | 77.59% | 90.10% | 87.51% | 89.95% |
| **Random** | 74.85% | 88.93% | 89.74% | 81.28% | 90.24% | 80.85% | 81.28% |
| **Partial Drop** | 69.99% | 88.83% | 89.77% | 81.87% | 90.31% | 49.36% | 72.53% |
| **Local** | 85.16% | n/d | n/d | 62.31% | 90.00% | 75.90% | 88.78% |
| **(c) MNIST-0.5** | | | | | | | |
| **Little (2.035)** | 79.15% | 88.71% | 90.02% | 86.56% | 90.36% | 80.83% | 90.34% |
| Little (1.035) | 89.88% | 68.06% | 90.01% | 86.51% | 90.35% | 88.80% | 90.41% |
| **Fall (10)** | 88.38% | 89.66% | 90.03% | 85.88% | 90.38% | 71.64% | 90.35% |
| Fall (1) | 90.11% | 84.03% | 89.99% | 85.48% | 90.41% | 88.77% | 90.30% |
| **Random** | 78.43% | 89.67% | 90.02% | 86.60% | 90.36% | 80.92% | 83.08% |
| **Partial Drop** | 72.06% | 89.66% | 90.00% | 86.86% | 90.43% | 57.11% | 73.48% |
| **Local** | 86.37% | n/d | n/d | 80.68% | 90.28% | 76.48% | 89.81% |

FL methods in the non-attack scenario. This is because benign gradients could be very different in the Non-IID setting, which may be regarded as malicious gradients and discarded by the existing FL method. As Mini-FL performs the aggregation by groups, it could comprehensively collect features from different groups and guarantee global accuracy.

Second, our Mini-FL shows better robustness and stability than most existing FL methods against different attacks and under different Non-IID settings. Specifically, most Mini-FLs can maintain the unattacked global accuracy even facing a state of art attack and under an extreme Non-IID setting; on the contrary, existing FL methods immensely decrease global accuracy and even be fully broken. For instance, Mini-median achieves 89.77% global accuracy in MNIST-1.0 under 'local attack,' while Median drops global accuracy from 53.60% to 2.85%. Table 5 illustrates the global accuracy of FL/Mini-FL methods under different Non-IID degrees and different attacks.

Moreover, the result shows that although the Mini-trimmed mean improves the robustness for the trimmed mean method, it achieves lower global accuracy than other Mini-Fl methods. For instance, Mini-trimmed mean achieves 62.33% global accuracy under drop attack in MNIST1.0 while other Mini-FL methods get around 90%. This is because the original FL method (Trimmed mean ($\beta = 20\%$)) draws a larger attack surface than Krum and Median as Trimmed mean ($\beta = 20\%$) accept and aggregates 80% gradients received while Krum and Median accept only one gradient.

Third, Mini-FL maintains the principles of focused collection and data minimization of FL. All of the information used for grouping (i.e., IP address, response time, and client ID) are naturally carried by the gradients when uploading. Mini-FL neither asks clients to upload their information further nor digs their features through reverse engineering, which provides the same privacy protection as the existing FL methods.

## 6 Discussion and Future Work

Mini-Krum and Bulyan: Mini-Krum and Bulyan [11] are different, although both of them rely on performing Krum and mean/trimmed methods. Specifically, Mini-Krum performs Krum by group and generates the weighted average as the global model. In contrast, Bulyan globally performs Krum $n$ times to select $n$ gradients and performs Trimmed-mean to generate the global model. As Bulyan does not consider the Non-IID setting of FL, it faces a similar degraded performance as other FL methods in Non-IID scenarios.

Non-IID sources: As Geo-feature, Time-feature and User-feature are the most common source of Non-IID in the real world, we select these three features as the grouping feature in this research, but we note that the Non-IID source could be more complicated and even be a combination in some cases [13]. We leave investigating further to explore more possibilities of Non-IID sources and improve the Mini-FL method.

## 7 Conclusion

We evaluated the robustness of existing FL methods in different Non-IID settings and proposed a new framework called Mini-FL to enhance Federated Learning robustness. The main difference between Mini-FL and existing FL methods is that Mini-FL considers FL's Non-IID nature and performs the byzantine tolerant aggregation in different groups. Our evaluation shows that Mini-FL effectively enhances existing FL methods' robustness and maintains a stable performance against untargeted model attacks and different Non-IID settings.

## References

1. Baruch, G., Baruch, M., Goldberg, Y.: A little is enough: circumventing defenses for distributed learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
2. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
3. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
4. Cao, X., Fang, M., Liu, J., Gong, N.Z.: FLtrust: byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv:2012.13995 (2020)
5. Collins, E., Stone, G.: Understanding patterns of library use among undergraduate students from different disciplines. Evid. Based Libr. Inf. Pract. **9**(3), 51–67 (2014)
6. Damaskinos, G., El-Mhamdi, E.M., Guerraoui, R., Guirguis, A., Rouault, S.: Aggregathor: byzantine machine learning via robust gradient aggregation. In: Proceedings of Machine Learning and Systems, vol. 1, pp. 81–106 (2019)
7. Deng, L.: The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. **29**(6), 141–142 (2012)
8. El-Mhamdi, E.M., Guerraoui, R., Guirguis, A., Hoang, L.N., Rouault, S.: Genuinely distributed byzantine machine learning. In: Distributed Computing, pp. 1–27 (2022)
9. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 2020), pp. 1605–1622 (2020)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
11. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning, pp. 3521–3530. PMLR (2018)
12. Hsieh, K., Phanishayee, A., Mutlu, O., Gibbons, P.: The non-IID data quagmire of decentralized machine learning. In: International Conference on Machine Learning, pp. 4387–4398. PMLR (2020)
13. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends® Mach. Learn. **14**(1–2), 1–210 (2021)
14. Kodinariya, T.M., Makwana, P.R.: Review on determining number of cluster in k-means clustering. Int. J. **1**(6), 90–95 (2013)

15. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-IID data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978. IEEE (2022)
16. Lu, J., Sibai, H., Fabry, E.: Adversarial examples that fool detectors. arXiv preprint arXiv:1712.02494 (2017)
17. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
18. Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recogn. **45**(1), 521–530 (2012)
19. Xie, C., Koyejo, O., Gupta, I.: Fall of empires: breaking byzantine-tolerant SGD by inner product manipulation. In: Uncertainty in Artificial Intelligence, pp. 261–270. PMLR (2020)
20. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659. PMLR (2018)