Computers & Security 132 (2023) 103319

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Enhancing federated learning robustness in adversarial environment through clustering Non-IID features

Yanli Li[a,*], Dong Yuan[a], Abubakar Sadiq Sani[b], Wei Bao[c]

[a] School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia
[b] School of Computing and Mathematical Sciences, The University of Greenwich, London SE10 9LS, UK
[c] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia

ARTICLE INFO

Article history:
Received 6 February 2023
Revised 28 May 2023
Accepted 1 June 2023
Available online 5 June 2023

Keywords:
Federated learning (FL)
Non-independent and identically distributed (Non-IID)
Byzantine-robust aggregation
Untargeted model attack

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 due to its distributed adversarial environment. 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 aforementioned weaknesses of existing FL methods in Non-IID and adversarial 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 security attacks and in Non-IID settings.

1. Introduction

Federated Learning (FL) is an emerging distributed learning paradigm that enables many clients to train a machine learning model collaboratively while keeping the training data decentralized and users' privacy protected (Kairouz et al., 2021). 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; 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 (Jagielski et al., 2018; Jiang et al., 2019; Tomsett et al., 2019), 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) (Sun et al., 2019; Tolpegin et al., 2020). One popular defence solution against the untargeted attack is introducing the byzantine-robust aggregation rule (Blanchard et al., 2017; Cao et al., 2021; Guerraoui and Rouault, 2018; Yin et al., 2018) 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 (Blanchard et al., 2017; Sattler et al., 2020; Yin et al., 2018) are designed and evaluated in an Independent and Identically Distributed (IID) setting and assume each client's data follows the same probability distribution, the training data in real-world FL applications are usually

* Corresponding author.
E-mail addresses: yanli.li@sydney.edu.au (Y. Li), dong.yuan@sydney.edu.au (D. Yuan), S.Sani@greenwich.ac.uk (A.S. Sani), wei.bao@sydney.edu.au (W. Bao).

https://doi.org/10.1016/j.cose.2023.103319

Non-IID due to location, time, and user clusters reasons (Briggs et al., 2020a; Li et al., 2022b; Ma et al., 2022b), making the existing byzantine-robust FL methods show little effectiveness and even fully break when facing the state-of-the-art attack (Fang et al., 2020).

The most common sources of Non-IID are a client corresponding to a particular location (Baghbani et al., 2022; Hsieh et al., 2020; Kairouz et al., 2021; Ma et al., 2022a; Moreno-Torres et al., 2012), a particular time window (Abildgren, 2022; Gupta and Verma, 2022; Jiang et al., 2023; Pollak et al., 2022; Reshi et al., 2023), and/or a particular user cluster (Ghosh et al., 2020; Kairouz et al., 2021; Kushwah and Ranga, 2022; Pujahari and Sisodia, 2022). 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 (Hsieh et al., 2020), customer profiles are different due to various city locations (Moreno-Torres et al., 2012), and emoji usage patterns are different due to the demographic locations (Kairouz et al., 2021). 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, (Collins and Stone, 2014) shows students from different disciplines have very different library usage patterns.

In this paper, we first evaluate the effectiveness of the existing Byzantine-robust FL methods in different level Non-IID settings. We find these methods show degrading performance when increasing the Non-IID degree and further 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 global model in each iteration. We use Krum (Blanchard et al., 2017), Median (Yin et al., 2018), and Trimmed-mean (Yin et al., 2018) 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.

To the best of our knowledge, this is the first work to enhance FL robustness through Non-IID feature-based grouping algorithm. This paper is a significant extension of our prior conference paper (Li et al., 2022c), our contributions are summarized as follows:

- We comprehensively compare and evaluate the performance of existing FL methods in the Non-IID setting. Our results show these methods witness a degrading performance while increasing the Non-IID degree.
- We propose the grouping aggregation method and identify three features (i.e., Geo-feature, Time-feature, and User-feature) as the based 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 (Biggio et al., 2012) and model poisoning attacks (Damaskinos et al., 2019; El-Mhamdi et al., 2022; Fang et al., 2020; Hsieh et al., 2020; Moreno-Torres et al., 2012), 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 (Damaskinos et al., 2019; El-Mhamdi et al., 2022; Fang et al., 2020; Hsieh et al., 2020; Moreno-Torres et al., 2012) (model downgrade attacks) and targeted attacks (Goodfellow et al., 2014; Lu et al., 2017) (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, (Goodfellow et al., 2014) can force GoogLeNet (Szegedy et al., 2015) to classify a panda as a gibbon by adding an imperceptibly small vector on the panda image; the Faster RCNN (Ren et al., 2015) can not detect the "stop" sign that added small perturbations (Lu et al., 2017). 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"** (Damaskinos et al., 2019) **and "Random attack"** (El-Mhamdi et al., 2022): "Reverse" and "random attack" poison the global model by uploading a reverse gradient and a random gradient.
- **"Partial drop attack"** (El-Mhamdi et al., 2022): "Partial drop attack" 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"** (Baruch et al., 2019) and "Fall of empires attack" (Xie et al., 2020): "Little is enough attack" and "Fall of empires attack" 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"** (Fang et al., 2020): "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.

2.2. Byzantine-robust aggregation rules for federated learning

The FL server can effectively average and aggregate the local models received in non-adversarial settings (McMahan et al., 2017; Nguyen et al., 2020; Wu and Wang, 2021). 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 (Blanchard et al., 2017). 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 (*Blanchard et al., 2017*)* 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

Table 1
Illustration of the robustness of the existing FL/Mini-FL methods against different attacks under the IID/Non-IID setting.

	"Reverse" (Damaskinos et al., 2019), "Random" (El-Mhamdi et al., 2022)		"Partial" (El-Mhamdi et al., 2022)		"Little" (Baruch et al., 2019), "Fall" (Xie et al., 2020)		"Local" (Fang et al., 2020)	
	IID	Non	IID	Non	IID	Non	IID	Non
Vanilla (McMahan et al., 2017)	×	×	×	×	×	×	×	×
Krum (Blanchard et al., 2017)	✓	×	✓	×	O	×	×	×
Tri-mean (Yin et al., 2018)	O	×	×	×	O	×	O	O
Median (Yin et al., 2018)	✓	O	✓	O	✓	O	O	×
Mini Krum	✓	✓	✓	✓	✓	✓	✓	✓
Mini Median	✓	✓	✓	✓	✓	✓	✓	✓
Mini Tri-Mean	✓	O	✓	×	✓	✓	✓	×

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

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 (Yin et al., 2018) 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 (Guerraoui et al., 2018) 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 (Cao et al., 2021) and Sageflow (Park et al., 2021) 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. Similarly, Sageflow also considers keeping a clean validation set at the server. In each round, the client's gradients are evaluated through the public validation set, the gradients received small loss value are consequently assigned a heavyweight in aggregation.

ShieldFL (Ma et al., 2022c) ShieFL enhances the robustness of federated learning from the privacy-preserving perspective. Specifically, ShieFL first measures the distance between two encrypted gradients based on a presented secure cosine similarity method. Then, ShieFL generates the confidence parameters for each gradient based on its cosine similarity and determines its weight in aggregation.

We compares the robustness of several FL methods and their corresponding Mini-FL methods against different attacks under the IID/Non-IID settings in Table 1. We use the accuracy of the Vanilla (Standard) FL method in non-adversarial as the baseline to compare with; "✓ (effective)" denotes the global model can maintain a similar accuracy, "× (ineffective)" denotes the global model has been fully broken. "O (partially effective)" denotes the target FL method can maintain robustness only in some particular, moderate Non-IID scenarios but drops the global accuracy when facing state-of-art attacks or high Non-IID degrees. We note our Mini-FL framework can collect the information of each clients cluster and effectively enhance the robustness in Non-IID settings, the full evaluation information of different FL/Mini-FL methods against adversary are shown in Section 5.

2.3. Clustering federated learning

In the federated learning task, the training data is considered distributed in a Non-IID fashion because clients may behave heterogeneously across different IPs, time windows, and client clusters. To identify these clusters and further benefit FL, clustering methods have been widely introduced in recent FL research (Briggs et al., 2020b; Ghosh et al., 2020; Kim et al., 2021; Li et al., 2022a; Sattler et al., 2020). Here, we discuss several popular clustering federated learning algorithms; we note that most of these studies have considered different scenarios from our work. For instance, (Briggs et al., 2020b; Ghosh et al., 2020; Li et al., 2022a) consider a benign learning environment, (Sattler et al., 2020) consider an IID scenario, while our work setup in the Non-IID and adversarial scenario.

Federated learning + hierarchical clustering (FL + HC) (Briggs et al., 2020b) FL + HC introduces a hierarchical clustering step in the standard FL algorithm to enable a more significant percentage of clients can reach the target accuracy compared to standard FL. Specifically, the clustering step (HC) is introduced at a preset round to merge the most similar clusters of clients based on their local model updates. Then, the clients in each determined cluster are trained independently but simultaneously with the initialization of the current joint model.

Federated learning with soft clustering (FLSC) (Li et al., 2022a) Considering the client may belong to multiple clusters in real-world scenarios, FLSC introduces a soft clustering method to capture the complex nature of real-world data. Within FLSC, clients are assigned into overlapping clusters, and the information of each participant can be utilized by multiple clusters concurrently with each iteration.

Clustered federated learning (CFL) (Sattler et al., 2020) Our work most closely resembles that of (Sattler et al., 2020), which presents CFL algorithm to enhance the robustness of FL in IID settings. CFL first calculates the cosine similarity between each client's gradient and merges them if under a preset threshold. The gradients within the largest cluster are regarded as benign and consequently participate in the aggregation, all other gradients are discarded.

3. Problem setup

3.1. Threat model

We consider the adversary controls some clients and aims to reduce the model's global accuracy through untargeted model poi-

Table 2
Illustration the upper bound of different FL/Mini-FL methods.

Aggregation rule	Mini/Krum (Blanchard et al., 2017)	Mini/T-mean (Yin et al., 2018)	Mini/Median (Yin et al., 2018)
Attacker upper bound	$2f+2<n$	$2f<n$	$2f<n$

f: the amount of attackers, n: the amount of all clients.

soning attacks (Goodfellow et al., 2014; Kairouz et al., 2021; Lu et al., 2017). The attackers can access the dataset on the controlled device which is owned by the original client, and utilize the dataset to generate the crafted gradient. We assume the attackers can collude in each iteration and know the aggregation rules. Although these assumptions maximize the attack performance, we demonstrate our Mini-FL framework can achieve byzantine robustness even against attackers with strong capability in Sections 4 and 5. Specifically, the attacking processes are shown as follows: After receiving the current global model from the server in each iteration, the attacker generates the crafted model updates based on the data recourse owned and the poisoning strategy selected. Then, the crafted gradients are sent back to the server to poison the global model.

As our Mini-FL framework works based on an existing FL framework, we keep the setting of the adversary amount's upper boundary of each existing FL method. For example, the mini-Krum keeps the attacker upper bound as $f<(n-2)/2$ which is the same as the original Krum method. Table 2 illustrates the attacker upper bound of the FL methods introduced in the mini-FL framework in this paper; as the Fed-avg is not robust (attacker upper bound is 0), it has not been utilized and listed in the table.

In Mini-FL, the grouping principle used to partition the clients' gradients is fully defined by the server and could be updated on demand (full information is given in Section 4 part B). We note that the adversary can know the grouping principle ONLY if the attacker keeps controlling the server during the learning task, which is not practical. Hence, in this study, we assume the adversary does not know the grouping principle.

3.2. Defense objective

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.

3.3. Defender's knowledge and capability

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 (Blanchard et al., 2017) (Yin et al., 2018); 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. Fig. 1 illustrates 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.

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. Fig. 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 (Kairouz et al., 2021) 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, (Hsieh et al., 2020) demonstrates the real-world example of skewed label partitions: geographical distribution of mammal pictures on Flickr, (Kairouz et al., 2021) 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 **Geo-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" (Kodinariya and Makwana, 2013) 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 (Cheng, 1995) algorithm to execute the unsupervised learning, and we use the "elbow" point under the selected feature F (which is generated in the Grouping feature definition step) as the number of groups (i.e., K) for clustering. By analyzing the gradient's feature value in different groups, the grouping boundaries could be defined.

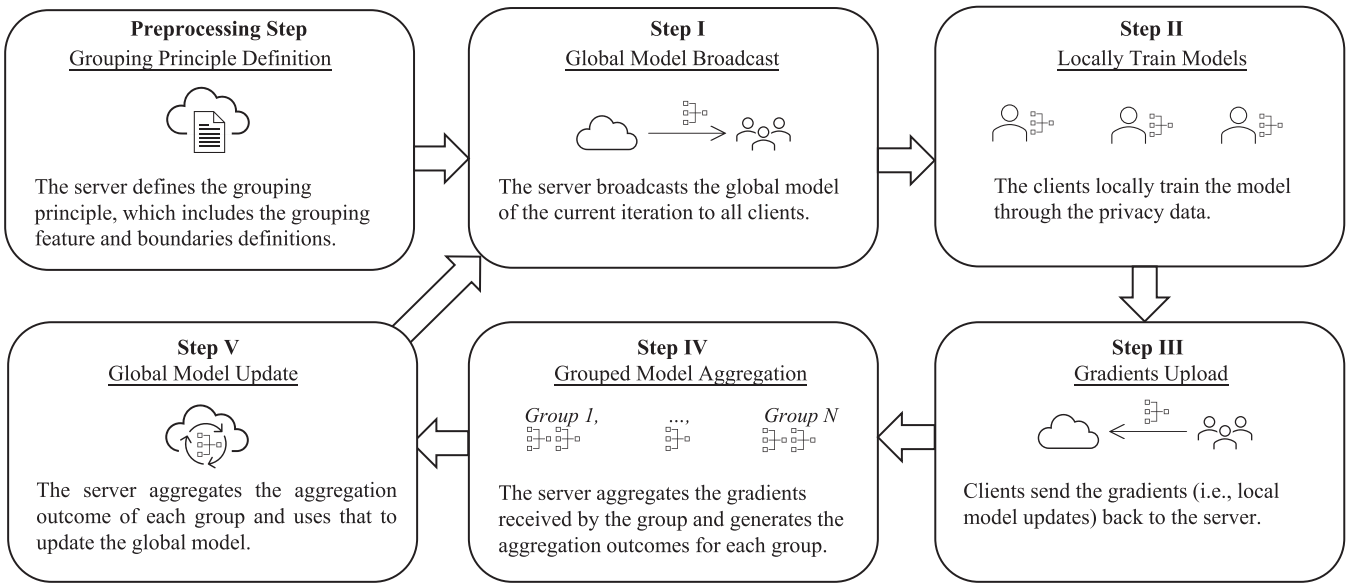

Fig. 1. Illustration of the Mini-FL framework.

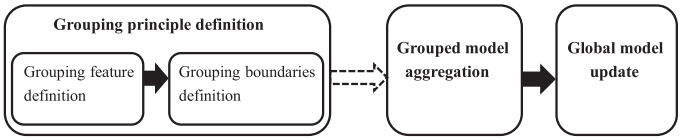

Fig. 2. Illustration of the Mini-FL aggregation rule.

In the Grouping Principle Definition step, all operations (i.e., elbow method and *K*-means method) are performed based on the gradients send from clients. Compared with the standard FL, The only additional information the Mini-FL used is/are the Geo-feature, Time-feature, or/and User-feature. We select these three features as they are the main source of Non-IID in FL, and the gradients naturally carry them during the most general server-client communication process. In other words, these features can be available to be collected even if they are not recorded in the client's data (consider the communication process when sending gradient to the server, the IP address of the message source, the response time, and the device ID are naturally carried by the message (gradient)), which does not break the client's privacy.

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 use Krum, Trimmed-mean, and Median for aggregation in this research, and the detail of the experiments are studied in Section 5. *Global model update* The server calculates the weighted mean of grouped gradients (i.e., the outcome from each group) base on the volume of each group to generate the global gradient and updates the global model to finish this iteration.

4.3. Toy example

We provide a toy example to illustrate how Mini-FL works. Suppose a learning task upgrades the existing FL framework to the Mini-FL framework, and the server receives *m* local model updates.

Processing step Server records *m* local model updates with nature information carried by the updates. Each update item is assigned an ID with its Geo-feature (IP address), Time-feature (sending time), and User-feature (Client ID); Table 3 illustrates the detail of the records.

Table 3
local model updates received with relevant information.

Update ID	Geo-feature	Time-feature	User-feature	Updates
U1	IP2	Time1	C1	G1
U2	IP1	Time1	C2	G2
U3	IP3	Time2	C3	G3
U4	IP2	Time1	C2	G4
U5	IP3	Time3	C4	G5
		...		
Um	IP4	Time3	C4	Gm

Table 4
Illustration of the records regrouped.

(a) Regroup by Geo-feature					
IP address	IP1	IP2	IP3	IP4	...
Local model updates	G2	G1,G4	G3,G5	Gm	...
(b) Regroup by Time-feature					
Time Stamp	Time1	Time2	Time3	Time4	...
Local model updates	G1,G2,G4	G3	G5,Gm	...	...
(c) Regroup by User-feature					
Client ID	C1	C2	C3	C4	...
Local model updates	G1	G2,G4	G3	G5,Gm	...

- **Grouping feature definition**: The server uses Geo-feature (IP address) as the index and regroups the local model updates; Table 4 illustrates the regrouped records. The server performs the "elbow method" for regrouped records and subsequently gets the SSE_Geo. Then, the server replaces the Geo-feature (IP address) as Time-feature (Time) and User-feature (Client ID) and generates the corresponding SSE_Time and SSE_User, respectively. Suppose SSE_Geo <SSE_User <SSE_Time; then the Geo-feature becomes the grouping feature.
- **Grouping boundaries definition**: The server performs the *K*-means algorithm for the regrouped records; suppose the results are generated as Table 5.
  So far, the grouping principle has been determinedthe updates that IP belongs to IP 1 range is assigned to Group1, IP belongs to IP 2 or IP 4 range is assigned to Group2, and IP belongs to IP 3 range assigned to Group3.

Table 5
Illustration of the *K*-means result.

(a) Geo-feature acts as the grouping feature				
Group	Group1	Group2	Group3	/
Geo-feature	IP1	IP2, IP4	IP3	/
(b) Time-feature acts as the grouping feature				
Group	Group1	Group2	Group3	Group4
Time-feature	Time1	Time2	Time3	Time4
(c) User-feature acts as the grouping feature				
Group	Group1	Group2	Group3	/
User-feature	C1,C2	C3	C4	/

In this proposed example, the Geo-feature achieves the lowest SSE and consequently acts as the grouping feature; the corresponding IP addresses clustered define the grouping boundaries (Table 5(a)). We also list the grouping boundaries if Time-feature and User-feature act as the grouping feature in Table 5(b) and (c); we note that there are different amounts of groups defined under different features.

Global model aggregation In each further iteration, the server assigns the updates to Group1, Group2, and Group3 based on its IP address and the grouping principle defined. Suppose we use Krum (Blanchard et al., 2017) as the aggregation rule, Group1, Group2, and Group3 performs Krum aggregation respectively and subsequently generate the grouped gradients: G_Group1, G_Group2, and G_Group3.

*Global model update*The server proportionally averages G_Group1, G_Group2, and G_Group3, suppose $\frac{1}{4}$ gradients received are assigned to Group1 and Group3, $\frac{1}{2}$ gradients received are assigned to Group2; the global aggregation result *G_aggre* is:

$$G\_aggre = \frac{1}{4}G\_(Group1) + \frac{1}{2}G\_(Group2) + \frac{1}{4}G\_(Group3)$$

Then, the server uses *G_aggre* to update the global model and finish this iteration.

4.4. Security enhancement analysis

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

Information asymmetry As discussed in Section 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 6 illustrates the scalier of gradient crafted in different attacks.

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.

Attack surface reduction Most existing byzantine-robust aggregation rules organize the attack surface by evaluating the Euclidean distances of the benign gradients, the attacker can craft its gradient to stay close to the boundary of attack surface to perform attack. In other words, as any gradient that beyond the boundary would be discarded, the attack surface limits the max of the perturbation.

We use Krum (Blanchard et al., 2017) and Local model attack (Fang et al., 2020) as example to formally derive the reduction of attack surface. Similar derivation can be applied to other defence and attack models. We first show that the attack surface is only limited by the benign models and then demonstrate the reduction of the attack surface by Mini-FL.

Suppose A_{krum} is the Krum aggregation rule, w_i is the local model that the ith worker device intends to send to the master device when there are no attacks. Without loss of generality, we assume the first c worker devices are compromised. Besides, w_i is the model before-attack, and w'_i is after-attack. Capital W refers to the global model. Thus, we have:

Before attack: $W = A_{krum}(w_1, \dots w_c, w_{c+1}, \dots, w_m)$

After attack: $W'_1 = A_{krum}(w'_1, \dots w'_c, w_{c+1}, \dots, w_m)$

We denote by Γ^a_w the set of local models among the crafted c compromised local models and $m-c$ benign local models that are the closest to the local model w with respect to Euclidean distance. Moreover, we denote by $\tilde{\Gamma^a_w}$ the set of benign local models that are the closest to w with respect to Euclidean distance. If w'_1 is chosen by Krum, we have the following:

$$\sum_{l\in\Gamma^{m-c-2}_{w'_1}} D(w_l, w'_1) \le min \sum_{l\in\Gamma^{m-c-2}_{w_i}} D(w_l, w_i) \tag{1}$$

Consider each malicious node could send the same w'_1 to max the attack effect, we have

$$\sum_{l\in\Gamma^{m-2c-2}_{w'_1}} D(w_l, w'_1) \le min[\sum_{l\in\Gamma^{m-c-2}_{w_i}} D(w_l, w_i) + CD(w'_1, w_i)] \tag{2}$$

Inequality (2) could approximately transfer to (3)

$$\sum_{l\in\Gamma^{m-2c-2}_{w'_1}} D(w_l, w'_1) - min[\sum_{l\in\Gamma^{m-c-2}_{w_i}} D(w_l, w_i)] \le CD(w'_1, w_i)] \tag{3}$$

In Eq. (3), as the subtrahend and c are fixed (i.e., the attackers cannot manipulate), the attack upper bound is only limited by $D(w'_1, w_i)$ However, because any $D \in \max\ C\sum_{l\in\Gamma^{m-c-2}_{w_i}} D(w_l, w_i)$ will be discarded; the upper bound should be:

$$max\ C \sum_{l\in\tilde{\Gamma}^{m-c-2}_{w_i}} D(w_l, w_i)$$

Table 6
Illustration of the crafted gradients range under different poisoning attacks.

Poisoning attack	Crafted gradients range
"Little is enough" (Baruch et al., 2019)	$(\mu - z\sigma, \mu + z\sigma)$ μ:mean, z:scalar (set 0 ~ 1.5 in research), σ:Std.
"Fall of empires" (Xie et al., 2020)	$(-z\mu, -z\mu)$ μ:mean, z:scalar (set 0 ~ 10 in research), σ:Std.
"Local model poisoning" (Fang et al., 2020)	$(\mu + 3\sigma, \mu + 4\sigma)$when the adversary has partial knowledge. or $(\mu$ - 4σ, μ - $3\sigma)$ depends on the gradient direction. $(W_{\max}, z * W_{\max})$when the adversary has full knowledge. or $(z * W_{\min}, W_{\min})$ depends on the gradient direction. μ:mean, z:scalar (set 2 in research), σ:Std, $W_{\max}/W_{\min}$:the max/min gradient value at that iteration.

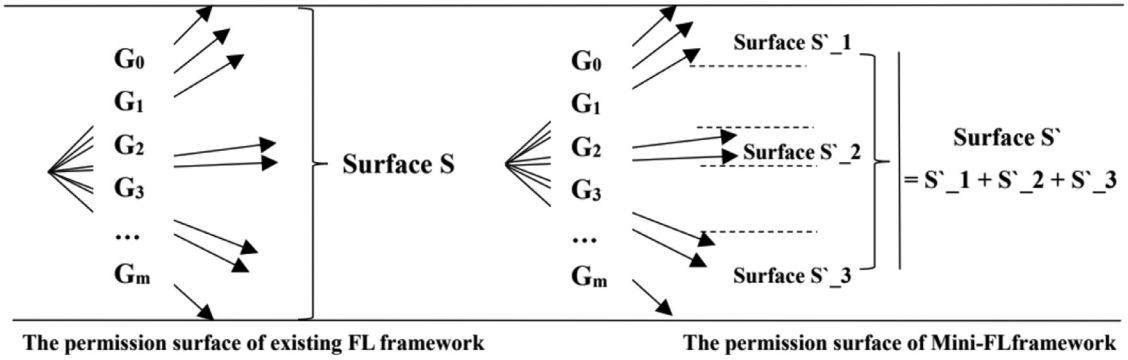

Fig. 3. attack surface comparison between the existing FL and Mini-FL.

In other words, the attack surface is only organized by benign model updates and limits the max of the perturbation.

However, because of the Non-IID character of federated learning, even benign gradients can still introduce a high Std and big attack surface. This brings difficulty to the existing FL methods for identifying, discarding malicious gradient, and reducing attack surface when the attackers stay close to the benign gradient boundary (i.e., G_0 and G_m in Fig. 3). On the contrary, in our Mini-FL, similar gradients are aggregated in the same group; the high inter-group similarity can lower Std and hence smaller attack surfaces. Suppose G_0, G_1 ... G_m are m gradients uploaded from benign clients, Fig. 3 illustrates the attack surface of the existing FL framework (i.e., S) and Mini-FL framework (i.e.,S'). The following derivation demonstrates how our Mini-FL reduces the attack surface and defense against a state of art attack (Fang et al., 2020).

To achieve the attack goal and to avoid being discarded by the byzantine-robust aggregation rule, Fang et al. (2020) sets the optimization problem as (4).

$$w_1' = w_{Re} - \lambda s \tag{4}$$

s is a column vector of the changing directions of all global model parameters (i.e., $s = 1 or -1$); w_{Re}presents the model received from sever this interaction and can be regarded as the initiation of crafting. Because of the training process, w_{Re} unavoidable to be broadcasted to all clients. Recall, Mini-FL method divides gradients received into different groups to generate the $w_g^{sub}, g = 1, 2, 3\ldots$ (g indicates the different group), and subsequently calculates the weighted mean of $w_g^{sub}, g = 1, 2, 3\ldots$ as w_{Re}. Thus, in Mini-FL, the attackers should use $w_g^{sub}, g = 1, 2, 3\ldots$ to replace w_{Re} in each group g and transfers the optimization problem as (5) to attack different group g to achieve the same attack effectiveness in Fang et al. (2020):

$$w_g^{sub'} = w_g^{re} - \lambda s, g = 1, 2, 3\ldots \tag{5}$$

However, because all group defining and gradients dividing work are only performed at the server side, all clients (including attackers) can only access w_{Re} instead of $w_g^{re}, g = 1, 2, 3\ldots$, attackers cannot access the initiation of crafting in Mini-FL.

On the other hand, λ is the key to executing the attack (Fang et al., 2020). In the proof proposed, λ is generated by solving the following inequality (6)

$$\sum_{l \in \Gamma_{w_1'}^{m-c-2}} D(w_l, w_1') \leq \min_{c+1 \leq i \leq m} \sum_{l \in \Gamma_{w_i}^{m-c-2}} D(w_l, w_i) \tag{6}$$

Then (6) transforms to (7) in Fang et al. (2020)

$$\sum_{l \in \tilde{\Gamma}_{w_1'}^{m-2c-1}} D(w_l, w_1') \leq \min_{c+1 \leq i \leq m} \sum_{l \in \Gamma_{w_i}^{m-c-2}} D(w_l, w_i) \tag{7}$$

We can find that the only difference is the research replaces the range of l from (8), (9)

$$l \in \Gamma_{w_1'}^{m-c-2} \tag{8}$$

$$l \in \tilde{\Gamma}_{w_1'}^{m-2c-1} \tag{9}$$

The necessary and sufficient condition of this replacement is the research assumes the other $c-1$ malicious models can stay closely with w_1' even same (i.e., The distance between w_1' and the other $c-1$ compromised local models is 0). However, the malicious nodes are sent to different groups and cannot stay the same because of grouping. Hence the result of the (7) will be represented as (10) under the Mini-FL framework:

$$\lambda \leq \frac{1}{\sqrt{d}}[\min_{c+1 \leq i \leq m} D(w_l, w_i) + \max_{c+1 \leq i \leq m} D(w_l, w_{Re})] \tag{10}$$

Recall λ in research (Fang et al., 2020) has been limited as (11)

$$\lambda \leq \frac{1}{(m-2c-1)\sqrt{d}} \min_{c+1 \leq i \leq m} (\sum_{l \in \tilde{\Gamma}_{w_i}^{m-c-2}} D(w_l, w_i)) + \frac{1}{\sqrt{d}} \max_{c+1 \leq i \leq m} D(w_i, w_{Re}) \tag{11}$$

As the upper bound of (10) is smaller than (11), λ's upper bound has been reduced by Mini-FL.

Because the initiation of crafting w_{Re} can not be assessed by malicious nodes and the upper bound of λ has been reduced, Mini-FL effectively relieves the attack (Fang et al., 2020) Furthermore, the Mini-FL method collects the information in each heterogeneous clusters which guarantee the information of different classes can be evenly learned by the global model even in extreme Non-IID settings.

5. Evaluation

5.1. Experimental setup

Dataset We use different datasets to evaluate our Mini-FL framework which include MNIST (Deng, 2012) and Fashion-MNIST (Xiao et al., 2017). Different from the IID settings, under Non-IID settings, the data from each client are drawn from the different label distributions. In this paper, we consider the most common Non-IID setting that the data volume of each client biases across different labels. Specifically, clients' data may be mainly occupied by different labels because of the interest heterogeneity. To simulate the dataset pattern mentioned, we set different Non-IID degrees when distributing training data. Suppose U is the universe of the data labels in the learning task, we set the training data size of each client as s and assign $p * s$ training examples with la-

bel L[1] ($L \in U$) to the client i to simulate the data of client i biases on the label L, where p is the probability. Then, we fill up the rest $s-(p*s)$ training data to this client evenly with other classes' data in $\complement_U L$. We note that the data of client i will evenly consist of all classes (not biased) if $p=\frac{1}{|U|}$, and only include class L if $p=1.0$ As the possibility parameter p controls the distribution of training data on clients, we call p as 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 (Deng, 2012) (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 (Kairouz et al., 2021), 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.

FMNIST-1.0: The FMNIST (Fashion-MNIST) (Xiao et al., 2017) dataset includes 60,000 gray-scale images of 10 fashion categories and a test set of 10,000 images. To simulate the changing trend of the fashion dress, we divide clients into five groups; each group belongs to a time window (i.e., Years, months) and has training examples with two different labels (reflect the popular dress). Similar to MNIST-1.0, we set $p=1.0$ for Fashion-MNIST-1.0 to simulate the extreme Non-IID setting.

FMNIST-0.75 and FMNIST-0.5: FMNIST-0.75 and FMNIST-0.5 reflect the different Non-IID degrees of the Fashion-MNIST dataset. We set $p=.75$ and $p=.5$ for Fashion-MNIST-0.75 and Fashion-MNIST-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 (Blanchard et al., 2017), Trimmed-mean (Yin et al., 2018), and Median (Yin et al., 2018) 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 attacks can achieve 90% attack success rate when the root dataset's bias probability is over 0.6 (Cao et al., 2021). We follow the threat model formed in Section 3 that assumes some clients have been controlled by the adversary and participant in the learning task each iteration. These malicious clients can craft the poisoned gradient based on the data owned by the original benign client through one of the following attack strategies. The data distribution and group information of the malicious client are shown in Table 7.

"Reverse attack" (Damaskinos et al., 2019): "Reverse attack" poisons the global model through uploading the reverse gradient. We follow the setting in Damaskinos et al. (2019) and set the attack multiple as 100.

[1] The L could be a single element or a set; in other words, the data could bias on one label or several labels.

Table 7
Illustration of the setting (Client & Data) for the MNIST and Fashion-MNIST (Non-IID degree = 1.0).

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

"Random attack" (El-Mhamdi et al., 2022): "Random attack" poisons the global model through uploading a random gradient.

"Partial drop attack" (El-Mhamdi et al., 2022): "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" (Baruch et al., 2019): "Little is enough attack" leverages the dimension curse of ML and uploads the crafted gradient where gradient $= \mu + z * \sigma$; here, μ and σ are the mean and standard deviation of the gradients respectively. z is the attack multiple, and we set z as 1.035 and 2.035.

"Fall of empires attack" (Xie et al., 2020): "Fall of empires attack" uploads the crafted gradient where gradient $= -z*\mu$. Here, μ is the mean of gradients and z is the attack multiple; we set z as 1 and 10.

"Local model poisoning attack" (Fang et al., 2020): "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 Fang et al. (2020) for the local model poisoning attack.

*Evaluation metrics*Since these attacks (i.e., untargeted attacks) 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 the 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 and Fashion-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 for MNIST and 500 for Fashion-MNIST. 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 Blanchard et al., 2017 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 to 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 amount of clients. In other words, we do

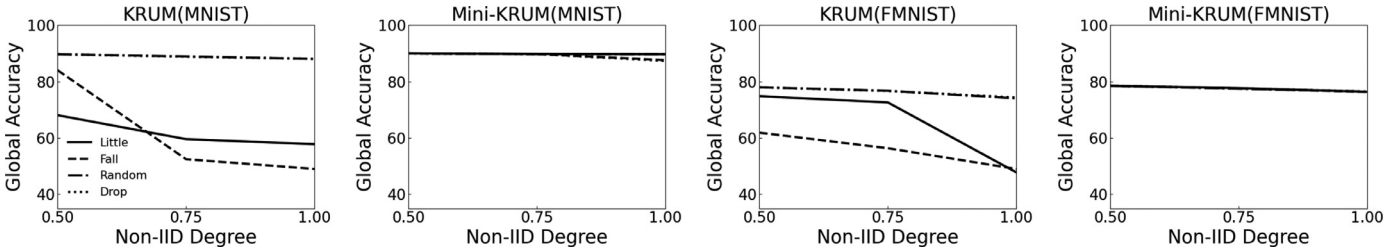

Fig. 4. The robustness comparison between Krum and Mini-Krum.

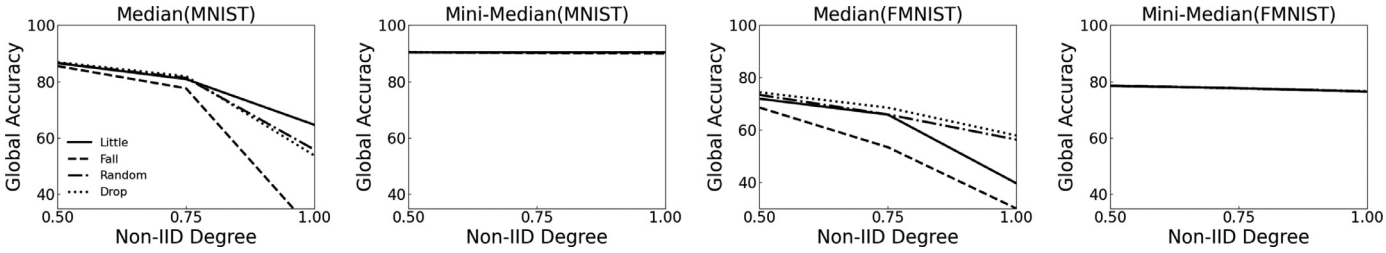

Fig. 5. The robustness comparison between Median and Mini-Median.

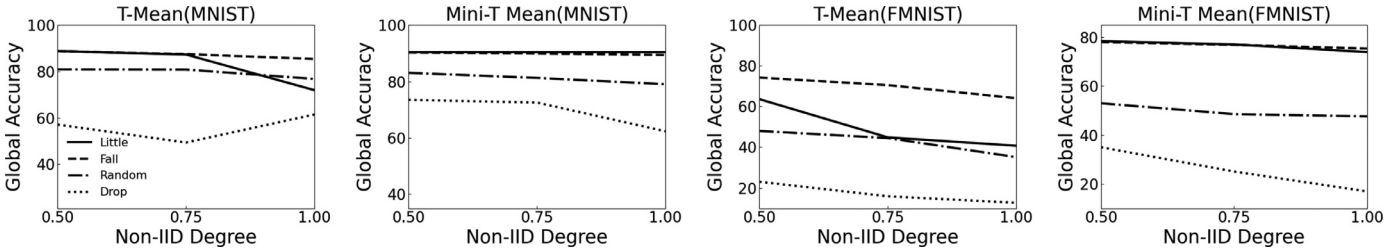

Fig. 6. The robustness comparison between Trimmed Mean and Mini-Trimmed Mean.

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 7 illustrates the setting detail for the MNIST and Fashion-MNIST (Non-IID degree = 1.0).

Table 8
The global accuracy of different FL/Mini-FL methods under different Non-IID degrees and non-attack setting.

	M-1.0	M-0.75	M-0.5	FM-1.0	FM-0.75	FM-0.5
Avg	88.89%	90.04%	90.38%	79.39%	77.33%	78.48%
Krum	88.25%	77.27%	87.47%	61.25%	65.80%	75.39%
Mini Krum	89.51%	90.03%	90.09%	76.51%	77.51%	78.49%
Median	53.60%	80.45%	86.30%	43.88%	65.87%	71.19%
Mini Med	90.34%	90.26%	90.47%	76.54%	77.74%	78.56%
Tri-Mean	86.88%	88.29%	89.24%	72.89%	74.92%	77.56%
Mini Tri-M	90.38%	90.47%	90.51%	76.53%	77.50%	78.47%

5.2. Experimental results

The results show the existing FL methods can not effectively aggregate the information of different classes in Non-IID scenarios; besides, the Mini-FL achieves better robustness than the existing FL methods. Figures 4, 5, and 6 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 Section 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.

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. We consider the reason includes the existing FL methods can not effectively aggregate the information of different clients (with corresponding classes) and defend against the adversary in Non-IID settings. 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; On FMNIST-1.0 (i.e., Fashion MNIST-1.0), FedAvg and all Mini-FL methods achieve the global accuracy around 77%, while 61.25%, 43.88%, and 72.89% for Krum, Median, and Trimmed mean, respectively. Table 8 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 FL methods in the non-attack scenario. This may because benign gradi-

Table 9
The global accuracy of FL/Mini-FL methods under different Non-IID degrees.

	Average	Krum (Blanchard et al., 2017)	Mini Krum	Median (Yin et al., 2018)	Mini Median	Trimmed-Mean (Yin et al., 2018)	Mini Trimmed-Mean
(a) MNIST-1.0							
"Little is enough" attack (Baruch et al., 2019), $z = 2.035$	74.71%	74.92%	89.62%	53.76%	90.37%	52.83%	89.76%
"Little is enough" attack (Baruch et al., 2019), $z = 1.035$	84.42%	57.78%	89.70%	64.62%	90.34%	71.95%	90.40%
"Fall of empires" attack (Xie et al., 2020), eps = 10	23.73%	77.34%	88.38%	54.98%	90.10%	61.54%	90.18%
"Fall of empires" attack (Xie et al., 2020), eps = 1	78.23%	48.97%	87.61%	26.51%	89.95%	85.41%	89.41%
"Random" attack (El-Mhamdi et al., 2022)	80.19%	88.03%	89.68%	55.79%	90.37%	76.80%	79.04%
"Partial Drop" attack (El-Mhamdi et al., 2022)	61.65%	88.05%	87.33%	53.68%	90.42%	61.47%	62.33%
"Local model poisoning" attack (Fang et al., 2020)	78.62%	n/d	n/d	2.85%	89.77%	64.51%	87.32%
(b) FMNIST-1.0							
"Little is enough" attack (Baruch et al., 2019), $z = 2.035$'	64.75%	74.15%	76.35%	43.79%	76.76%	29.66%	58.82%
"Little is enough" attack (Baruch et al., 2019), $z = 1.035$'	70.51%	47.76%	76.36%	39.56%	76.43%	40.72%	73.98%
"Fall of empires" attack (Xie et al., 2020), eps = 10	35.57%	74.35%	76.22%	44.18%	76.21%	38.48%	76.60%
"Fall of empires" attack (Xie et al., 2020), eps = 1	48.67%	48.97%	76.28%	26.91%	76.44%	64.10%	75.43%
"Random" attack (El-Mhamdi et al., 2022)	66.95%	74.04%	76.44%	56.26%	76.47%	35.12%	47.64%
"Partial Drop" attack (El-Mhamdi et al., 2022)	19.78%	74.33%	76.46%	57.84%	76.58%	12.69%	16.88%
"Local model poisoning" attack (Fang et al., 2020)	45.88%	n/d	n/d	2.82%	76.26%	45.97%	60.51%
(c) MNIST-0.75							
"Little is enough" attack (Baruch et al., 2019), $z = 2.035$	66.89%	83.84%	89.74%	81.25%	90.22%	61.05%	89.94%
"Little is enough" attack (Baruch et al., 2019), $z = 1.035$	89.49%	59.55%	89.81%	80.65%	90.34%	87.33%	90.41%
"Fall of empires" attack (Xie et al., 2020), eps = 10	85.33%	77.27%	89.77%	79.15%	89.98%	64.09%	90.23%
"Fall of empires" attack (Xie et al., 2020), eps = 1	89.63%	52.43%	89.74%	77.59%	90.10%	87.51%	89.95%
"Random" attack (El-Mhamdi et al., 2022)	74.85%	88.93%	89.74%	81.28%	90.24%	80.85%	81.28%
"Partial drop" attack	69.99%	88.83%	89.77%	81.87%	90.31%	49.36%	72.53%
"Local model poisoning" attack (Fang et al., 2020)	85.16%	n/d	n/d	62.31%	90.00%	75.90%	88.78%
(d) FMNIST-0.75							
"Little is enough" attack (Baruch et al., 2019), $z = 2.035$	31.07%	76.48%	77.52%	65.26%	77.84%	36.18%	71.30%
"Little is enough" attack (Baruch et al., 2019), $z = 1.035$	61.64%	72.59%	77.68%	65.81%	77.62%	44.84%	77.10%
"Fall of empires" attack (Xie et al., 2020), eps = 10	64.79%	76.32%	77.64%	64.11%	77.69%	42.56%	77.71%
"Fall of empires" attack (Xie et al., 2020), eps = 10	75.72%	56.33%	77.55%	53.35%	77.63%	70.54%	76.95%
"Random" attack (El-Mhamdi et al., 2022)	28.77%	76.72%	77.44%	65.87%	77.66%	44.50%	48.56%
"Partial drop" attack (El-Mhamdi et al., 2022)	22.18%	76.65%	77.62%	68.47%	77.56%	15.88%	25.05%
"Local model poisoning" attack (Fang et al., 2020)	61.03%	n/d	n/d	31.84%	77.51%	51.57%	63.58%
(e) MNIST-0.5							
"Little is enough" attack (Baruch et al., 2019) $z = 2.035$	79.15%	88.71%	90.02%	86.56%	90.36%	80.83%	90.34%
"Little is enough" attack (Baruch et al., 2019) $z = 1.035$	89.88%	68.06%	90.01%	86.51%	90.35%	88.80%	90.41%
"Fall of empires" attack (Xie et al., 2020), eps = 10	88.38%	89.66%	90.03%	85.88%	90.38%	71.64%	90.35%
"Fall of empires" attack (Xie et al., 2020), eps = 1	90.11%	84.03%	89.99%	85.48%	90.41%	88.77%	90.30%
"Random" attack (El-Mhamdi et al., 2022)	78.43%	89.67%	90.02%	86.60%	90.36%	80.92%	83.08%
"Partial drop" attack (El-Mhamdi et al., 2022)	72.06%	89.66%	90.00%	86.86%	90.43%	57.11%	73.48%
"Local model poisoning" attack (Fang et al., 2020)	86.37%	n/d	n/d	80.68%	90.28%	76.48%	89.81%
(f) FMNIST-0.5							
"Little is enough" attack (Baruch et al., 2019) $z = 2.035$	37.92%	77.80%	78.48%	73.12%	78.57%	42.14%	74.07%
"Little is enough" attack (Baruch et al., 2019) $z = 1.035$	73.56%	74.80%	78.48%	71.89%	78.46%	63.63%	78.46%
"Fall of empires" attack (Xie et al., 2020), eps = 10	70.55%	77.97%	78.46%	71.56%	78.51%	52.36%	78.47%
"Fall of empires" attack (Xie et al., 2020), eps = 1	77.17%	61.85%	78.45%	68.52%	78.47%	74.23%	78.04%
"Random" attack (El-Mhamdi et al., 2022)	33.06%	77.96%	78.47%	73.36%	78.54%	48.00%	53.02%
"Partial drop" attack (El-Mhamdi et al., 2022)	29.65%	77.96%	78.52%	74.35%	78.51%	23.08%	34.90%
"Local model poisoning" attack (Fang et al., 2020)	29.65%	n/d	n/d	63.93%	78.33%	70.18%	76.01%

ents 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 9 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

Table 10
Illustration of the data deviation.

	Whole range	→	Range1	Range2	...	Range9	Range10
DIS Range	3.5 ~ −1.5	→	3.5 ~ 3.0	3.0 ~ 2.5	...	−0.5 ~ −1	−1 ~ −1.5

achieves lower global accuracy than other Mini-Fl methods. For instance, Mini-trimmed mean achieves 62.33% and 16.88% global accuracy under drop attack in MNIST1.0 and Fashion MNIST-1.0 while other Mini-FL methods get around 90% and 76.5%, respectively. 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.

5.3. Case study

In this section, we provide a case study of "Boston House Price Forecast" to further demonstrate the Mini-FL work process and evaluate the effectiveness of Mini-FL in the real world. *Case study setup* **Dataset and Data Deviation**: Boston house price dataset (Harrison and Rubinfeld, 1978) records 13 features (e.g., crime rate, pupil-teacher ratio, etc.) of 506 sample houses in Boston. Although Boston house price (Harrison and Rubinfeld, 1978) has not directly recorded the address of each property, the feature DIS (i.e., weighted distances to Boston employment centers) could be alternatively regarded as the address of each house. To simulate the federated learning scenario that different clients participating in the learning task at different addresses, we equally divide the whole DIS range into ten sub-ranges; according to the DIS subrange, all data are assigned into these ten groups subsequently, Table 10 illustrates the data deviation.

Model architecture and FL system settings: We train a deep neural network to predict the Boston house price; this model consists of three dense layers with two Relu activations: Dense + Relu (13×32), Dense + Relu (32×16), Dense (16×1).

We set 0.01, 10, and 10 as the learning rate, batch size, and epoch, respectively, and train the global model 300 iterations. We assume that 12 clients (from different addresses) participate in the learning task, and 25% are malicious.

Adversary, FL/Mini-FL methods, and evaluation metric: We use Median (Yin et al., 2018) and Mini-Median as the control group and select Reverse attack (Biggio et al., 2012) ($t = 10$) and a state of art attack Fall attack (Moreno-Torres et al., 2012) ($eps = 10$) to evaluate the robustness of each method. We use the loss value of the global model as the evaluation metric; specifically, the FL method is more robust if its global model achieves a lower loss under attacks.

*Mini-FL*We follow the Mini-FL proposed in Section 4. As only the Geo-feature (DIS) is available in this case, we set Geo-feature (DIS) as the grouping feature and perform the "elbow method" for all gradients received to detect the cluster. The 'elbow method' shows the gradients received can be divided into 6 clusters; Fig. 7 illustrates the "elbow method" outcome. Based on the result, we further divide the clients into 6 groups and generate the grouping policy; Table 11 illustrates the grouping policy.

In further iterations, we follow the grouping policy to distribute the gradients received in different groups and perform the Median method to generate the grouped model, respectively. Then, we utilize the weighted mean proposed in Section 4 to update the global model and finish each iteration.

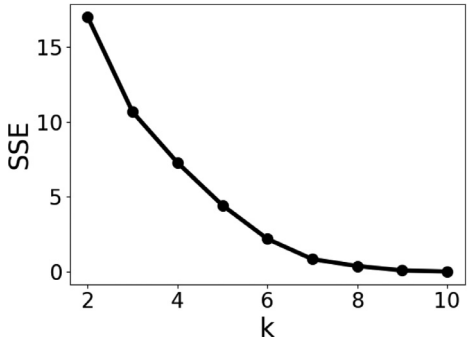

Fig. 7. Illustration of the "elbow method" outcome.

*Experimental results*The experimental results show our Mini-FL method is more robust and achieves a lower loss. Specifically, the Median achieves 31 and 47 global loss under the reverse and Fall attacks; but Mini-Median decreases the global loss to 26 and 24, respectively. Fig. 8 illustrates the global loss of the Median/Mini-Median method against Reverse and Fall attacks.

6. Discussion and future work

Mini-FL and CFL: We note that the CFL (Sattler et al., 2020) algorithm has been designed based on a strong assumption that "only the largest cluster is benign and all other clusters are adversarial", which is valid only under the perfect IID settings. Once the benign clients behave heterogeneously, the expected large cluster will separate into several sub-clusters, putting CFL in the dilemma of selecting. Instead of identifying and separating the adversary, our Mini-FL introduces clustering methods to reduce the attack surface. Furthermore, Mini-FL considers a more practical Non-IID scenario; we demonstrate that Mini-FL can effectively enhance the robustness of FL even though several (more than one) benign clusters exist.

Mini-Krum and Bulyan: Mini-Krum and Bulyan (Guerraoui et al., 2018) are different, although both of them rely on performing Krum and trimmed-mean methods. Specifically, Mini-Krum performs Krum by group and generates the weighted average as the global model. In contrast, Bulyan (Guerraoui et al., 2018) globally performs Krum n times to select n gradients and performs Trimmed-mean to generate the global model. As Bulyan (Guerraoui et al., 2018) 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: In this research, we select Geo-feature, Time-feature, and User-feature as the grouping feature candidates as they are the most common source of Non-IID in the real world and available to be collected in most scenarios without breaking participant's privacy. Except for the scenarios aforementioned in Section 1, the Geo-feature derived scenarios may further include: natural disaster exploration (Baghbani et al., 2022), customer taste analyzing (Ma et al., 2022a), etc. The Time-feature derived scenarios may include: pandemic pattern evaluation (Abildgren, 2022; Jiang et al., 2023), consumer behavior preferences (Pollak

Table 11
Illustration of the grouping policy.

	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
Client ID	C1, C2, C3	C4	C5	C6	C7, C8, C9, C10, C11	C12

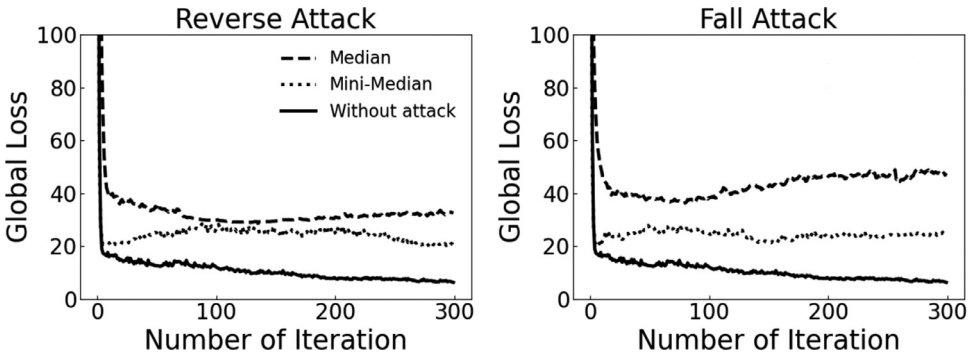

Fig. 8. The robustness comparison between Median and Mini-Median.

et al., 2022; Reshi et al., 2023), road monitoring (Gupta and Verma, 2022), etc. The User-feature derived scenarios may include (network) user profile generation (Kushwah and Ranga, 2022; Pujahari and Sisodia, 2022), etc. However, we note that the Non-IID source could be more complicated in some particular settings and even be a combination in some cases (Kairouz et al., 2021; Zhu et al., 2021; 2014). We leave investigating further to explore other Non-IID sources and combination possibilities and further improve the Mini-FL method.

Evaluation dataset: The effectiveness of our proposed Mini-FL framework is dependent on the grouping principle used to classify the received gradients, i.e., the definitions of Non-IID features and their boundaries, rather than the gradients themselves. To emulate the real-world Federated Learning process, we assign each gradient in our experiments with identified Non-IID features, namely IP address, Time Window, and User ID. We note that employing the same distribution strategy for Non-IID features will yield similar experimental results across different evaluation datasets.

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.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yanli Li: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Resources, Visualization, Writing – original draft, Writing – review & editing. **Dong Yuan:** Data curation, Resources, Supervision, Writing – review & editing. **Abubakar Sadiq Sani:** Visualization, Writing – review & editing. **Wei Bao:** Supervision, Visualization.

Data availability

Data will be made available on request.

References

Abildgren, K., 2022. Archival big data and the Spanish Flu in Copenhagen. Inf. Discov. Deliv. 50 (2), 133–141.
Baghbani, A., Choudhury, T., Costa, S., Reiner, J., 2022. Application of artificial intelligence in geotechnical engineering: a state-of-the-art review. Earth Sci. Rev. 228, 103991.
Biggio, B., Nelson, B., Laskov, P., 2012. Poisoning attacks against support vector machines. arXiv:1206.6389
Baruch, G., Baruch, M., Goldberg, Y., 2019. Proceedings of the 33rd International Conference on Neural Information Processing Systems 32, 8635–8645.
Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J., 2017. Machine learning with adversaries: byzantine tolerant gradient descent. Proceedings of the 31st International Conference on Neural Information Processing Systems, 118–128.
Briggs, C., Fan, Z., Andras, P., 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–9.
Briggs, C., Fan, Z., Andras, P., 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. doi:10.1109/IJCNN48605.2020.9207469.
Cao, X., Fang, M., Liu, J., Gong, N.Z., 2021. Fltrust: byzantine-robust federated learning via trust bootstrapping. ISOC Network and Distributed System Security Symposium (NDSS).
Cheng, Y., 1995. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17 (8), 790–799.
Collins, E., Stone, G., 2014. Understanding patterns of library use among undergraduate students from different disciplines. Evid. Based Libr. Inf. Pract. 9 (3), 51–67.
Damaskinos, G., El-Mhamdi, E.-M., Guerraoui, R., Guirguis, A., Rouault, S., 2019. Aggregathor: byzantine machine learning via robust gradient aggregation. Proc. Mach. Learn. Syst. 1, 81–106.
Deng, L., 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process. Mag. 29 (6), 141–142.
El-Mhamdi, E.-M., Guerraoui, R., Guirguis, A., Hoang, L.-N., Rouault, S., 2022. Genuinely distributed byzantine machine learning. Distrib. Comput. 1–27.
Fang, M., Cao, X., Jia, J., Gong, N., 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 1605–1622.
Ghosh, A., Chung, J., Yin, D., Ramchandran, K., 2020. An efficient framework for clustered federated learning. Adv. Neural Inf. Process. Syst. 33, 19586–19597.
Goodfellow, I. J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572
Guerraoui, R., Rouault, S., et al., 2018. The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning. PMLR, pp. 3521–3530.
Gupta, H., Verma, O.P., 2022. Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach. Multimed. Tools Appl. 81 (14), 19683–19703.
Harrison Jr., D., Rubinfeld, D.L., 1978. Hedonic housing prices and the demand for clean air. J. Environ. Econ. Manag. 5 (1), 81–102.
Hsieh, K., Phanishayee, A., Mutlu, O., Gibbons, P., 2020. The non-IID data quagmire of decentralized machine learning. In: International Conference on Machine Learning. PMLR, pp. 4387–4398.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B., 2018. Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: 2018 IEEE symposium on security and privacy (SP). IEEE, pp. 19–35.
Jiang, F., Zhao, Z., Shao, X., 2023. Time series analysis of COVID-19 infection curve: achange-point perspective. J. Econom. 232 (1), 1–17.
Jiang, W., Li, H., Liu, S., Ren, Y., He, M., 2019. A flexible poisoning attack against machine learning. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6.
Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al., 2021. Advances and open problems in federated learning. Found. Trends® Mach. Learn. 14 (1–2), 1–210.
Kim, Y., Al Hakim, E., Haraldson, J., Eriksson, H., da Silva, J.M.B., Fischione, C., 2021. Dynamic clustering in federated learning. In: ICC 2021-IEEE International Conference on Communications. IEEE, pp. 1–6.
Kodinariya, T.M., Makwana, P.R., 2013. Review on determining number of cluster in *k*-means clustering. Int. J. 1 (6), 90–95.
Kushwah, G.S., Ranga, V., 2022. Detecting DDoS attacks in cloud computing using extreme learning machine and adaptive differential evolution. Wirel. Pers. Commun. 124 (3), 2613–2636.
Li, C., Li, G., Varshney, P.K., 2022. Federated learning with soft clustering. IEEE Internet Things J. 9 (10), 7773–7782. doi:10.1109/JIOT.2021.3113927.
Li, Q., Diao, Y., Chen, Q., He, B., 2022. Federated learning on non-IID data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, pp. 965–978.
Li, Y., Sani, A.S., Yuan, D., Bao, W., 2022. Enhancing federated learning robustness through clustering non-IID features. In: Proceedings of the Asian Conference on Computer Vision (ACCV) Workshops, pp. 41–55.
Lu, J., Sibai, H., Fabry, E., 2017. Adversarial examples that fool detectors. arXiv:1712.02494
Ma, C., Ma, B., Wang, J., Wang, Z., Chen, X., Zhou, B., Li, X., 2022. Geographical origin identification of chinese white teas, and their differences in tastes, chemical compositions and antioxidant activities among three production regions. Food Chem. 16, 100504.
Ma, X., Zhu, J., Lin, Z., Chen, S., Qin, Y., 2022. A state-of-the-art survey on solving non-IID data in federated learning. Future Gener. Comput. Syst. 135, 244–258.
Ma, Z., Ma, J., Miao, Y., Li, Y., Deng, R.H., 2022. Shieldfl: mitigating model poisoning attacks in privacy-preserving federated learning. IEEE Trans. Inf. Forensics Secur. 17, 1639–1654.
McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.
Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F., 2012. A unifying view on dataset shift in classification. Pattern Recognit. 45 (1), 521–530.
Nguyen, H.T., Sehwag, V., Hosseinalipour, S., Brinton, C.G., Chiang, M., Poor, H.V., 2020. Fast-convergent federated learning. IEEE J. Sel. Areas Commun. 39 (1), 201–218.
Park, J., Han, D.-J., Choi, M., Moon, J., 2021. Sageflow: robust federated learning against both stragglers and adversaries. Adv. Neural Inf. Process. Syst. 34, 840–851.
Pollak, F., Markovic, P., Vachal, J., Vavrek, R., 2022. Analysis of e-consumer behavior during the COVID-19 pandemic. In: Intelligent Processing Practices and Tools for E-Commerce Data, Information, and Knowledge, pp. 95–114.
Pujahari, A., Sisodia, D.S., 2022. Item feature refinement using matrix factorization and boosted learning based user profile generation for content-based recommender systems. Expert Syst. Appl. 206, 117849.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1, 91–99.
Reshi, I.A., Dar, S.A., Ansar, S.S., 2023. An empirical study on the factors affecting consumer behavior in the fast-food industry. J. Account. Res., Util. Finance Digit. Assets 1 (4), 376–381.
Sattler, F., Müller, K.-R., Wiegand, T., Samek, W., 2020. On the byzantine robustness of clustered federated learning. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 8861–8865.
Sun, Z., Kairouz, P., Suresh, A. T., McMahan, H. B., 2019. Can you really backdoor federated learning? arXiv:1911.07963
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision And Pattern recognition, pp. 1–9.
Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L., 2020. Data poisoning attacks against federated learning systems. In: Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25. Springer, pp. 480–501.
Tomsett, R., Chan, K., Chakraborty, S., 2019. Model poisoning attacks against distributed machine learning systems. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, vol. 11006. SPIE, pp. 481–489.
Wu, H., Wang, P., 2021. Fast-convergent federated learning with adaptive weighting. IEEE Trans. Cognit. Commun. Netw. 7 (4), 1078–1088.
Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747
Xie, C., Koyejo, O., Gupta, I., 2020. Fall of empires: breaking byzantine-tolerant SGD by inner product manipulation. In: Uncertainty in Artificial Intelligence. PMLR, pp. 261–270.
Yin, D., Chen, Y., Kannan, R., Bartlett, P., 2018. Byzantine-robust distributed learning: towards optimal statistical rates. In: International Conference on Machine Learning. PMLR, pp. 5650–5659.
Zhu, H., Xu, J., Liu, S., Jin, Y., 2021. Federated learning on non-IID data: a survey. Neurocomputing 465, 371–390.
Zhu, T., Xiong, P., Li, G., Zhou, W., 2014. Correlated differential privacy: hiding information in non-IID data set. IEEE Trans. Inf. Forensics Secur. 10 (2), 229–242.

Yanli Li received the B.S. degree from Dalian University of Foreign Languages, Dalian, China, in 2017, the M.S. degree from University of Technology Sydney, Sydney, NSW, Australia, in 2020. He is currently working toward the Ph.D. degree in information engineering with the Faculty of Engineering, University of Sydney, Sydney, NSW, Australia. His primary research interests include federated learning, cybersecurity, and adversarial machine learning.

Dong Yuan received the B.Eng. and M.Eng. degrees from Shandong University, Jinan, China, in 2005 and 2008, respectively, and the Ph.D. degree from the Swinburne University of Technology, Melbourne, VIC, Australia, in 2012, all in computer science. He is currently a Senior Lecturer at The University of Sydney, Sydney, NSW, Australia. His research interests include machine learning, edge and cloud computing, and parallel and distributed systems.

Abubakar Sadiq Sani (Member, IEEE) received the M.Sc. degree in computer and network security from Middlesex University, London, U.K., in 2012, the Professional Education in applied cyber security from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2014, and the Ph.D. degree in engineering and information technologies from The University of Sydney, Sydney, NSW, Australia, in 2020.,He is a Certified Ethical Hacker and EC-Council Certified Security Analyst and worked in the industry for a few years. He is currently an Industry-Oriented Postdoctoral Researcher with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney. His primary research interests include cybersecurity and privacy for the Internet of Things, network and communication security, enterprise resource planning, secure software engineering, and blockchain for cybersecurity and its application in the Internet of Things.

Wei Bao received the B.E. degree in communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, the M.A.Sc. degree in electrical and computer engineering from The University of British Columbia, Vancouver, BC, Canada, in 2011, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2016. He is currently a Senior Lecturer at the School of Computer Science, The University of Sydney, Sydney, NSW, Australia. His research covers the area of network science, with a particular emphasis on edge computing and distributed machine learning. Dr. Bao received the Best Paper Awards in the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM) in 2013 and 2019 and the IEEE International Symposium on Network Computing and Applications (NCA) in 2016.