Security and Safety, Vol. 2, 2023006 (2023)
https://doi.org/10.1051/sands/2023006
sands.edpsciences.org

Security and Safety in Unmanned Systems

Research Article

Open Access

Information Network

MPHM: Model poisoning attacks on federal learning using historical information momentum

Lei Shi[1], Zhen Chen[1], Yucheng Shi[2], Lin Wei[1], Yongcai Tao[3], Mengyang He[1], Qingxian Wang[1], Yuan Zhou[4], and Yufei Gao[1,*]

[1] *School of Cyber Science and Engineering, Zhengzhou University; SongShan Laboratory, Zhengzhou 450000, China*
[2] *College of Intelligence and Computing, Tianjin University, 300072 Tianjin, China*
[3] *School of Computer and Artificial Intelligence, Zhengzhou University, 450000 Zhengzhou, China*
[4] *Zhengzhou Zhengda Information Technology Co., Ltd, 450001 Zhengzhou, China*

Received: 22 December 2022 / Revised: 9 March 2023 / Accepted: 20 April 2023 / Published online: 30 June 2023

Abstract Federated learning(FL) development has grown increasingly strong with the increased emphasis on data for individuals and industry. Federated learning allows individual participants to jointly train a global model without sharing local data, which significantly enhances data privacy. However, federated learning is vulnerable to poisoning attacks by malicious participants. Since federated learning does not have access to the participants' training process, *i.e.*, attackers can compromise the global model by uploading elaborate malicious local updates to the server under the guise of normal participants. Current model poisoning attacks usually add small perturbations to the local model after it is trained to craft harmful local updates and the attacker finds the appropriate perturbation size to bypass robust detection methods and corrupt the global model as much as possible. In contrast, we propose a novel model poisoning attack based on the momentum of history information (MPHM), that is, the attacker makes new malicious updates by dynamically crafting perturbations using the historical information in the local training, which will make the new malicious updates more effective and stealthy. Our attack aims to indiscriminately reduce the testing accuracy of the global model with minimal information. Experiments show that in the classical defense case, our attack can significantly corrupt the accuracy of the global model compared to other advanced poisoning attacks.

Keywords Federated learning, Poisoning attacks, Security, Privacy

Citation Chen Z, Shi YC, et al. MPHM: Model poisoning attacks on federal learning using historical information momentum. Security and Safety 2023; **2**: 2023006. https://doi.org/10.1051/sands/2023006

1 Introduction

With the rapid development of big data and artificial intelligence, the industry is increasingly concerned about data privacy. As a result, data, which is the "nutrition" of learning algorithms, is difficult to be fully shared [1, 2]. For example, it is difficult to fully share data between different banks or between e-commerce platforms and banks due to security concerns. In industrial application scenarios, few enterprises are willing to share their data resources due to attention to data privacy and security, which has become a worldwide trend. Countries are also strengthening the protection of data security and privacy, as evidenced by the EU's implementation of the General Data Protection Regulation (GDPR) bill in 2018. As a result,

* Corresponding authors (email: yfgao@zzu.edu.cn)

the issue of "data islands" [3] has become a serious problem. Even for individual participants, there are concerns that the privacy risks associated with outsourcing local data sets to service providers may outweigh the benefits of convenient online services [4].

The emergence of federated learning (FL) [5] has attracted significant attention from both academia and industry. FL allows participants to conduct joint training without sharing their local data. In a federated learning framework, multiple participants train their data locally, and the central server iteratively updates the global model by collecting the parameters of the local model. Because private data does not leave the local device, FL is considered an innovative approach to protecting user data privacy [6]. FL has been applied in various technology areas involving security-sensitive information, such as edge computing [7, 8], medical diagnosis [9, 10], and autonomous driving technologies [11, 12].

Despite the advantages of federated learning mentioned above, it still faces security threats, such as poisoning attacks [13–15]. There are several reasons for this. Firstly, in the federated learning framework, the cloud server does not have access to the participant's local data or training process, which means that malicious participants can upload incorrect model updates to corrupt the global model [16]. For example, an internal attacker can train a poisoned model with modified training data, effectively reducing the accuracy of the global model. Secondly, since the data of each participant may not be identically and independently distributed, the differences between the local updates generated by participants can be significant enough to make it difficult to detect malicious updates through anomaly detection [17].

Poisoning attacks on FL. The potential presence of dishonest participants in FL training makes FL vulnerable to poisoning attacks [18]. Attackers can compromise the global model of FL by uploading malicious updates [19]. The targets of poisoning attacks can be divided into two types, untargeted attacks [13, 19–23] that aim to reduce the accuracy of the global model on any test input, and targeted attacks [15, 24–27] that aim to reduce the utility of the global FL model on the attacker's selected inputs.

Our work. Currently, the trend in attacking FL is to use malicious poisoning updates that replace normal local updates in a poisoning attack. Typically, attackers compute a benign reference aggregation using some benign data samples they know, then they compute a malicious perturbation vector, and finally, they compute their malicious model update by adding perturbations to the benign reference aggregation to avoid detection by the robust aggregation rules. Current research has focused on the scale of the perturbation vector when producing malicious updates, and the choice of the perturbation vector is often straightforward, such as using the unit vector. This paper proposes a novel model poisoning attack on FL, called the momentum of historical information-based poisoning attack (MPHM). In this attack, the attacker gathers historical information from FL training, dynamically crafts malicious perturbations in each round of FL training, and uses them to build more covert malicious updates. By leveraging this information, the attacker can make their malicious updates harder to detect and mitigate, effectively bypassing FL defense mechanisms. The experimental results demonstrate that our attack can significantly reduce the accuracy of the FL global model compared to other advanced poisoning attacks.
The contributions of this paper are summarized as follows:

- We studied the effect of momentum accumulation of historical information on the production of malicious updates.
- We propose a new poisoning attack MPHM on FL which is dedicated to reducing the accuracy of the FL global model.
- Experiments show that our attack can effectively reduce the accuracy of the FL global model using robust aggregation rules on the CIFAR10 and FEMNIST datasets.

The rest of the paper is organized as follows. In Section 2, we present the background and related work on federated learning and poisoning attacks. In Section 3, we introduce the threat model, and we introduce our attack framework in Section 4. In Section 5, we give the experimental setup, and in Section 6, the results and discussion are given. Finally, we conclude the paper and present future work in Section 7.

2 Related work

2.1 Federated learning

In the FL [19] setting, we assume that there are n clients, which jointly train a global model. During FL training, each client gets the global model sent from the server, computes the stochastic gradient based on the local dataset, and sends it to the server. In detail, in the t-th round of FL, the server sends the latest global model θ^t to the client, then the k-th client computes the stochastic gradient $\nabla_k^t = \frac{\partial \mathcal{L}(\theta^t, b)}{\partial \theta^t}$ using local data, where $\mathcal{L}(\theta^t, b)$ refers to the loss function and b refers to the sample. Then, the client sends ∇_k^t to the cloud server. The server aggregates these gradients and obtains the aggregated gradient ∇^t as follows:

$$\nabla^t = \mathcal{A}(\nabla_1^t, \nabla_2^t, \nabla_3^t, \ldots, \nabla_n^t) \tag{1}$$

where $\mathcal{A}$ is the server's aggregation rule, and then the server computes the global model θ^{t+1} by the optimizer *e.g.*, SGD, and broadcasts it to the selected clients for the next round of FL training. These steps are repeated until the global model converges.

2.2 Several popular aggregation rules in FL

Google proposed a federated average aggregation algorithm [5], however, researchers [28] have shown that non-robust aggregation algorithms can lead to the manipulation of the global model at will even if there is only one malicious client. Therefore, multiple Byzantine-robust aggregation algorithms [18, 21, 28, 29] have been proposed to combat poisoning attacks. Next, we will introduce four common Byzantine-robust aggregation rules.

Krum. Krum's [28] algorithm is proposed based on the intuition that malicious gradients are far away from benign gradients. In an FL system with n clients, suppose there are m malicious clients. Krum calculates the distance sum of each client to the $n - m - 2$ clients that are closest to itself in the squared Euclidean norm space and then chooses the gradient of the one with the smallest sum as the gradient of the global model.

Trimmed-mean. Trimmed-mean [18, 29] is a dimension-level aggregation method, which aggregates each dimension of the input gradients separately. For a given dimension j, Trimmed-mean sorts the j-th dimensional gradients of all clients, *i.e.*, sort $\nabla_{1j}, \nabla_{2j}, \ldots, \nabla_{nj}$, where ∇_{ij} is the parameter of the j-th dimension of the i-th client. Then the largest and smallest β values are removed and the remaining $n - 2\beta$ values are aggregated equally as the value of the j-th dimension, where β is the specified value, *e.g.*, $\beta = m$. This procedure is carried out for each dimension.

Bulyan. The Euclidean distance between different clients may be largely influenced by a single-dimensional parameter, which causes Krum not to aggregate the model well [21]. Thus Mhamdi *et al.* [21] proposed Bulyan, which can be seen as a variant combination of Krum and Trimmed-mean. Specifically, Bulyan first iteratively uses Krum to select $\kappa (\kappa \leq n - 2m)$ clients' parameters and then uses the variant Tr to aggregate the κ clients' parameters.

Median. Median [18, 29] is also a dimensional-level aggregation algorithm, which aggregates each dimension of the input gradients separately. As the name of the algorithm suggests, for a given dimension j, Median sorts all client j-th dimension parameters and selects their median value as the j-th dimension parameter, with each dimension making such a selection.

2.3 Poisoning attack on federated learning

Due to the influence of potentially dishonest customers, studies [30–32] have shown that FL is vulnerable to poisoning attacks. Poisoning attacks can be divided into two categories according to the target of the attacker: untargeted poisoning attacks [13, 20, 22, 23] and targeted poisoning attacks [15, 25, 26, 33]. Untargeted poisoning attacks refer to the attacker's efforts to reduce the testing accuracy of the global model. Target poisoning attacks refer to the attacker making the global model output low accuracy for specific inputs and maintaining high accuracy for other inputs.

According to the attacker's capability, poisoning attacks can be divided into two categories: data poisoning attacks [31, 34–36] and model poisoning attacks [13, 22, 23, 37]. Data poisoning attacks mean that the attacker cannot directly manipulate the client parameters uploaded to the server, but can only indirectly modify the uploaded client parameters by crafting malicious local datasets. Whereas in model poisoning attacks, the attacker can directly manipulate the client parameters uploaded to the server to attack FL training. In this paper, we focus on untargeted poisoning attacks on FL.

Currently, model poisoning attacks in FL are commonly performed by attackers who compute benign parameters of the local client, add a perturbation vector, and upload the malicious parameters to the server to poison the global model. Baruch *et al.* [20] have proposed a method of compromising the global model by adding tiny attacks to the local updates. Fang *et al.* [13] propose an optimization objective of adding a perturbation vector to the local updates to craft malicious updates. Shejwalkar *et al.* [19] optimize the scale of the perturbation vector in their approach. Later, they [22] argue that having an excessive number of compromised clients is not reasonable in this setting. Recently, Cao *et al.* [23] have proposed a new approach for model poisoning in FL, which involves injecting fake clients to poison the model and effectively mitigates the problem of having an excessive number of compromised clients.

3 Threat model

3.1 Attacker's goal

In this paper, the attacker's goal is to reduce the test accuracy of the global model by crafting malicious gradients, for all inputs without exception. That is the untargeted model poisoning attack.

3.2 Attacker's capability

In an FL training framework with n clients, we assume that the attacker controls m clients. Here we call the controlled clients as malicious clients and the uncontrolled as clients benign clients. The number of malicious clients is less than benign clients. The attacker can modify the gradients of malicious clients at will, but cannot control benign clients. Also, the attacker can control the communication between each malicious client.

3.3 Attacker's knowledge

The attacker's background knowledge can be described in two dimensions: the aggregation rule and the gradients of benign clients.

Aggregation rule. The background knowledge of the attacker can be divided into two categories based on whether they know the server aggregation rule or not. In FL training, the server can choose whether to make its aggregation rule public or not. Exposing the aggregation rule can increase the transparency of FL, but may increase the corresponding risk, such as an attacker can set up focused poisoning attacks based on the aggregation rule. In Fang attacks [13], knowledge of the server's aggregation rule is assumed.

Benign gradients. The background knowledge of the attacker can also be divided into two cases based on whether the gradients of the benign clients are known or not. Knowing that the gradients of all clients are strong background knowledge, the attacker can make the crafted malicious gradients more stealthy. In the LIE attacks [20], the gradients of understanding the benign clients are assumed not to be known.

It can be seen that the attacker who knows the aggregation rule and the benign gradients is the strongest adversary against FL. But in practice such conditions are harsh. Therefore, our attack does not require these conditions, we focus on the weakest adversary condition, *i.e.*, the attacker who knows neither the aggregation rule nor the benign gradient.

4 Our attack

4.1 Framework

The overall framework of our method is shown in Figure 1. In round t of FL training, the server sends the global model θ^t for this round to each client; in step 2, the benign clients compute the stochastic gradients

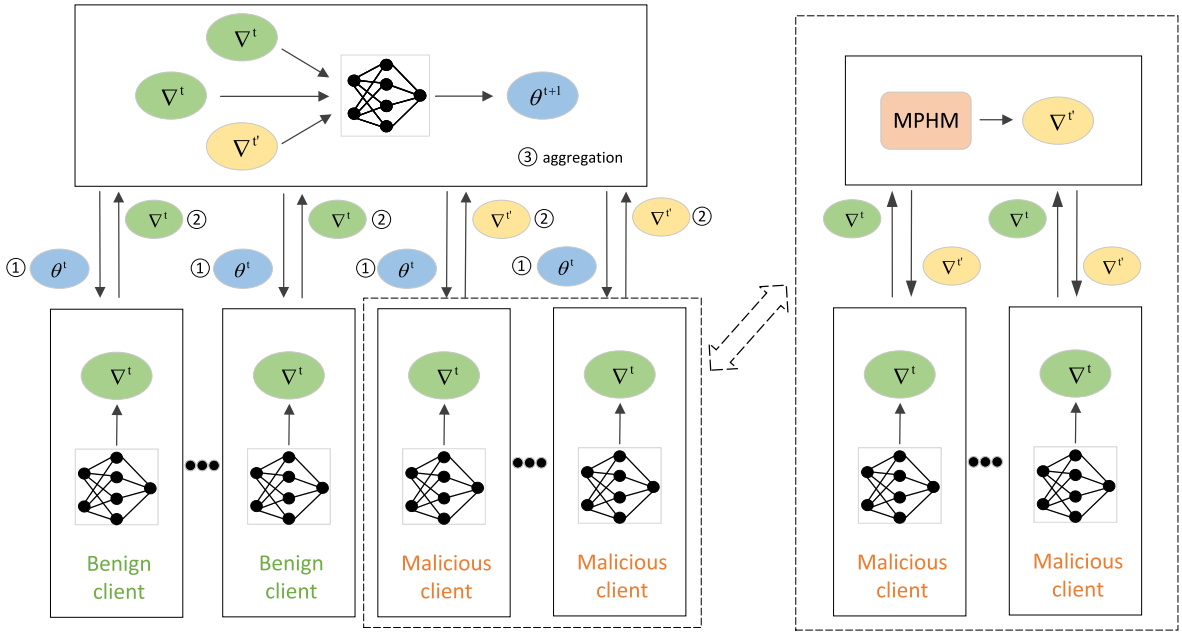


Figure 1. Overall framework of our MPHM. In step 1, the server sends the global model θ^t for this round to each client; in step 2, the benign clients compute the stochastic gradients ∇^t based on the model and upload them to the server. And the malicious clients, after computing the gradients, compute the malicious gradients $\nabla^{t'}$ for by our MPHM, and then upload them to the server; in step 3, the server aggregates these gradients and computes new global model θ^{t+1} for next round.

∇^t based on the model and upload them to the server. And the malicious clients, after computing the gradients, communicate together and compute the malicious gradients $\nabla^{t'}$ for this round by our MPHM, and then upload them to the server; finally, the server aggregates these gradients and computes a new global model θ^{t+1}. The new global model θ^{t+1} will replace the original global model θ^t for the next round of FL training.

4.2 MPHM

In this section, we introduce our optimization objective and then our attack method.

The objective of the attacker is to manipulate the malicious clients to deliver malicious gradients that can evade detection in each round of FL training. In epoch t, the attacker produces a malicious gradient denoted as $\nabla^{t'}$, while the average of benign gradients is represented by ∇_b. The malicious gradient is computed as $\nabla^{t'} = \nabla_b - \lambda\nabla_p^t$, where ∇_p^t is the perturbation vector in epoch t, and λ is the perturbation coefficient. In order to make the malicious gradient effective, similar to [13], we propose the optimization objective:

$$\begin{aligned} &\max_{\nabla_M^{t'}} \|\nabla_b - \nabla_M^{t'}\| \\ \text{subject to: } &\nabla_b = \mathcal{A}_{\text{avg}}(\nabla_1^t, \ldots, \nabla_m^t, \nabla_{m+1}^t, \ldots, \nabla_n^t) \\ &\nabla_M^{t'} = \mathcal{A}(\nabla_1^{t'}, \ldots, \nabla_m^{t'}, \nabla_{m+1}^t, \ldots, \nabla_n^t) \end{aligned} \tag{2}$$

where $\nabla_{\{i\in[n]\}}^t$ refer to the benign gradients known by the attacker in round t, $\nabla_{\{i\in[m]\}}^{t'}$ refer to the malicious gradients made by the attacker in round t, $\|\cdot\|$ refers to the l_2 norm, $\mathcal{A}_{\text{avg}}$ refers to the mean aggregation and $\mathcal{A}$ refers to the server's aggregation rule.

To achieve this objective, the proposed approach is a poisoning attack based on historical information momentum (MPHM), which aims to make malicious updates more difficult to detect. To elaborate on MPHM, we first analyze the perturbation gradient ∇_p^t. Previous works [19] have selected three intuition-based perturbation vectors, namely the sign vector, unit vector, and standard deviation vector. The sign vector refers to $\nabla_p^t = \text{sign}(\nabla_b)$, the unit vector refers to $\nabla_p^t = \frac{\nabla_b}{\|\nabla_b\|}$, and the standard deviation vector

refers to $\nabla_p^t = \text{std}(\nabla_{\{i\in[n]\}})$, where sign() refers to the sign function and std() refers to the standard deviation function. These perturbation vectors are based solely on the information from the current training round. In contrast, in the MPHM attack, the attacker references the previous training information when crafting the perturbation vectors. Specifically, the attacker will use gradient information from the previous training rounds and accumulate this information momentum to the newly crafted perturbation vectors, making them more stealthy. Thus, we propose a new method for calculating the perturbation vector as follows:

$$\nabla_p^t = \nabla_b + \alpha \nabla_p^{t-1} \tag{3}$$

Where ∇_b is the mean value of the known gradient of the attacker, ∇_p^{t-1} is the perturbation vector in epoch $t-1$, and α is the decay factor.

To ensure that the malicious gradients bypass the aggregation rules, the magnitude of the added perturbation needs to be regulated. In our approach, we search for the optimal perturbation coefficient λ within a predefined range, based on the experience of [22]. Additionally, a higher standard deviation allows for a higher magnitude of perturbation to be introduced, so we use the deflating scale $\lambda = \frac{\|\sigma\|}{\|\nabla_p^t\|}$, where σ represents the standard deviation $\sigma = \text{std}(\nabla_{i\in[m]})$.

Algorithm 1 describes how MPHM calculates the malicious updates. The attacker calculates their mean $\nabla_b = \frac{1}{m}\sum_{i\in[m]} \nabla_i$ based on the benign gradients in hand and then calculates the perturbation gradients ∇_p^t to be added based on the perturbation gradients from the previous round. After that, the scaling factor λ of the perturbation gradient is calculated. Finally, the malicious updates $\nabla^{t'} = \nabla_b - \lambda \nabla_p^t$ uploaded by the malicious clients in this round are successfully constructed.

Algorithm 1 MPHM gets malicious updates.

Input: Benign gradients $\nabla_{\{i\in[m]\}}$ under attacker control, perturbation gradients ∇_p^{t-1} of epoch $t-1$, decay factor α;

Output: Malicious updates exported by malicious clients $\nabla^{t'}$;

1: $\nabla_b = \frac{1}{m}\sum_{i\in[m]} \nabla_i$;
2: $\nabla_p^t = \nabla_b + \alpha \nabla_p^{t-1}$;
3: $\sigma = std(\nabla_{\{i\in[m]\}})$;
4: $\lambda = \frac{\|\sigma\|}{\|\nabla_p^t\|}$;
5: **return** $\nabla^{t'} = \nabla_b - \lambda \nabla_p^t$;

5 Experimental setup

5.1 Experimental environment

The experiments in this study were conducted on a server equipped with an Intel Xeon Silver 4210 CPU, 64GB RAM, NVIDIA Tesla T4 GPU, and 16GB RAM, running on the Ubuntu 20.04 server operating system. The FL experiments were implemented using the PyTorch framework.

5.2 Datasets and model architectures

The validation of the proposed attack is conducted on two visual domain datasets, *i.e.*, CIFAR10 [38], and FEMNIST [39, 40].

CIFAR10. CIFAR10 [38] is a 10-class color image dataset with 60 000 images. 50 000 of them are used for training examples and 10 000 are used for testing examples, the size of the images is 32×32. To create non-IID data, the 50 000 training examples were divided into 100 clients using the Dirichlet distribution [41] with a coefficient of $\alpha = 0.5$. In each epoch of Federated Learning (FL) training, 40 clients were randomly selected to participate. The Alexnet [42] architecture was used for FL training on this dataset.

FEMNIST. FEMNIST [39, 40] is a dataset containing grayscale images of 62 different classes, containing a total of 805 263 samples. The image size in FEMNIST is 28×28, and 62 classes are character types (10

digits, 26 lowercase, 26 uppercase). [40] divides it into 3500 clients, each client contains a sample mean of 226.83 and variance of 88.94, which is a non-IID FL setting. During each epoch of FL training, 60 clients were randomly selected to participate. The FL global model was constructed using a simple CNN framework contains two convolutional layers and three fully connected layers.

5.3 FL and attack settings

For the CIFAR10 dataset, we use the Alexnet architecture for training. The optimizer used is Adam, the batch size is 64, the number of training rounds is 1000, and the learning rate per training round is 0.001×0.998^t. For the FEMNIST dataset, we use CNN architecture training. The optimizer used is SGD, the number of training rounds is 1200, using the entire data from clients per batch, and the learning rate per training round is 0.2×0.998^t.

By default, the malicious client's ratio is set to 20%, *i.e.*, $m/n = 0.2$, and the percentage of malicious clients is fixed in each round of FL training. In the attack setup, except for the Fang attack, the attacker's knowledge is that neither benign updates nor aggregation rules are known. Due to Fang's algorithm setup, Fang needs to know the aggregation rule, which is a stronger adversarial setup compared to other attacks. In addition, the factor α in our attack is 0.5, unless otherwise specified.

5.4 Baseline attacks

LIE. Little is enough (LIE) [20] attack, as his name implies, jeopardizes FL training by adding small amounts of noise to each dimension of the benign gradients. Specifically, the attacker calculates the mean μ and standard deviation σ based on the benign gradient he himself possesses. Then the coefficient z is calculated based on the number of malicious and benign clients, and finally, the update of malicious clients is calculated $\mu + z\sigma$.

Fang. Fang *et al.* [13] propose a generic framework for FL poisoning attacks. It computes the benign gradients' mean μ and then computes the perturbation vector ∇_p. Denote the benign parameter as ∇_b, The final poisoning update of the malicious clients as $\nabla_M = \nabla_b - \lambda\nabla_p$ is derived by solving for the coefficient λ.

Min-Max. Shejwalkar *et al.* [19] propose a generic framework for FL poisoning attacks. Similar to Fang [13], the update of the malicious client in Min-Max is $\nabla_M = \nabla_b - \lambda\nabla_p$. It uses the constraint "so that the maximum distance between the malicious gradients and any other gradients is an upper bound on the maximum distance between any two benign gradients" to solve for a more appropriate factor λ.

Min-Sum. Min-Sum is another method in [19] that solves for the coefficient λ with the constraint that "the upper bound of the sum of the squares of the distances between the malicious gradients and all the benign gradients is the sum of the squares of the distances between any benign gradient and the other benign gradients". The specific details are in [19].

5.5 Evaluation metric

The untargeted poisoning attack is designed to decrease the testing accuracy of the global model, and the effectiveness of the attack is evaluated using the testing accuracy loss δ as the metric. Specifically, the notation $\mathcal{P}$ represents the test accuracy of the global model without any attack, while $\mathcal{P}'$ represents the test accuracy of the global model with the attack. Therefore, the evaluation index is defined as $\delta = \mathcal{P} - \mathcal{P}'$.

6 Results and discussion

6.1 Impact of attacks on robust aggregation rules

In this section, the impact of the proposed attacks on robust FL training is explored in comparison to baseline attacks. The training process of FL with various robust aggregation rules under multiple attacks is presented in Figure 2, while the impact of different attacks on the robust FL is summarized in Table 1.

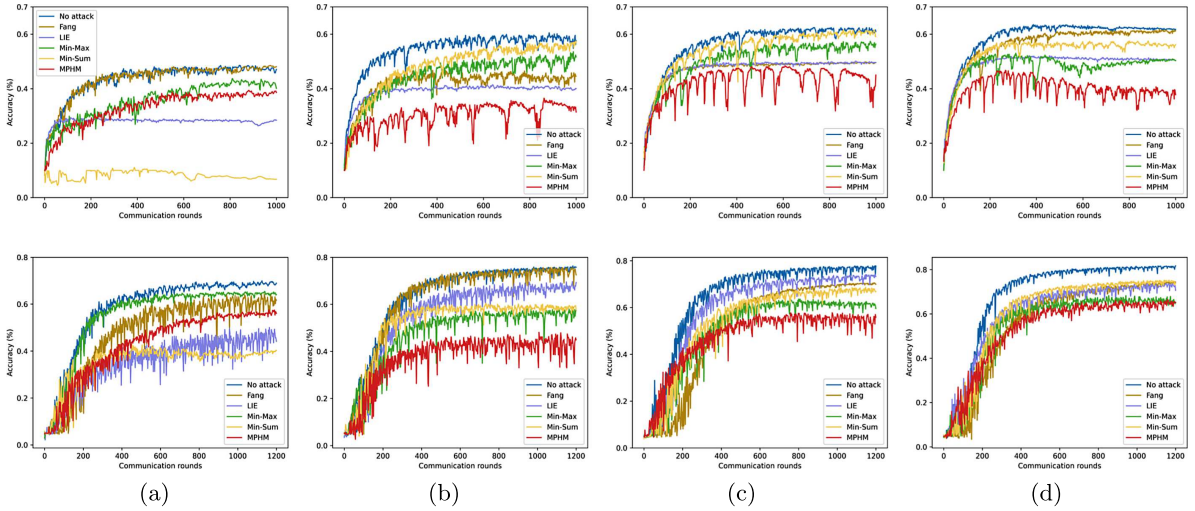


Figure 2. FL training process using different robust aggregation rules. The horizontal coordinate represents the number of training rounds and the vertical coordinate represents the test accuracy. The datasets are CIFAR10 (first row) and FEMNIST (second row). (a) krum, (b) Bulyan, (c) Median, (d) Trimmed-mean.

Table 1. Impact of different attacks on FL training using different robust aggregation rules.

Dateset		CIFAR10				FEMNIST			
Aggregation rule		Krum	Bulyan	Median	Trimmed-mean	Krum	Bulyan	Median	Trimmed-mean
No attack accuracy		48.74	60.51	62.61	63.60	69.95	76.23	77.89	81.68
Attack Impact δ	Fang	0.19	13.59	12.75	1.90	5.11	0.41	7.19	7.16
	LIE	19.29	19.09	12.40	11.42	19.29	6.10	3.74	7.78
	Min-Max	4.98	6.94	5.0	10.24	4.54	17.34	14.12	12.89
	Min-Sum	**37.60**	2.97	0.81	5.98	**27.10**	15.81	9.20	6.29
	Our attack	9.38	**24.4**	**14.23**	**16.86**	12.23	**28.01**	**19.80**	**15.06**

From Figure 2, it can be observed that MPHM and baseline attacks both impact the robust aggregation rules. Specifically, on the CIFAR10 dataset, MPHM shows significant effectiveness against the Bulyan, Median, and Trimmed-mean aggregation rules, outperforming baseline attacks. However, when using the Krum aggregation rule, MPHM is less effective than Min-Max and LIE. We hypothesize that this may be due to the excessive scaling factor added by MPHM, which may interfere with the Krum algorithm's selection of the client update as the global model update. As can be observed from the curves in the figure, MPHM made the global model difficult to converge with the Bulyan, Median, and Trimmed-mean aggregation rules. These results demonstrate the superiority of MPHM in poisoning federated learning.

On the FEMNIST dataset, the MPHM attack has a significant impact on FL training with robust aggregation rules. Similar to CIFAR10, MPHM outperforms baseline attacks for FL training using Bulyan, Median, and Trimmed-mean aggregation rules, while being less effective than LIE and Min-Sum for FL training using Krum aggregation rules. The MPHM attack is particularly effective for Bulyan and Median aggregation rules, making it more difficult for the global model to converge.

From Table 1, it can be observed that using the classical defense, the MPHM attack can significantly reduce the accuracy of the global model. On the CIFAR10 dataset, the MPHM attack on Bulyan reduces the global accuracy by 24%, while the attack on Trimmed-mean reduces the global accuracy by 17%. On the FEMNIST dataset, the MPHM attack on Bulyan reduces the global accuracy by 28%, and the attack on Median reduces the global accuracy by 20%.

6.2 Impact of the proportion of malicious clients on FL

Figure 3 illustrates the impact of attacks on FL training with different percentages of malicious clients.

The percentage of malicious clients varies from 5% to 25%. On the CIFAR10 dataset, the effectiveness of the MPHM attack increases with the percentage of malicious clients, and in the Bulyan, Median, and

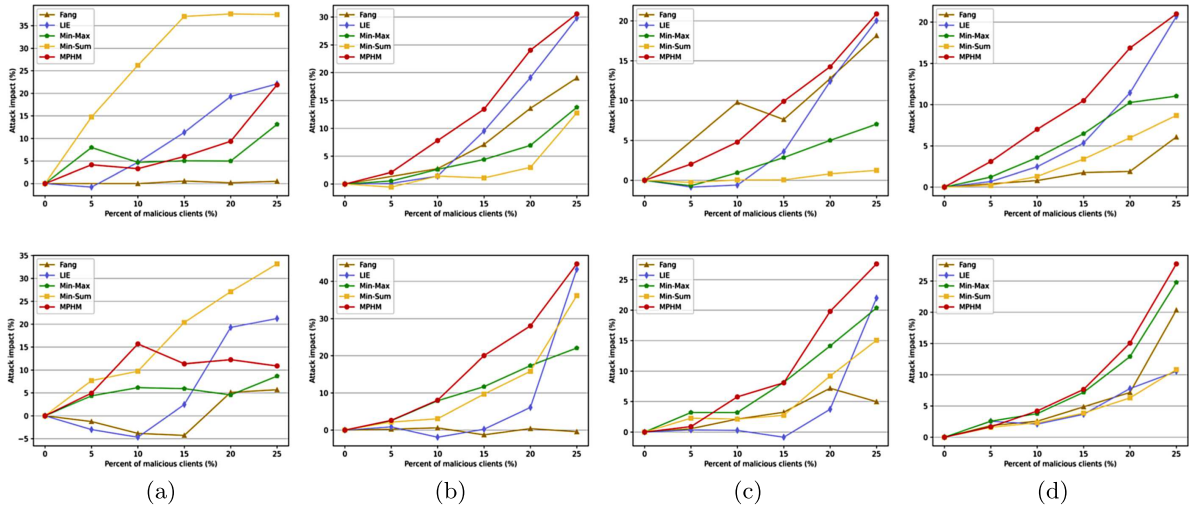


Figure 3. The effect of different attack proportions on the accuracy of the global model. The horizontal coordinate represents the percentage of malicious clients and the vertical coordinate represents the attack impact. The datasets used in this study are CIFAR10 (first row) and FEMNIST (second row). (a) krum, (b) Bulyan, (c) Median, (d) Trimmed-mean.

Trimmed-mean aggregation rule cases, the MPHM attack outperforms other attacks. On the FEMNIST dataset, the effect of the MPHM attack also increases with the proportion of malicious clients, except for the Krum aggregation rule case. In the Bulyan, Median, and Trimmed-mean aggregation rule cases, the MPHM attack is superior to other attacks. In addition, it can be seen from the figure that the MPHM attack is still effective for each aggregation algorithm with a small proportion of malicious clients, while part of the baseline attack is effective only with a large proportion of malicious clients. This indicates that the MPHM attack has better concealment.

6.3 Impact of the decay factor α

In this section, the impact of the decay factor on the MPHM attack is demonstrated. The results are shown in Figure 4, where the effect of different decay factors on the accuracy of the global model is presented.

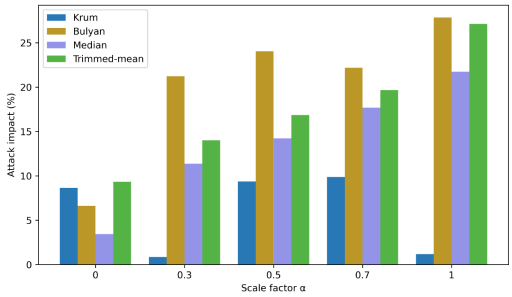
Our perturbation vector is $\nabla_p^t = \nabla_b + \alpha\nabla_p^{t-1}$. Noting that our perturbation vector degenerates to a unit perturbation vector when $\alpha = 0$, we use $\alpha = 0$ as the baseline for comparing the effectiveness of our attacks.

On CIFAR10, when $\alpha = 0$, the attack effect is significantly smaller than the other α values. As can be seen from Figure 4a, the attack effect increases with increasing α under Median and Trimmed-mean. $\alpha = 0.5$ is the most effective for the Krum aggregation rule and $\alpha = 1$ is most effective for the Bulyan aggregation rule. Overall, as α increases, our attacks are more effective for FL. At $\alpha = 0.5$, the attacks are significantly more effective for all four aggregation rules than at $\alpha = 0$.

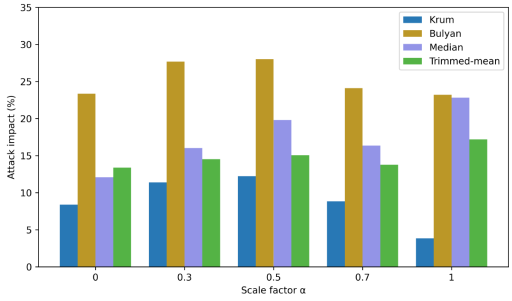
On FEMNIST, the attack effect increases and then decreases with increasing α. As can be seen from Figure 4b, with Krum and Bulyan, the attack effect increases and then decreases with increasing α and maximizes around $\alpha = 0.5$. At $\alpha = 1$, it is most effective for Median and Trimmed-mean aggregation rules. At $\alpha = 0.5$, the attack effect is significantly better for all four aggregation rules than at $\alpha = 0$. It can be seen that the momentum accumulation of historical information can effectively assist malicious updates to escape detection by robust aggregation rules. In addition, taking into account the individual datasets and aggregation rules, we take the default value of 0.5 for α.

6.4 Discussion

The experimental results demonstrate that the proposed MPHM can effectively disrupt the accuracy of the FL global model. While Fang *et al.* [13] proposed an optimization target for model poisoning attacks, their method requires more prior knowledge as the attacker needs to know the aggregation algorithm



(a)



(b)

Figure 4. Effect of different decay factors α on the test accuracy of the FL global model. (a) CIFAR10, (b) FEMNIST.

used by the FL architecture. Brauch *et al.* [20] proposed a simple and effective model poisoning attack, but its effectiveness is strongly influenced by the proportion of malicious clients. Shejwalkar *et al.* [19] proposed several optimization approaches for different prior knowledge cases, but their study lacks a thorough investigation of the direction of the perturbation added by local updates.

This paper proposes a new way of computing perturbations, and the experimental results show that elaborate malicious perturbations can make the malicious updates of attackers more covert. However, the effectiveness of the proposed attack in this paper still needs verification in large datasets and mega-federated learning frameworks due to the limitations of devices. The untarget model poisoning attack is still in the early stage of research, and more researchers are expected to join the research of attack and defense methods in FL.

7 Conclusion

In this work, we propose a new model poisoning attack on FL based on historical information momentum (MPHM). We use a setup where the attacker knows minimal information, and experiments show that our attack is effective compared to other advanced attacks in the face of classical defenses. Our approach focuses on the generation of perturbations, where we have found that carefully crafted malicious perturbations can enhance the surreptitious nature of the attacker's updates. We believe that there is significant research value in this area and will continue to focus on the generation of perturbation vectors in our future work.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability

We make data available on request through sending an email to the authors.

Authors' Contributions

Lei Shi constructed and drafted this paper. Zhen Chen researched the related work and contributed to the experimental design. Yucheng Shi contributed to the theory and revised the manuscript. Lin Wei, Mengyang He, and Yongcai Tao revised

and embellished the manuscript. Qingxian Wang and Yuan Zhou discuss the effectiveness of the method and correct the typos. Yufei Gao designed the whole structure of the paper.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions.

Funding

This work was supported in part by the National Key R&D Program of China (2020YFB1712401, 2018YFB1701400), the Nature Science Foundation of China (62006210, 62001284, 62206252), the Key Scientific and Technology Project of Henan Province of China (221100210100), the Key Project of Public Benefit in Henan Province of China (201300210500), the Research Foundation for Advanced Talents of Zhengzhou University (32340306), the Key Research Projects of Universities in Henan Province of China (7A520015, 21B520018), Fundamental Science Projects of Railway Police College (2020TJJBKY002), Advanced research project of SongShan Laboratory(YYJC022022001), The Key R&D and Promotion Project in Science and Technology of Henan (232102210154), and the Key Scientific and Technological Research Projects in Henan Province of China (192102310216).

References

[1] Yang Q, Yang L, Chen T and Yongxin Tong. Federated machine learning: Concept and applications. ACM Trans Intell Syst Technol 2019; **10**: 1–19.
[2] Li T, Sahu AK, Talwalkar A and Smith V. Federated learning: Challenges, methods, and future directions. IEEE Signal Process Mag 2020; **37**: 50–60.
[3] Savazzi S, Nicoli M, Bennis M, Kianoush S and Barbieri L. Opportunities of federated learning in connected, cooperative, and automated industrial systems. IEEE Commun Mag 2021; **59**: 16–21.
[4] Zhang K, Song X, Zhang C and Yu S. Challenges and future directions of secure federated learning: a survey. Front Comput Sci 2022; **16**: 1–8.
[5] McMahan B, Moore E, Ramage D, Hampson S and Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, PMLR, 2017, 1273–1282.
[6] Alazab M, Swarna Priya RM and Parimala M et al. Federated learning for cybersecurity: Concepts, challenges, and future directions. IEEE Trans Ind Inf 2022; **18**: 3501–3509.
[7] Doku R and Rawat DB. Mitigating data poisoning attacks on a federated learning-edge computing network. In: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2021, 1–6.
[8] Ahmed J, Razzaque MdA, Rahman MdM, Alqahtani SA and Hassan MM. A stackelberg game-based dynamic resource allocation in edge federated 5g network. IEEE Access, 2021; **10**: 10460–10471.
[9] Ma Z, Ma J, Miao Y, Liu X, Choo KKR and Deng R. Pocket diagnosis: Secure federated learning against poisoning attack in the cloud. IEEE Trans Serv Comput, 2021; **15**: 3429–3442.
[10] Kuo TT and Pham A. Detecting model misconducts in decentralized healthcare federated learning. Int J Med Inf, 2022; **158**: 104658.
[11] Niknam S, Dhillon HS and Reed JH. Federated learning for wireless communications: motivation, opportunities, and challenges. IEEE Commun Mag, 2020; **58**: 46–51.
[12] Chen JH, Chen MR, Zeng GQ and Weng JS. Bdfl: a byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle. IEEE Trans Veh Technol, 2021; **70**: 8639–8652.
[13] Fang M, Cao X, Jia J and Gong N. Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), 2020, 1605–1622.
[14] Jagielski M, Oprea A, Biggio B, Liu C, Nita-Rotaru C and Li B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, 19–35.
[15] Bagdasaryan E, Veit A, Hua Y, Estrin D and Shmatikov V. How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, 2938–2948.
[16] So J, Güler B and Avestimehr AS. Byzantine-resilient secure federated learning. IEEE J Sel Areas Commun 2020; **39**: 2168–2181.
[17] Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A and Srivastava G. A survey on security and privacy of federated learning. Future Gener Comput Syst 2021, **115**:619–640.
[18] Yin D, Chen Y, Kannan R and Bartlett P. Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning, PMLR, 2018, 5650–5659.
[19] Shejwalkar V and Houmansadr A. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: Internet Society, 2021, 18.
[20] Baruch G, Baruch M and Goldberg Y. A little is enough: Circumventing defenses for distributed learning. Adv Neural Inf Proc Syst 2019; **32**: 8635–8645.
[21] Guerraoui R and Rouault S. The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning, PMLR, 2018, 3521–3530.
[22] Shejwalkar V, Houmansadr A, Kairouz P and Ramage D. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In: 2022 IEEE Symposium on Security and Privacy (SP), IEEE, 2022, 1354–1371.
[23] Cao X and Gong NZ. Mpaf: Model poisoning attacks to federated learning based on fake clients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, 3396–3404.

[24] Sun Z, Kairouz P, Suresh AT and McMahan HB. Can you really backdoor federated learning? [arXiv:1911.07963], 2019.
[25] Zhang J, Chen J, Wu D, Chen B and Yu S. Poisoning attack in federated learning using generative adversarial nets. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2019, 374–380.
[26] Wang H, Sreenivasan K, Rajput S, Vishwakarma H, Agarwal S, Sohn JY, Lee K and Papailiopoulos D. Attack of the tails: Yes, you really can backdoor federated learning. Adv Neural Inf Proc Syst 2020; **33**: 16070–16084.
[27] Zhang S, Yin H, Chen T, Huang Z, Nguyen QVH and Cui L. Pipattack: Poisoning federated recommender systems for manipulating item promotion. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, 1415–1423.
[28] Blanchard P, El Mhamdi EM, Guerraoui R and Stainer J. Machine learning with adversaries: Byzantine tolerant gradient descent. Adv Neural Inf Proc Syst 2017; **30**.
[29] Xie C, Koyejo O and Gupta I. Generalized byzantine-tolerant sgd. [arXiv:arXiv:1802.10116], 2018.
[30] Muñoz-González L, Biggio B, Demontis A, Paudice A, Wongrassamee V, Lupu EC and Roli F. Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM workshop on artificial intelligence and security, 2017, 27–38.
[31] Tolpegin V, Truex S, Gursoy ME and Liu L. Data poisoning attacks against federated learning systems. In: European Symposium on Research in Computer Security, Springer, 2020, 480–501.
[32] Nguyen TD, Rieger P, Miettinen M and Sadeghi AR. Poisoning attacks on federated learning-based iot intrusion detection system. In: Proc. Workshop Decentralized IoT Syst. Secur.(DISS), 2020, 1–7.
[33] Gong X, Chen Y, Huang H, Liao Y, Wang S and Wang Q. Coordinated backdoor attacks against federated learning with model-dependent triggers. IEEE Network, 2022; **36**: 84–90.
[34] Sun G, Cong Y, Dong J, Wang Q, Lyu L and Liu J. Data poisoning attacks on federated machine learning. IEEE Internet of Things J. 2021; **9**: 11365–11375.
[35] Xiao X, Tang Z, Li C, Xiao B and Li K. Sca: Sybil-based collusion attacks of iiot data poisoning in federated learning. IEEE Trans Ind Inf, 2022; **19**: 2608–2618.
[36] Nuding F and Mayer R. Data poisoning in sequential and parallel federated learning. In: Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics, 2022, 24–34.
[37] Zhou X, Xu M, Wu Y and Zheng N. Deep model poisoning attack on federated learning. Future Internet 2021; **13**: 73.
[38] Krizhevsky A and Hinton G. Learning Multiple Layers of Features from Tiny Images, 2009.
[39] Cohen G, Afshar S, Tapson J and Van Schaik A. Emnist: Extending mnist to handwritten letters. In: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, 2921–2926.
[40] Caldas S, Duddu SMK, Wu P, Li T, Konečnỳ J, McMahan HB, Smith V and Talwalkar A. Leaf: A benchmark for federated settings. [arXiv:1812.01097], 2018.
[41] Hsu TMH, Qi H and Brown M. Measuring the effects of non-identical data distribution for federated visual classification. [arXiv:1909.06335], 2019.
[42] Krizhevsky A, Sutskever I and Hinton GE. Imagenet classification with deep convolutional neural networks. Commun ACM 2017; **60**: 84–90.

Lei Shi received the M.S. and Ph.D. degrees in Computer System Architecture and Computer Application Technology from Nanjing University and Beijing Institute of Technology, China, in 1992 and 2006, respectively. He is currently a professor and doctoral supervisor at Zhengzhou University, China. His current research interests include cloud computing and big data, networking and distributed computing, service computing, artificial intelligence, and smart cities.

Zhen Chen received a B.S. degree in Information and Computing Sciences from Northwest A&F University, Yangling, China, in 2020. He is currently pursuing an M.S. degree in Cyberspace Security at Zhengzhou University, Zhengzhou, China. His current research interests include federated learning and poisoning attacks.

Yucheng Shi received a B.S. degree from Tianjin University, Tianjin, China, in 2017. He is currently pursuing a Ph.D. degree with the College of Intelligence and Computing, Tianjin University, at Tianjin, China. His research interests include computer vision, adversarial machine learning, and federated learning.

Lin Wei received an M.S. degree in software engineering from Zhengzhou University, Zhengzhou, China, in 2006. She is currently an associate professor and master's supervisor at Zhengzhou University. Her current research interests include network and distributed computing, data science and intelligent computing, and information security.

Yongcai Tao received M.S. and Ph.D. degrees in computer applications and computer system architecture from Zhengzhou University and Huazhong University of Science and Technology, China, in 2005 and 2009, respectively. He is currently a lecturer at the School of Computer and Artificial Intelligence, Zhengzhou University. His current research interests include theory and application research on science and intelligent computing, high-performance computing and cloud computing, service computing, and smart city, and network and information security.

Mengyang He received her Ph.D. degree in software engineering from Zhengzhou University, Zhengzhou, China, in 2021. She is currently an assistant research fellow at the School of Cyber Science and Engineering, Zhengzhou University & Song Shan Laboratory. Her main research interests include next-generation Internet, Internet applications, etc.

Qingxian Wang is currently a Distinguished Professor at the School of Cyber Science and Engineering, at Zhengzhou University. His research interests include information system vulnerability detection and analysis, network security protocol testing.

Yuan Zhou is currently the GM Assistant of Zhengzhou Zhengda lnformation Technology Co., Ltd. His research interests include electronic trading systems and cyber security.

Yufei Gao received his Ph.D. degree from the college of artificial intelligence in Beijing Normal University, Beijing, China, in 2020. He is currently an Assistant Professor at the School of Cyber Science and Engineering, at Zhengzhou University. His current research interests include pattern recognition, machine learning, and medical image analysis.