Denial-of-Service or Fine-Grained Control: Towards Flexible Model Poisoning Attacks on Federated Learning

Hangtao Zhang[1] , **Zeming Yao**[2] , **Leo Yu Zhang**[3] , **Shengshan Hu**[4,5,6,7,8] , **Chao Chen**[9] ,
Alan Liew[3] and **Zhetao Li**[1]

[1]Xiangtan University

[2]Swinburne University of Technology

[3]Griffith University

[4]School of Cyber Science and Engineering, Huazhong University of Science and Technology

[5]National Engineering Research Center for Big Data Technology and System

[6]Services Computing Technology and System Lab

[7]Hubei Key Laboratory of Distributed System Security

[8]Hubei Engineering Research Center on Big Data Security

[9]RMIT University

{zhanghangtao7, nick.yao.zm}@gamil.com, leo.zhang@griffith.edu.au, hushengshan@hust.edu.cn, chao.chen@rmit.edu.au, a.liew@griffith.edu.au, liztchina@hotmail.com

Abstract

Federated learning (FL) is vulnerable to poisoning attacks, where adversaries corrupt the global aggregation results and cause *denial-of-service* (DoS). Unlike recent model poisoning attacks that optimize the amplitude of malicious perturbations along certain prescribed directions to cause DoS, we propose a **F**lexible **M**odel **P**oisoning **A**ttack (FMPA) that can achieve versatile attack goals. We consider a practical threat scenario where no extra knowledge about the FL system (e.g., aggregation rules or updates on benign devices) is available to adversaries. FMPA exploits the global historical information to construct an estimator that predicts the next round of the global model as a benign reference. It then fine-tunes the reference model to obtain the desired poisoned model with low accuracy and small perturbations. Besides the goal of causing DoS, FMPA can be naturally extended to launch a fine-grained controllable attack, making it possible to precisely reduce the global accuracy. Armed with precise control, malicious FL service providers can gain advantages over their competitors without getting noticed, hence opening a new attack surface in FL other than DoS. Even for the purpose of DoS, experiments show that FMPA significantly decreases the global accuracy, outperforming six state-of-the-art attacks.

1   Introduction

Federated learning (FL) [McMahan and Ramage, 2017] has recently emerged as a new distributed learning paradigm. It enables multiple clients (e.g., mobile devices) to jointly learn a global model without revealing their privacy-sensitive data. Specifically, in FL, numerous clients can train local models on their private datasets and iteratively upload models/updates to the central server (e.g., Apple, Google). The server will aggregate their updates to obtain global model updates using an *aggregation algorithm* (AGR).

However, Due to its decentralized nature, FL is susceptible to poisoning attacks. By poisoning local data, malicious clients (i.e., clients compromised by an adversary) can drive the global model to learn the wrong knowledge or embed backdoors (known as *data poisoning attack*) [Tolpegin *et al.*, 2020; Wang *et al.*, 2020]. Or they can submit malicious models to the central server to disrupt the training process (a.k.a. *model poisoning attack*) [Zhou *et al.*, 2021; Xie *et al.*, 2020], such that the learnt model suffers from high testing error indiscriminately and eventually causes a *denial-of-service* (DoS).

These attacks can be either untargeted, with the goal of reducing the overall quality of the learnt model to cause DoS, or targeted, controlling the model to misclassify samples into an adversary's desired class (known as *backdoor attack*). **Our work focuses on the untargeted model poisoning attack (MPA), but the outcome is not merely causing DoS. It opens up a new attack surface (dubbed fine-grained control attacks) in FL that is more stealthy than DoS.**

To mitigate MPAs in FL, a growing number of defense options have been proposed [Cao *et al.*, 2021; Prakash and Avestimehr, 2020], where the server employs a robust AGR. Their main approach is to compare the clients' local updates and exclude abnormal ones before aggregation. Recent studies [Fang *et al.*, 2020; Shejwalkar and Houmansadr, 2021] show further adaptive or optimized MPAs can still bypass robust AGRs, especially when AGRs are known to the adversary. These novel MPAs have a huge impact on FL compared to traditional *data poisoning attacks*.
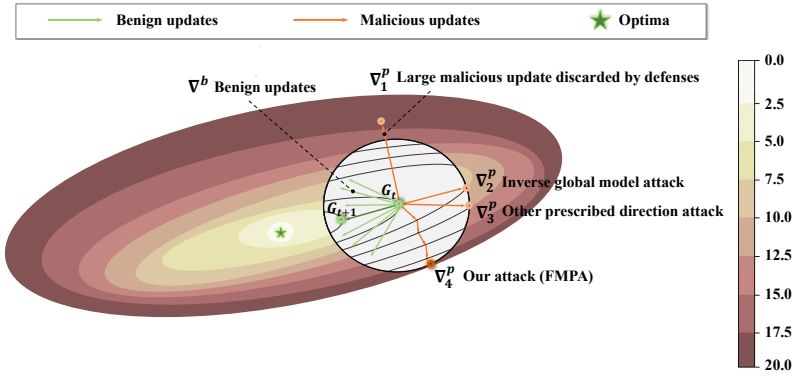
Figure 1: **A schematic of our attack:** G_t and G_{t+1} represent the global model for for rounds t and $t + 1$, respectively. The gray circle indicates the detection scope of a specific defense, thus red arrows (malicious local updates) outside the scope will be discarded (e.g., ∇_1^p). The red arrows (e.g., ∇_2^p and ∇_3^p) represent current wisdom used for designing malicious updates, where our FMPA (∇_4^p) can achieve the maximum loss (best attack effect) with the same amount of perturbation.

However, most existing advanced MPAs suffer from (part of) the following limitations:

- Assumptions about the adversary are often strong. For instance, the adversary knows the AGRs adopted by the central server [Fang *et al.*, 2020], or can assess all benign clients' local model updates (or their compele statistics) [Baruch *et al.*, 2019]. Practically, without these assumptions, the attack would be greatly weakened.

- Advanced MPAs [Shejwalkar and Houmansadr, 2021] commonly craft poisoned updates via maximizing the distance between poisoned and benign updates towards a prescribed direction (e.g., inverse of the global model). However, the prescribed direction itself does not guarantee good attack performance, which in turn makes the advanced MPAs built upon it less optimal.

- The adversary cannot precisely control the impact of the submitted poisoned model updates. To cause DoS, the adversary needs to craft poisoned updates farthest from the global model, which makes the attack easier to detect.

To address aforementioned challenges, we introduce a novel (untargeted) **F**lexible **M**odel **P**oisoning **A**ttack (FMPA). FMPA works efficiently without additional information about the FL system, except for the global model that all clients receive during normal FL training. The high-level approach of FMPA is to find the current benign reference model by exploiting the key information from the historical global models. In contrast to optimizing for the best perturbation magnitude along a prescribed malicious direction (e.g., inverse of the global model, inverse standard deviation) suggested in existing works [Fang *et al.*, 2020; Shejwalkar and Houmansadr, 2021], FMPA fine-tunes the reference model to directly obtain a malicious model with low accuracy and small perturbations. Fig. 1 intuitively illustrates the difference between FMPA and existing attacks through visualizing the loss landscape.

Our contributions are highlighted as follows:

- We introduce a new MPA for FL risk assessment. Local poisoning models are designed by attacking vulnerable parts of neural networks and minimizing the degree of modification to evade detection of Byzantine-robust AGRs. Extensive experiments reveal that, FMPA achieves better attack impact, albeit with weaker and more realistic assumptions on the adversary.

- Historical data in FL has been recently employed for detecting anomalous updates [Zhang *et al.*, 2022]. To the best of our knowledge, our FMPA is the first MPA to focus on both the vertical and horizontal information of FL systems and exploit them to assist poisoning attacks.

- We introduce a new concept of model poisoning to the community, which is not to merely target for degradation of the global model's performance indiscriminately (a.k.a. DoS), but to conceal malicious clients under a precise control of the extent of the attack. This poses an even greater threat to commercialized FL applications since an adversary can inject this subtle attack into peer FL systems to gain an advantage in competing FL services without getting noticed.

2 Background and Related Work

2.1 Federated Learning

Assume there is a general FL system, comprising of N clients and a central server. Each client $i \in [N]$ has a local training data set $\mathcal{D}_i$. We define the vector $g \in \mathbb{R}^d$ as the parameters of the co-learned model, while $j(g; c)$ denotes the loss function (e.g., *cross-entropy*) of g on input-output pairs c. Formally, the optimization objective of client i is $J_i(g) = \mathbb{E}_{c \sim \mathcal{D}_i}[j(g; c)]$. FL aims to find a model $g^\star$ that minimizes the sum of losses among clients:

$$g^\star = \min_{g \in \mathbb{R}^d} \left[J(g) := \sum_{i=1}^{N} \beta_i^t J_i(g) \right], \quad (1)$$

where client i is weighted by $\beta_i^t > 0$ in the t-th iteration.

Specifically, FL executes the following three steps:

- **Step I:** In the t-th iteration, the global model g^t is synchronized by the central server to all clients or a subset of them.

- **Step II:** After receiving g^t, the i-th client trains a new local model w_i^t over $\mathcal{D}_i$ by solving the optimization problem $\arg\min_{w_i^t} J_i(w_i^t)$ and then send its local model update $\nabla_i^t = w_i^t - g^t$ to the server.

- **Step III:** The central server aggregates all the local model updates based on an aggregation algorithm $F_{\text{AGR}}(\cdot)$ (e.g., FedAvg [McMahan *et al.*, 2017]) to obtain global update for this round as $\nabla^t = F_{\text{AGR}}(\nabla_{i \in [N]}^t)$.

Finally, the server updates the global model, i.e., $g^{t+1} = g^t - \eta \cdot \nabla^t$, where η is the global learning rate.

2.2 Poisoning Attacks on FL

Poisoning attacks are divided into two categories, namely *model poisoning* and *data poisoning* attack. This paper focuses on the more dangerous MPAs.

Model Poisoning Attack (MPA)

MPA is more dangerous as the adversary can directly manipulate local model updates. Previous MPAs include sign flipping attacks [Li *et al.*, 2019], Gaussian attacks [Xie *et al.*, 2018], etc. But these attacks cause a significant difference between poisoned and benign updates, which makes it easier to detect. As discussed below, state-of-the-art (SOTA) MPAs are either adaptive-based or optimization-based.

`LIE` [Baruch *et al.*, 2019] computes the mean μ and standard deviation σ of all updates, and finds the fixed coefficient z based on the total number of clients. The final generated malicious update is a weighted sum: $\widetilde{\nabla} = \mu + z\sigma$. But `LIE` unrealistically assumes complete knowledge of all clients' updates. Under the assumption that the exact defenses are known to adversaries, [Fang *et al.*, 2020] tailored attacks for different defenses by formulating it as an optimization problem of finding the optimized poisoned updates against known defenses. Derived from this, [Shejwalkar and Houmansadr, 2021] proposes `Min-Max` and `Min-Sum`, which minimize the maximum and the sum of distances between poisoned and all the benign updates, respectively. As reported in [Shejwalkar and Houmansadr, 2021], choosing an appropriate perturbation direction for the malicious update $\widetilde{\nabla}$ is the key to an effective MPA. Different heuristic directions, e.g., inverse standard deviation or inverse sign, are used by them, while our work directly investigates the optimal $\widetilde{\nabla}$. [Xie *et al.*, 2020] devised a MPA known as inner production manipulation (`IPM`). `MPAF` [Cao and Gong, 2023] launches an attack based on fake clients injected into the FL system. But both `IPM` and `MPAF` are limited in breaking a few simple defenses.

2.3 Existing Byzantine-Robust AGRs

We divide the current defenses into three categories, namely *statistics-based*, *distance-based* and *performance-based*, depending on the principles that the server relies to detect or evade suspicious models. *Distance-based* defenses focus on comparing the distance between local updates to find outliers. *Statistics-based* defenses utilize statistical features to remove statistical outliers. *Performance-based* defenses depend on a validation dataset to evaluate the quality of uploaded models. But some recent *performance-based* defenses are screened by evaluating the performance of local models with the need for

the central server to use a clean dataset [Wan *et al.*, 2022; Cao *et al.*, 2021]. This violates the privacy principle of FL, and similar to other Byzantine-Robust AGRs like [Karimireddy *et al.*, 2021; Shejwalkar and Houmansadr, 2021], we do not evaluate these defenses in detail.

3 Flexible Model Poisoning Attack (FMPA)

In this section, before presenting our `FMPA`, the threat model is discussed.

Adversary's Capabilities. Suppose an adversary controls m of total N clients (i.e., malicious clients). We assume that the malicious clients in all adversarial settings are below 50% (i.e., $(m/N) < 0.5$), otherwise the adversary will easily manipulate FL. The adversary can obtain the global model broadcasted each round and manipulate the local updates on malicious clients.

Adversary's Knowledge. We assume a weaker adversary who neither knows the exact AGR used by the server nor any information from other $(N - m)$ benign clients (e.g., models/gradients or local data). The adversary only has access to the data of the malicious clients as in normal FL training.

Adversary's Goals. Similar to existing MPAs, some attackers want to indiscriminately downgrade the accuracy of the global model at its testing time. Different from existing studies, some attackers may want to precisely control the effect of poisoning while maintaining concealment so as to gain advantages in establishing a competing FL service.

Next, we will introduce the method of designing concealed and effective malicious models in FL. We start with an analysis to show how to build the general model poisoning framework for deep classification models and then adapt it to flexibly poison FL systems to meet the above goals.

3.1 Model Poisoning Problem Formulation

Consider a classification task with L classes and a DNN model Θ. For a clean input-output pair (x_i, y_i), we denote the output probability for the l-th category in a neuron network as $Z_l(x_i, \Theta)$, where $x_i \in \mathcal{X}$ and its ground-truth label $y_i \in \mathcal{Y}$. Predictions for a test sample x are usually measured using the *softmax function*:

$$P_l(\Theta) = Z_l(x, \Theta) = \text{softmax}(v(x)), \qquad (2)$$

where $P = [P_l]_{l=1}^L$ represents the confidence score vector of each category. Here, $P_l \geq 0$, $\sum_{l \in [L]} P_l = 1$ and $v(x)$ is the logit vector of the penultimate layer of the neural network. Finally, the predicted label of the test sample (x, y) is $\hat{y} = \arg\max_{l \in [1,L]} P_l$. Hereinafter, we follow the MPA literature to focus on image classification tasks for easy presentation.

The ultimate goal of any untargeted MPA is to add adversarial perturbations to network parameters to make the inference results deviate from the true labels of the images. This is similar to the case of adversarial attack [Szegedy *et al.*, 2014; Zhang and Wu, 2020], with the difference that perturbations are added to the input images. Inspired by this observation, we aim to design an appropriate adversarial objective function to account for creating malicious models by gradient-based optimization. Generally, any untargeted attack can be

defined as the following constrained optimization problem:

$$\min \quad \mathcal{H}\left(\Theta, \Theta + \delta_p\right),$$
$$\text{s.t.} \quad y \neq \hat{y} = \arg\max_l Z_l(x, \Theta + \delta_p), \quad (3)$$

where $\mathcal{H}$ is a measure of the distance (e.g., ℓ_0, ℓ_2, or ℓ_∞ distance) between the benign and malicious models, δ_p is a perturbation crafted by the adversary to poison the model Θ.

To address the difficulty of directly solving the above optimization problem (*non-convex constraints* and *non-linear*), we adopt the heuristic from [Carlini and Wagner, 2017] to express it in a form that is suitable for gradient-based optimization. Say $\Theta' = \Theta + \delta_p$, we define a new loss function $\mathcal{L}(\Theta') = \mathcal{L}(\Theta + \delta_p)$ such that the constraint $y \neq \hat{y} = \arg\max_l Z_l(x, \Theta')$ is satisfied if and only if $\mathcal{L}(\Theta') \leqslant 0$. We use a variant *hinge loss* to characterize this requirement, i.e.,

$$\mathcal{L}\left(\Theta'\right) := \max\left(0, \ P_y(\Theta') - \max_{j \neq y} P_j(\Theta')\right), \quad (4)$$

where penalties are incurred when the entry of the ground-truth label is the largest among all labels.

Use the typical ℓ_2-norm to measure the similarity between benign and poisoned models, the untargeted MPA problem can be represented by the following equation:

$$\min \lambda \cdot \|\delta_p\|_2 + \mathcal{L}(\Theta + \delta_p), \quad (5)$$

where λ is a scalar coefficient to control the relative importance of perturbation. The perturbation δ_p (and the malicious model Θ') can be now solved by gradient-based optimization through backpropagation.

The above formulation of the general untargeted MPA reveals the key fact that, regardless of centralized or FL, the desired model perturbation δ_p (and thus a poisoned model) can be directly optimized. This is how the proposed FMPA differentiates from the existing works [Shejwalkar and Houmansadr, 2021; Baruch *et al.*, 2019; Fang *et al.*, 2020], where the standard deviation noise or the inverse of the global model (refer to Fig. 1 for visualization) is used while optimization is enforced to the magnitude of the prescribed direction perturbation to bypass defenses.

3.2 Optimization-Based Malicious Model Update

Instead of poisoning a fixed model Θ as shown above, poisoning of FL is a sequential process along its entire training. To apply the new formulation to FL, the adversary first needs a **reference model** as a starting point to craft a poisoned update. The basic principle to be followed is that an ideal reference model should approximate the aggregation result $F_{\text{AGR}}(w_{\{i \in [N]\}})$ in the absence of attacks. Next, we illustrate how an attacker can obtain an appropriate reference model.

Different Reference Models. We discuss three reference models, namely *historical reference model* (HRM), *alternative reference model* (ARM) and *predictive reference model* (PRM). HRM is a naive approach that takes the historical global model g^t of round t as the reference. ARM is usually adopted in existing MPAs [Fang *et al.*, 2020; Shejwalkar and Houmansadr, 2021], and it actually simulates the average aggregation of all m local updated models

$w_{\{i \in [m]\}}$ available to the attacker, i.e., $\frac{1}{m}\sum_{i \in [m]} w_i$. It is clear that the quality of ARM depends on the number m of compromised clients and their local datasets. PRM provides a better way of computing the reference model and does not depend on any other conditions. It predicts g^{t+1} from global historical information as discussed below.

We use exponential smoothing [Brown and Meyer, 1961], a well-known lightweight forecasting algorithm for time series, to build a PRM. Assume that g^t is global model in the t-th round, $s_t^{(1)}$ and $s_t^{(2)}$ are the first and second order exponential smoothing values for round t, respectively. According to *Taylor series* and *Exponential smoothing*, we calculates the estimator $\widehat{g}^{t+1}$ as the reference model:

$$\widehat{g}^{t+1} = \mathbb{P}(s_t^{(1)}, s_t^{(2)}) = \frac{2-\alpha}{1-\alpha}s_t^{(1)} - \frac{1}{1-\alpha}s_t^{(2)}, \quad (6)$$

where $\begin{cases} s_t^{(1)} = \alpha g^t + (1-\alpha)s_{t-1}^{(1)} \\ s_t^{(2)} = \alpha s_t^{(1)} + (1-\alpha)s_{t-1}^{(2)}, \end{cases}$ and $\alpha \in (0,1)$ adjusts the portion of the latest global model. We take the average of the first several true values to initialize $s_0^{(1)}$ and $s_0^{(2)}$, and recursively computes $s_t^{(1)}$ and $s_t^{(2)}$. Then we can easily update $\widehat{g}^{t+1}$ using only $s_t^{(1)}$ and $s_t^{(2)}$.

It is worth noting that unlike the literature, prediction of g^{t+1} is used for detecting malicious clients [Zhang *et al.*, 2022], we are the first to use it to aid in designing MPA. As will be validated in the experiments, PRM achieved the most stable poison effect among the three reference models.

Indiscriminate Untargeted Attack (`I-FMPA`)

In this scenario, for each training round of FL, malicious clients upload crafted poisoned updates to affect the to-be-aggregated central model. Note that we keep all poisoned updates the same to maximize the attack effect. Taking $\widehat{g}^{t+1}$ as the reference model, malicious model can be crafted by solving Eq. (5) using *stochastic gradient descent*. Algorithm 1 gives a complete description of `I-FMPA`.

To cause DoS, the best attack performance that untargeted MPAs can achieve is random guessing. To maximize attack impact, for a classification model with L categories, we set the accuracy threshold τ in Algorithm 1 to be $\left(\frac{1}{L}\right)$. First, the adversary divides the available data $\mathbb{D}$ on compromised clients into a training set $\mathbb{D}^{\text{train}}$ and a validation set $\mathbb{D}^{\text{val}}$. In lines 5-6, the adversary initializes the malicious model by the reference model $\widehat{g}^{t+1}$. In lines 8-9, the adversary finds a **certified radius** $\mathcal{R}$, which is subsequently used to tailor malicious updates to a suitable size for circumventing defenses. In lines 13-14, the adversary trains a fresh malicious model Θ' with the optimization objective of Eq. (5). The desired accuracy threshold τ is ensured through validating on $\mathbb{D}^{\text{val}}$ in line 15. In line 21, the adversary conceals itself by projecting the update on a ball of radius δ ($\|\delta\|_2 < \mathcal{R}$) around $\overline{\nabla}^t$.

As mentioned in Section 2.3, we focus on two main categories of defenses, i.e., *statistics-based* and *distance-based*. Since models close to each other may well infer the same label for the same data point, more similar models imply better robustness (against poisoning attack) [Panda *et al.*, 2022]. In these categories of defenses, outliers of uploaded model updates (which we describe with the **certified radius** $\mathcal{R}$ in the

Algorithm 1 Generating Malicious Model (Update)

Input: m malicious clients, global model $g^t \in \mathbb{R}^d$ broadcasted at round t, accuracy threshold τ, malicious clients data $\mathbb{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_m\}$ available to the adversary, the first and second order exponential smoothing $s_t^{(1)}, s_t^{(2)}$.
Function: $\mathcal{A}(\cdot)$: optimizer for parameter updates
$\mathbb{P}(\cdot, \cdot)$: Predict the next-round model ($g^t \to \widehat{g}^{t+1}$) as Eq. (6)
Output: malicious model Θ' and model update $\widetilde{\nabla}^t$

1: **if** $t = 0$ **then**
2:      $s_0^{(1)} \leftarrow g^0, s_0^{(2)} \leftarrow g^0$ ▷ initialize values
3: **else**
4:      Split $\mathbb{D}$ into train set $\mathbb{D}^{\text{train}}$ and validation set $\mathbb{D}^{\text{val}}$
5:      Set $\widehat{g}^{t+1} \leftarrow \mathbb{P}(s_t^{(1)}, s_t^{(2)})$ ▷ construct estimator
6:      Initialize malicious model $\Theta' \leftarrow \widehat{g}^{t+1}$, certified radius $R \leftarrow 0$, average update $\overline{\nabla}^t \leftarrow \widehat{g}^{t+1} - g^t$
7:      **for** each malicious client $i = 1$ to m **do**
8:          Set $\nabla_i^t \leftarrow \text{NormalLocalTraining}(g^t, \mathcal{D}_i)$
9:          $R = \text{Max}\left(\left\|\nabla_i^t - \overline{\nabla}^t\right\|_2, R\right)$▷ find certified radius
10:      **end for**
11:      **for** local epoch $e = 1$ **to** E **do**
12:          **for** batch b in $\mathbb{D}^{\text{train}}$ **do**
13:              Calculate $\mathcal{L}\left(\Theta', \mathbb{D}_b^{\text{train}}\right)$ based on Eq. (5)
14:              $\Theta' \leftarrow \Theta' - \mathcal{A}\left(\nabla\mathcal{L}(\Theta', \mathbb{D}_b^{\text{train}})\right)$
15:              **if** $\text{Accuracy}(\Theta', \mathbb{D}^{\text{val}}) \leq \tau$ **then**
16:                  **Goto** *Line* 20   ▷ early stopping
17:              **end if**
18:          **end for**
19:      **end for**
20:      Set $\widetilde{\nabla}^t \leftarrow \Theta' - g^t$ ▷ calculate malicious update
21:      $\widetilde{\nabla}^t = \prod_{\overline{\nabla}^t + \{\delta \in \mathbb{R}^d : \|\delta\|_2 < R\}}\left(\widetilde{\nabla}^t\right)$ ▷ project it on ball
22:      **Output** Θ' and $\widetilde{\nabla}^t$
23: **end if**
24: Update $s_{t+1}^{(1)}$ and $s_{t+1}^{(2)}$ according to Eq. (6)

---

proof) are likely to be considered as suspected attackers and those updates will be reprocessed (e.g., eliminated, purified, corrected) before aggregation. However, our attack ensures that poisoned models are close to the clique of benign ones. As practiced in [Shejwalkar and Houmansadr, 2021], such a fundamental attack principle achieves good performance because it can seriously confuse defenses by avoiding making itself an outlier. Further, I-FMPA searches the best-influential poisoned model in the neighborhood of the benign reference model, while the search distance is upper bounded by the maximum distance between any normally trained model (available to the attacker) and the reference model. We also provide the theoretical analysis of I-FMPA.

Definition 1 (ϱ-Corrupted Poisoning Attack). *Let $\zeta := \{\nabla_1^t, \dots, \nabla_k^t \dots, \nabla_N^t\}$ be the union of local updates $\nabla \in \mathbb{R}^d$ among all clients at round t. The set of poisoned unions $\mathcal{S}(\varrho)$ consists of $\zeta^\star := \left\{\widetilde{\nabla}_1^t, \dots, \widetilde{\nabla}_m^t, \nabla_{m+1}^t, \dots, \nabla_k^t, \dots, \nabla_N^t\right\}$, which is identical to ζ except the updates $\widetilde{\nabla}_{\{i \in [m]\}}^t$ is a ϱ-corrupted version of $\nabla_{\{i \in [m]\}}^t$. That is, for any round t, we*

have $\zeta^\star = \zeta + \epsilon$ *for certain ϵ with $\|\epsilon\|_2 \leq \varrho$.*

Definition 2 (Certified Radius). *Suppose ζ is a union, the mean of the updates among all clients is $\overline{\nabla}^t = \frac{1}{N}\sum_{i \in [1,N]} \nabla_i^t$. We call $\mathcal{R}$ a certified radius for ζ if $\sup\left\{\left\|\nabla_i^t - \overline{\nabla}^t\right\|_2 : i \in [N]\right\} = \mathcal{R}$. For simplicity, we denote it as $\zeta \models \mathcal{R}$.*

Lemma 1. *For a union and it associated certified radius $\zeta \models \mathcal{R}$, we assume that $\zeta^\star \in \mathcal{S}(\varrho)$ is a ϱ-corrupted version of ζ obtained by Algorithm 1. Algorithm 1 guarantees that, for any $\zeta^\star$, $\zeta^\star \models \mathcal{R}$ is still satisfied.*

Proof. Algorithm 1 guarantees that the distance between poisoned updates $\widetilde{\nabla}_{\{i \in [m]\}}^t$ and benign average update $\overline{\nabla}^t$ are upper bounded by the certified radius $\mathcal{R}$ by projecting the poisoned update onto a ball of radius δ ($\|\delta\|_2 < \mathcal{R}$) around $\overline{\nabla}^t$, i.e., $\left\|\mathbb{E}[\widetilde{\nabla}_{\{i \in [m]\}}^t] - \overline{\nabla}^t\right\|_2 < R$. $\square$

Theorem 1. *From lemma 1, in union $\zeta^\star$, there must be a benign client's update ∇_k^t ($k \in [m + 1, N]$) such that $\left\|\mathbb{E}[\widetilde{\nabla}_{\{i \in [m]\}}^t] - \overline{\nabla}^t\right\|_2 < \left\|\nabla_k^t - \overline{\nabla}^t\right\|_2$. If ∇_k^t is removed from the $\zeta^\star$, we have $\zeta^\star \models (R - \epsilon)$, where ϵ is a small positive constant value. That is, $\left\|\mathbb{E}[\widetilde{\nabla}_{\{i \in [m]\}}^t] - \overline{\nabla}^t\right\|_2 \leq \mathcal{R} - \epsilon < \left\|\nabla_k^t - \overline{\nabla}^t\right\|_2 \leq R$.*

Example 1. *Consider a scenario where the defense knows the number of attackers (e.g., 20%). Then, for safe aggregation, at least 20% of the elements in union $\zeta^\star$ will be reprocessed. Theorem 1 shows that there exist benign updates with the largest deviation (i.e., outliers) from the mean. They have a higher probability of being reprocessed by defenses compared to our poisoned updates.*

Fine-Grained Control Attack (`F-FMPA`)

As emphasized in the Introduction, a new feature of our `FMPA` is to enable the adversary to precisely control the performance drop of the central aggregated model. For instance, when the FL training is finished, the aggregated model g' (in the presence of fine-grained model poisoning) is inferior to model g (without model poisoning) with 10% drop in accuracy. Considering the fact that the FL central server knows neither the final accuracy of the converged model nor the clients' data, it is difficult, if not impossible, for the server to even realize that it is suffering from MPA. In particular, competing FL service providers can easily gain an advantage with such precise and subtle MPAs, hence opening a new attack surface for FL.

To perform our fine-grained control attack (precisely manipulating the MPA effects), the adversary will set τ to his desired accuracy level, say $\tau = 80\%$, and call Algorithm 1 to produce a malicious model Θ' (skip execution of lines 7-10 and 21). The remaining key issue is how to design model updates $\widetilde{\nabla}_{\text{Precise}}$ for malicious clients such that the aggregation result will be exactly Θ'. This can be achieved analytically by looking at the server-side aggregation. Specifically, for

Dataset (Model)	Defense	No attack (%)	AGR-tailored	AGR-agnostic					
			AGRT	LIE	IPM	MPAF	Min-Max	Min-Sum	I-FMPA
CIFAR-10 (Resnet20)	Krum	67.41	25.43	37.55	29.26	32.83	25.36	48.30	**56.10**
	Mkrum	84.53	19.57	35.11	23.94	27.51	45.54	48.92	**53.36**
	Bulyan	83.78	26.80	47.24	16.85	21.02	48.93	55.56	**58.26**
	Median	80.62	25.37	38.05	31.22	37.63	51.22	**56.74**	51.43
	TrMean	84.20	28.51	35.70	24.37	38.26	53.27	21.97	**57.82**
	CC	83.88	—	13.77	12.38	17.63	29.15	22.49	**33.51**
	AFA	82.57	—	16.38	7.55	18.07	41.34	37.34	**48.70**
	DNC	84.15	—	3.63	2.59	3.08	5.18	4.06	**9.77**
MNIST (FC)	Krum	88.92	20.77	10.61	17.91	5.84	3.79	**25.56**	19.36
	Mkrum	96.26	11.03	3.43	6.14	12.05	15.61	13.11	**21.08**
	Bulyan	95.38	7.44	7.02	4.30	7.81	5.35	8.07	**10.62**
	Median	93.82	1.61	1.93	3.14	**12.85**	7.65	3.02	5.97
	TrMean	96.74	1.77	2.30	1.88	9.67	9.25	8.84	**13.62**
	CC	95.71	—	1.48	0.96	1.50	1.80	2.11	**5.76**
	AFA	96.35	—	2.29	1.13	2.84	3.23	4.06	**8.64**
	DNC	95.97	—	0.21	0.39	1.21	0.35	1.26	**2.15**
EMNIST (CNN)	Krum	61.93	16.65	2.53	15.38	19.32	4.81	18.83	**36.93**
	Mkrum	80.21	22.85	14.61	12.60	26.77	52.72	37.93	**63.97**
	Bulyan	80.24	18.27	20.78	10.42	19.81	17.23	24.55	**25.79**
	Median	72.16	8.36	17.69	22.75	24.93	24.36	19.20	**36.47**
	TrMean	79.86	4.96	13.71	4.51	16.90	**26.70**	17.43	23.62
	CC	80.25	—	12.44	13.59	8.19	15.83	19.58	**25.46**
	AFA	78.92	—	9.05	6.27	24.06	35.75	32.28	**48.40**
	DNC	80.31	—	6.80	5.47	8.05	15.69	11.62	**19.54**

Table 1: Comparison of the *attack impact* ϕ between SOTA MPAs and I-FMPA.

FedAvg with and without MPA, we conclude:

$$g^{t+1} = g^t - \eta(m \cdot \nabla + \sum_{i=m+1}^{N} \nabla^i)/N,$$

$$\Theta' = g^t - \eta(m \cdot \widetilde{\nabla}_{\text{Precise}} + \sum_{i=m+1}^{N} \nabla^i)/N, \quad (7)$$

where ∇ is the averaged model update of all the malicious clients without launching MPA. In view of this fact, it is easy to see the malicious model update is:

$$\widetilde{\nabla}_{\text{Precise}} = \nabla + \frac{N}{\eta m}(g^{t+1} - \Theta'). \quad (8)$$

Replacing the unknown g^{t+1} with the predicted model $\widehat{g}^{t+1}$ using Eq. (6), the goal of precisely controlling the MPA by attackers is achieved. In the case that $\rho = \frac{N}{\eta m}$ is unknown, F-FMPA search for a suitable scaling factor ρ by iteratively increasing it each round and validate the effect on $\mathbb{D}^{\text{val}}$. Experiments show that F-FMPA also generalizes well to other robust AGRs (especially those based on filtered averaging).

4 Experimental Results and Analyses

4.1 Experimental Setup

Datasets and Models. We conduct experiments on the following datasets and models, MNIST [Lecun *et al.*, 1998] with a fully connected network (FC), EMNIST [Cohen *et al.*, 2017] with a CNN, and CIFAR-10 with Resnet20/VGG-16, to evaluate FMPA under various settings.

Parameter and Attack Settings for FL. Unless otherwise specified, we assume 20% malicious clients in all adversarial setups. Given that poisoning FL with IID data is the hardest [Fang *et al.*, 2020], following existing work [Shejwalkar and Houmansadr, 2021], our evaluation mainly considers the

case where the data is IID distributed among clients (EMNIST dataset is set to be non-IID). We also investigate the effects of different non-IID degrees on FMPA.

Poisoning Attacks for Comparison. We adopt the following MPAs for comparison, i.e., AGRT [Fang *et al.*, 2020], LIE [Baruch *et al.*, 2019], IPM [Xie *et al.*, 2020], MPAF [Cao and Gong, 2023], and Min-Max & Min-Sum [Shejwalkar and Houmansadr, 2021].

Evaluated Defenses. We consider eight classical defenses, i.e., Krum & Mkrum [Blanchard *et al.*, 2017], Median [Yin *et al.*, 2018], Trmean (Trimmed-mean) [Yin *et al.*, 2018], Norm-bounding [Sun *et al.*, 2019], Bulyan [Mhamdi *et al.*, 2018], FABA [Xia *et al.*, 2019], and AFA [Muñoz-González *et al.*, 2019], as well as two newly proposed defenses, i.e., CC (Centered Clip) [Karimireddy *et al.*, 2021] and DNC [Shejwalkar and Houmansadr, 2021].

Measurement Metrics. Let $\text{acc}_{\text{benign}}$ represent the accuracy of the optimal global model without attacks and acc_{drop} represent the accuracy degradation caused by attacks. We define *attack impact* $\phi = \text{acc}_{\text{drop}}/\text{acc}_{\text{benign}} \times 100\%$ as the degree of performance degradation of the global model. Obviously, for a given attack, the larger ϕ is, the better the attack effect is.

4.2 Evaluation Results

We compare I-FPMA with the previously described MPAs and the results are given in Tab. 1. The column *No attack* reports the global accuracy without attacks, while the remaining columns show the *attack impact* ϕ under various AGRs. In particular, AGRT is AGR-tailored (AGR is known to adversaries), while the others are all AGR-agnostic (AGR is unknown to adversaries). For fairness, all attacks except LIE are performed without knowledge of updates from benign clients (LIE uses statistics from all updates). From Tab. 1,
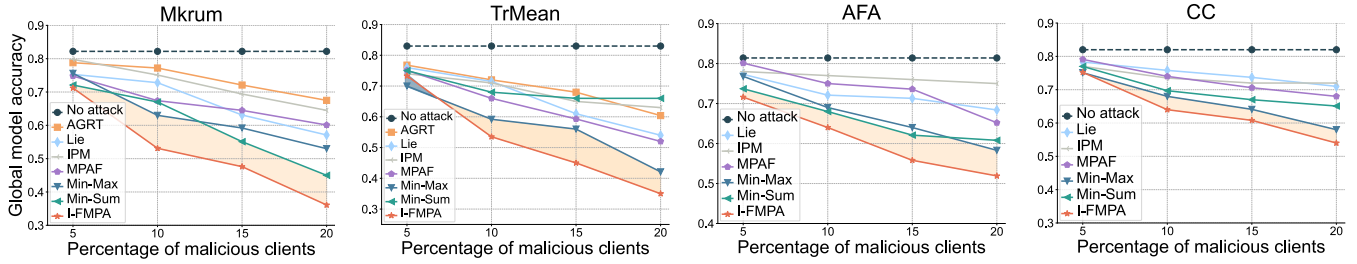
Figure 2: The global model accuracy against increasing percentage of malicious clients under different AGRs.

the *attack impact* ϕ of I-FMPA is on average $2.4\times$, $2.7\times$, $3.5\times$ and $2.8\times$ higher than AGRT, LIE, IPM, and MPAF, respectively. Under the same assumptions, I-FMPA outperforms Min-Max/Sum in most cases, showing that our attack imposes a stronger threat for causing DoS than SOTA MPAs.

Besides DoS attacks, we evaluate how precisely F-FMPA can control the attack effect. For simplicity, we define the adversary's expected accuracy drop is ξ (usually between $0 - 10\%$). Tab. 2 shows the *attack deviation* (i.e., absolute value of the difference between expected and actual accuracy drop) under different combinations of defenses and ξ. We observe that F-FMPA generalizes well to other defenses, showing that **22** out of **27** (81.48%) combinations achieve an *attack deviation* of less than 1%.

4.3 Ablation Studies

Impact of the percentage of attackers. Fig. 2 shows the impact of various MPAs as the percentage of attackers changes from 5% to 20% on CIFAR-10 with Resnet20. We note that, due to the difference in design rationale, FMPA is always superior to existing attacks for most combinations of malicious client percentages, defenses, datasets, and models.

Effects of the reference model. We compared the attack effect of three reference models (i.e., HRM, ARM and PRM). The results are shown in Fig. 3. Among them, the performance of HRM is always poor. As expected, ARM does not have an advantage when there are fewer malicious clients. The proposed PRM performs consistently well, revealing it to be a cost-effective option.

Impact of the non-IID degree. We generate the non-IID MNIST using the method in [Fang *et al.*, 2020] and evaluate the performance of MPAs under different non-IID degrees. The *attack impact* of all MPAs increases as non-IID degree increases, and I-FMPA performs better than SOTA MPAs.

Figure 3: Impacts of three different reference models on attack effect when malicious clients are armed with I-FMPA ($\alpha = 0.7$).
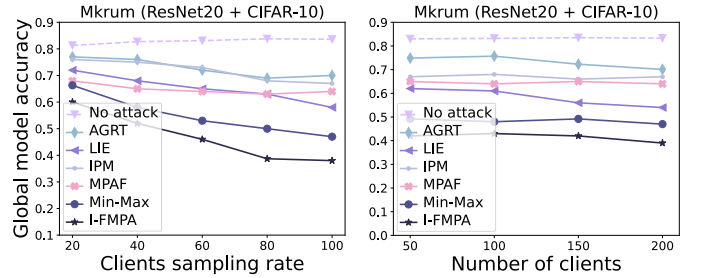
Figure 4: Testing the effect of various MPAs as we vary the clients sampling rate and total number of clients.

Different FL scenarios. Fig. 4 shows that all MPAs become less effective when FL uses a random client sampling strategy, because the adversary cannot consistently corrupt the global model. But I-FMPA still performs the best with different total client counts and sample rates.

5 Conclusion

This paper introduces a novel model poisoning framework: FMPA. To overcome the inefficiency of attacks caused by the prescribed perturbation direction in existing works, our I-FMPA utilizes historical data in FL to iteratively optimize the malicious model through neuron perturbations to launch more effective DoS attacks. It outperforms the SOTA attacks, albeit with weaker assumptions. We also propose F-FMPA, opening up a new attack surface in FL MPAs. It enables malicious service providers to surpass competitors stealthily. Currently, FMPA still utilizes data from malicious clients. We leave it as future work to explore FMPA on zero-shot attacks.

Aggregation algorithm	No attack (%)	Attack deviation (%) ↓		
		$\xi = 2.5\%$	$\xi = 5\%$	$\xi = 10\%$
FedAvg	84.70	0.17 (±0.08)	0.23 (±0.11)	0.43 (±0.18)
Norm-bounding	84.48	0.31 (±0.15)	0.38 (±0.19)	0.62 (±0.30)
Mkrum	84.53	0.35 (±0.22)	0.65 (±0.30)	0.71 (±0.34)
Median	80.62	0.50 (±0.29)	1.39 (±0.45)	2.17 (±0.72)
Bulyan	83.78	0.34 (±0.22)	0.74 (±0.34)	1.45 (±0.53)
TrMean	84.20	0.36 (±0.14)	0.79 (±0.26)	0.86 (±0.34)
CC	83.88	0.44 (±0.19)	0.87 (±0.31)	0.95 (±0.38)
AFA	82.57	0.39 (±0.20)	0.65 (±0.28)	0.88 (±0.36)
DNC	84.15	0.58 (±0.40)	1.21 (±0.51)	3.29 (±0.86)

Table 2: *Attack deviation* of F-FMPA under different AGRs and ξ.

Acknowledgments

References

[Baruch *et al.*, 2019] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, pages 8632–8642, 2019.

[Blanchard *et al.*, 2017] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

[Brown and Meyer, 1961] Robert G. Brown and Richard F. Meyer. The fundamental theorem of exponential smoothing. *Operations Research*, 1961.

[Cao and Gong, 2023] Xiaoyu Cao and Neil Zhenqiang Gong. Mpaf: Model poisoning attacks to federated learning based on fake clients. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.

[Cao *et al.*, 2021] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021*. The Internet Society, 2021.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.

[Cohen *et al.*, 2017] Gregory Cohen, Saeed Afshar, Jonathan Tapson, van Schaik, et al. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, 2017.

[Fang *et al.*, 2020] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*. USENIX Association, 2020.

[Karimireddy *et al.*, 2021] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[Lecun *et al.*, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2019] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[McMahan and Ramage, 2017] H. Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, 2017. Accessed: 2022-08-14.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, pages 1273–1282, 2017.

[Mhamdi *et al.*, 2018] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3518–3527, 2018.

[Muñoz-González *et al.*, 2019] Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *CoRR*, abs/1909.05125, 2019.

[Panda *et al.*, 2022] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

[Prakash and Avestimehr, 2020] Saurav Prakash and Amir Salman Avestimehr. Mitigating byzantine attacks in federated learning. *arXiv: Distributed, Parallel, and Cluster Computing*, 2020.

[Shejwalkar and Houmansadr, 2021] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security Symposium*, 2021.

[Sun *et al.*, 2019] Ziteng Sun, Peter Kairouz, and Ananda Theertha Suresh. Can you really backdoor federated learning? *CoRR*, abs/1911.07963, 2019.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, et al. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations ICLR*, 2014.

[Tolpegin *et al.*, 2020] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *Proceedings of the 25th European Symposium on Research in Computer Security, ESORICS 2020*, 2020.

[Wan *et al.*, 2022] Wei Wan, Shengshan Hu, jianrong Lu, Leo Yu Zhang, Hai Jin, and Yuanyuan He. Shielding federated learning: Robust aggregation with adaptive client

selection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 753–760, 2022.

[Wang *et al.*, 2020] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, You really can backdoor federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[Xia *et al.*, 2019] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. Faba: An algorithm for fast aggregation against byzantine attacks in distributed neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4824–4830, 2019.

[Xie *et al.*, 2018] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant SGD. *CoRR*, abs/1802.10116, 2018.

[Xie *et al.*, 2020] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 261–270, 2020.

[Yin *et al.*, 2018] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 5636–5645, 2018.

[Zhang and Wu, 2020] Zekun Zhang and Tianfu Wu. Learning ordered top-k adversarial attacks via adversarial distillation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3364–3373, 2020.

[Zhang *et al.*, 2022] Zaixi Zhang, Xiaoyu Cao, Jin Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022.

[Zhou *et al.*, 2021] X. Zhou, M. Xu, Y. Wu, and N. Zheng. Deep model poisoning attack on federated learning. *Future Internet*, 13(3):73, 2021.