Performance Guaranteed Poisoning Attacks in Federated Learning: A Sliding Mode Approach

Huazi Pan[1] , **Yanjun Zhang**[2] , **Leo Yu Zhang**[3] , **Scott Adams**[1] , **Abbas Kouzani**[1] , **Suiyang Khoo**[1]

[1]Deakin University

[2]University of Technology Sydney

[3]Griffith University

{panhuaz, scott.adams, abbas.kouzani, sui.khoo}@deakin.edu.au, yanjun.zhang@uts.edu.au,
leo.zhang@griffith.edu.au

Abstract

Manipulation of local training data and local updates, i.e., the poisoning attack, is the main threat arising from the collaborative nature of the federated learning (FL) paradigm. Most existing poisoning attacks aim to manipulate local data/models in a way that causes denial-of-service (DoS) issues. In this paper, we introduce a novel attack method, named Federated Learning Sliding Attack (FedSA) scheme, aiming at precisely introducing the extent of poisoning in a subtle controlled manner. It operates with a predefined objective, such as reducing global model's prediction accuracy by 10%. FedSA integrates robust nonlinear control-Sliding Mode Control (SMC) theory with model poisoning attacks. It can manipulate the updates from malicious clients to drive the global model towards a compromised state, achieving this at a controlled and inconspicuous rate. Additionally, leveraging the robust control properties of FedSA allows precise control over the convergence bounds, enabling the attacker to set the global accuracy of the poisoned model to any desired level. Experimental results demonstrate that FedSA can accurately achieve a predefined global accuracy with fewer malicious clients while maintaining a high level of stealth and adjustable learning rates.

1 Introduction

Federated learning (FL) is a distributed optimization framework in machine learning where the learning process is decentralized across multiple client devices, such as mobile phones, rather than being centralized [McMahan *et al.*, 2017; Zhang *et al.*, 2021; Zhang *et al.*, 2019; Jebreel and Domingo-Ferrer, 2023]. This approach enables users to train models on locally stored data without needing to disclose their data. In FL, the central server (e.g., a cloud server) initially sends the global model to selected clients or all participating clients. Each client then trains the model locally using its own data. The updated models are subsequently uploaded to the central server, which aggregates them using an aggregation algorithm (AGR) to generate a new global model.

However, the natural collaboration paradigm of FL exposes it to the risk of poisoning attacks [Shen *et al.*, 2022; Zhang *et al.*, 2020]. These attacks can occur through either data poisoning or model poisoning. Data poisoning [Jagielski *et al.*, 2018; Muñoz-González *et al.*, 2017] involves contaminating the local data, while model poisoning [Bagdasaryan *et al.*, 2020; Baruch *et al.*, 2019; Fang *et al.*, 2020; Mhamdi *et al.*, 2018; Xie *et al.*, 2020] refers to altering local updates. The goals of typical poisoning attacks can be divided into untargeted attack [Bhagoji *et al.*, 2019; Fang *et al.*, 2020; Mahloujifar *et al.*, 2019; Mhamdi *et al.*, 2018; Xie *et al.*, 2020] and targeted attack [Bagdasaryan *et al.*, 2020; Bhagoji *et al.*, 2019]. The former aims to corrupt the global model to produce incorrect predictions on arbitrary test samples while the latter aims to manipulate the model to misclassify specific classes or sets of samples chosen by the attacker.

In this work, we investigate a newly emerging attack: the controllable poisoning attack against FL. This type of attack aims to precisely control the degree of performance degradation. Unlike traditional poisoning attacks, which typically result in a denial-of-service (DoS), the controllable poisoning attack operates with a predefined objective, such as reducing prediction accuracy by 10%. This attack, for example, can be utilized to subtly influence competitors in the commercial market. Instead of drastically impacting the performance of a competitor's algorithm, the attacker precisely introduces the extent of poisoning in a subtle controlled manner.

This emerging attack surface remains largely unexplored, as previous studies only primarily focused on the traditional model replacement strategy, where the global model is replaced with a poisoned reference model [Zhang *et al.*, 2023]. However, this approach cannot guarantee a poisoning trajectory that converges to the attack's objectives. The attack's objective also remains fixed during the poisoning process, as it is predetermined by the reference model and cannot adapt to dynamic or evolving goals. Additionally, the attack lacks the ability to regulate its poisoning speed, increasing the risk of detection by Byzantine-robust aggregation rules (AGRs) if the poisoning occurs too rapidly.

To address these challenges, we propose Federated Learning Sliding Attack (FedSA), a method designed to enable performance-guaranteed poisoning attacks with precise control and adjustable poisoning speed, drawing on principles from Sliding Mode Control (SMC) theory [Young *et al.*,

1999]. Furthermore, the attack objective can be flexibly adjusted during the poisoning process to align with evolving goals.

To achieve precise control, we conceptualize the entire FL process as a system and design a control law to adjust malicious clients' models, steering the global model towards the poisoned reference model. Specifically, Sliding Mode Control (SMC)—an effective tool for robust controller design in complex nonlinear dynamic systems—offers low sensitivity to parameter variations and disturbances, reducing the need for exact modeling [Young *et al.*, 1999]. By applying SMC, the system state is ensured to slide along a predefined sliding surface, thereby forcing the outputs of the FL process to align with the intended attack objective. The flexibility of attack objectives is achieved by introducing an adjustable variable to control the difference between the global model and the reference model. Additionally, the attack's progression speed can be subtly regulated by adjusting the control gain, eliminating the need for continuous parameter optimization as required in traditional model replacement strategies. In a nutshell, these new features significantly enhance the attack's stealth, allowing FedSA to evade detection and bypass defenses.

Our key contributions can be concluded as follows:

- We introduce a novel controllable attack, FedSA, that achieves precision control during poisoning the FL training. The SMC control law is specifically designed to ensure that the attack reaches its desired objectives while evading various Byzantine-robust AGRs.

- The proposed scheme allows the attacker to flexibly adjust its objective, without the need to generate new poisoned reference models for different targets.

- We theoretically demonstrate the guarantee of the precise control of the model performance. We prove that FedSA converges to the predefined objectives within a finite time, with freely controllable convergence speed. We conduct extensive experiments on benchmarking datasets against a variety of representative robust AGRs. We demonstrate that our design outperforms the existing SOTA poisoning attacks consistently across all settings.

2 Background and Related Work

2.1 Federated Learning

As shown in [McMahan *et al.*, 2017], a typical FL framework consists of a central server and N clients, each client i ($i \in [1, N]$) has its local private dataset D_i divided from a data distribution D, and the datasets can be IID (i.e., independently and identically distributed) or Non-IID. Define the global model as $w_t \in \mathbb{R}^r$ and local model as $w_{\{t,i\}} \in \mathbb{R}^r$ at the t-th iteration, the objective of FL is to optimize

$$w^* = \underset{w_{\{t,i\}}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} L(w_{\{t,i\}}, D_i), \quad (1)$$

where $w^* \in \mathbb{R}^r$ is the optimal global model, r represents the total number of parameters (including weights and biases) of a neural network, and L denotes the loss function. To solve this, the central server first delivers the current global model

parameters w_t to all clients or a subset of them in the t-th iteration. Upon receiving w_t, each client trains its local model $w_{\{t,i\}}$ using w_t and its private training dataset D_i. Then the clients send the local model update $\nabla_{\{t,i\}} = w_{\{t,i\}} - w_t$ to the central server. Following a certain aggregation rule $F_{\text{AGR}}(\cdot)$, the central server aggregates the received local updates to obtain a new global update $\nabla_t = F_{\text{AGR}}(\{\nabla_{\{t,i\}} | i \in [1, N]\})$. Finally, the central server generates a new global model $w_{t+1} = w_t - \eta \nabla_t$ and sends it back to all (or a subset of) clients, where η is the global learning rate. This process of FL is repeated until the global model converges.

2.2 Poisoning Attacks on FL

Poisoning attacks in Federated Learning (FL) can be broadly categorized into two types: data poisoning and model poisoning attacks (MPAs). In data poisoning attacks [Jagielski *et al.*, 2018; Muñoz-González *et al.*, 2017], the adversary poisons training datasets on the malicious devices while MPAs [Bagdasaryan *et al.*, 2020; Baruch *et al.*, 2019; Fang *et al.*, 2020; Mhamdi *et al.*, 2018; Xie *et al.*, 2020] directly manipulate the local model gradients on malicious participants and send them to the server during the learning process. [Zhang *et al.*, 2023] presented a novel FL model poisoning attack called Flexible Model Poisoning Attack (FMPA), which aims to reduce the global model's performance with a predefined objective. FMPA uses the current round global model as a reference, and fine-turns it to obtain a desired poisoned model. It then replaces the next ground global model with the poisoned version and iteratively searches for optimal parameters to update the poisoned model, achieving a degree of control over the attack effects. However, model replacement strategy cannot ensure that the global model will converge to the attacker's objective. In addition, the attack suffers from an uncontrollable speed and requires a large proportion of malicious clients, making it easily detected by robust AGRs. Moreover, the predetermined attack objective cannot be dynamically adjusted once the attack is initiated.

2.3 Existing Byzantine-robust Defenses

Current defenses can be categorized based on the principles that the server employs for detecting-then-removing or suppressing suspicious models, falling into three types: statistics-based, distance-based, and performance-based approaches [Shen *et al.*, 2022; Gong *et al.*, 2024; Gong *et al.*, 2025]. Statistics-based defenses, such as Median [Yin *et al.*, 2021] and Trimmed Mean [Yin *et al.*, 2021], use statistical feature (e.g., mean or median) to aggregate on each dimension of input gradients individually. Distance-based defenses like Krum [Blanchard *et al.*, 2017], Mkrum [Blanchard *et al.*, 2017], and Bulyan [Mhamdi *et al.*, 2018] evaluate the distance between local updates, such as Euclidean distance and Cosine similarity, to determine statistics outliers. Performance-based defenses, such as Fang [Fang *et al.*, 2021], rely on a validation dataset to compute the performance of the uploaded models and remove divergent ones.

2.4 Sliding Mode Control

Sliding Mode Control [Khoo *et al.*, 2009; Lee *et al.*, 2013; Alqumsan *et al.*, 2019; Yin *et al.*, 2011; Khoo *et al.*, 2013;

Yu *et al.*, 2021] is a robust nonlinear control approach that uses a discontinuous control signal to alter the dynamics of a system, forcing it to slide along a specified switching manifold s_t. The objective is to design a control law u_t that ensures the system state w_t tracks the desired state $\tilde{w}$, on the sliding surface $s_t = 0$.

Consider a first order system $\dot{w}_t = u_t$ as a toy example. Throughout this paper, the dot symbol above a letter represents its derivative with respect to time t, e.g., $\dot{w}_t$ is the derivative of w_t. In general, the sliding surface is chosen as:

$$s_t = \dot{e}_t + k e_t, \tag{2}$$

where $e_t = \tilde{w} - w_t$ is the error between the current state w_t and the desired state $\tilde{w}$, at t-th iteration, and k is a hyperparameter that governs the convergence speed of e_t. If the control can ensure that $s_t = 0$, i.e., $\dot{e}_t = -k e_t$, solving this first order differential equation yields $e_t = e_0 e^{-kt}$, which converges exponentially as t increases.

Next is to show how to design u_t to ensure $\lim_{t \to T} s_t = 0$. The control law u_t can be designed as:

$$\dot{u}_t = -k u_t + \alpha \cdot \text{sign}(s_t), \tag{3}$$

where α is a positive constant selected to force the system trajectory to reach the sliding mode surface, and $\text{sign}(\cdot)$ is the sign function given below:

$$\text{sign}(s_t) = \begin{cases} +1 & \text{if } s_t > 0; \\ 0 & \text{if } s_t = 0; \\ -1 & \text{if } s_t < 0. \end{cases}$$

We define an energy function $V_t = \frac{1}{2} s_t^2$ to ensure the convergence of s_t. By applying Eqs. (2) and (3), we get the derivative of V_t as:

$$\begin{aligned} \dot{V}_t &= s_t \cdot \dot{s}_t = s_t \cdot (-\dot{u}_t - k u_t) \\ &\leq s_t(-\alpha \cdot \text{sign}(s_t)) = -\alpha|s_t| = -\sqrt{2}\alpha V_t^{1/2}. \end{aligned} \tag{4}$$

Eq. (4) satisfies the condition of finite-time stability theorem in [Khoo *et al.*, 2009] for s_t to converge to zero in a finite time. Graphically, referring to Fig. 1, for any s_0, the initial state of s_t, be negative (blue points) or positive (green points), due to the fact that V_t is a quadratic function, $\dot{V}_t \leq 0$ (i.e., $V_{t+1} < V_t$ for $V_t \neq 0$), s_t will move along the direction of the arrow in a decreasing manner and eventually converge to zero. When $s_t = 0$, the desired performance, $e_t = e_0 e^{-kt}$ is achieved, and $e_t \to 0$ exponentially fast. Since $e_t = \tilde{w} - w_t$, this also implies the desired $w_t \to \tilde{w}$ exponentially.

3 The Design of FedSA

3.1 Problem Formulation

Threat Model. The same as other MPAs, we assume that the adversary controls m out of N FL clients, with $m/N = 10\%$ as the default setting in the evaluation. The adversary[1] has access to the broadcast global model w_t and can freely manipulate the set of local models $w'_t = \{w'_{\{t,i\}} | i \in [1, m]\}$ by fully leveraging the SMC control principles in Sec. 2.4.

---
[1]We remark that the adversary is unaware of the AGR rules and this will be discussed in detail in Sec. 3.2.
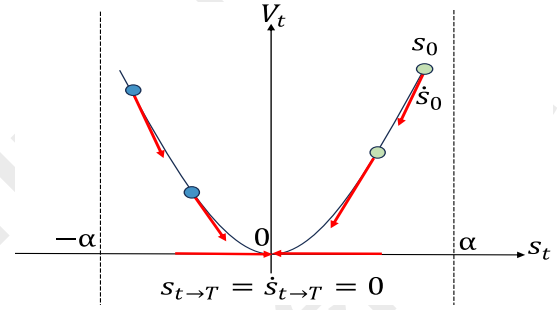

Figure 1: Convergence of s_t.

Adversary's Goal. For a FL classification task, the goal of a flexible MPA is to steer the global model $w_t = F_{\text{AGR}}\{w_{\{t,1\}}, w_{\{t,2\}}, ..., w_{\{t,m\}}, ..., w_{\{t,N\}}\}$, which would otherwise converge to the global optimum without attack, toward a subpar reference model $\tilde{w}$. This reference model can be obtained through early stopping, say using the criterion that $\tilde{w}$ exhibits a 10% reduction in accuracy.

As depicted in Fig. 2, the differences between the existing MPAs and FedSA lies in the guarantees provided by our approach. FedSA ensures that the attack objective (represented by the red star, chosen by the attacker and close to the optimum) is achieved with controlled progression speed and flexibly adjustable attack objectives. This fundamentally sets FedSA apart from SOTA works such as LIE [Baruch *et al.*, 2019], Min-Max/Min-Sum [Shejwalkar and Houmansadr, 2021] and FMPA [Zhang *et al.*, 2023], which lack theoretical convergence guarantees and face significant challenges—if not impossible—in controlling poisoning speed and adapting to evolving goals.

3.2 FedSA Details
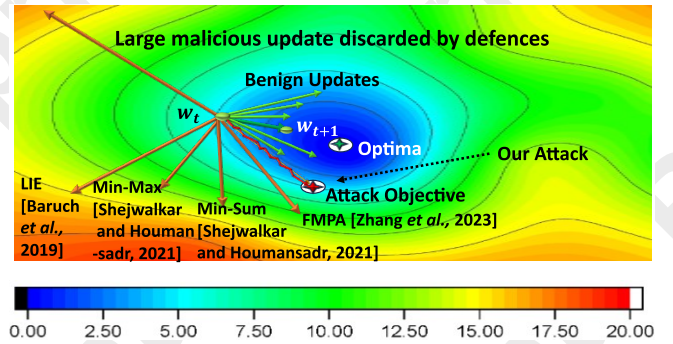
Figure 2: The comparison of existing attacks and our attack The attack effect is illustrated via loss contours—blue area indicates low loss and red area indicates high loss.

Similar to the toy example in Sec. 2.4, we treat the overall FL global model

$$w_t = F_{\text{AGR}}\{w_{\{t,1\}}, w_{\{t,2\}}, ..., w_{\{t,m\}}, ..., w_{\{t,N\}}\} \tag{5}$$

as a nonlinear system, and in particular, the malicious local models are chosen as

$$\dot{w}'_{\{t,i\}} = u_t. \tag{6}$$

The key difference from the toy example is that the goal of FedSA is to design the control law u_t to ensure that w_t—rather than $w'_{\{t,i\}}$, as in the toy example—slides along the surface $s_t = 0$ to achieve:

$$e_t(\tilde{w}, w_t) = -C/k \qquad (7)$$

exponentially fast, where $C \in \mathbb{R}$ is a constant to adjust the convergence status of e_t, and $k \in \mathbb{R}$ ($k \neq 0$) is a parameter to adjust the convergence speed of e_t. To achieve the adversary's goal, we design the error function as

$$e_t = \tilde{w} - w_t, \qquad (8)$$

To realize this new error, we design the sliding surface as

$$s_t = \int (\dot{e}_t + ke_t + C)\mathrm{dt} + C_1. \qquad (9)$$

where $C_1 \in \mathbb{R}$ is the initial value of the sliding surface s_t, which can be any constant.

After selecting the sliding surface s_t, the control law u_t is designed based on the FL system, the dynamic model in Eq. (6), the error function in Eq. (8) and sliding surface s_t in Eq. (9), as follows:

$$u_t = \left[\frac{dF_{\mathrm{AGR}}(\cdot)}{dw'_{\{t,i\}}}\right]^{-1}[ke_t + \eta\mathrm{sign}(s_t) - \Theta_t + C], \quad (10)$$

where $\eta > 0$ is a positive constant selected to force the system trajectory to reach the sliding mode surface. Here, $dF_{\mathrm{AGR}}(\cdot)/dw'_{\{t,i\}}$ is the derivative of F_{AGR} with respective to $w'_{\{t,i\}}$ and $\Theta_t = \sum_{i=1,i\neq m}^{N} \frac{dF_{\mathrm{AGR}}(\cdot)}{dw_{\{t,i\}}} \cdot \dot{w}_{\{t,i\}}$. It is noted if F_{AGR} is not differentiable or unknown to the adversary, $dF_{\mathrm{AGR}}(\cdot)/dw'_{\{t,i\}}$ can be approximated by finite differences as:

$$\frac{dF_{\mathrm{AGR}}(\cdot)}{dw'_{\{t,i\}}} \approx \lim_{\Delta w'_{\{t,i\}}\to 0} \frac{F_{\mathrm{AGR}}(t) - F_{\mathrm{AGR}}(t - \Delta t)}{\Delta w'_{\{t,i\}}}.$$

This explains the threat model outlined in Sec. 3.1 and we use this approximation by default in the experimental analysis in Sec. 4.

FedSA's Algorithm. The algorithm of the FedSA is shown in Algorithm 1. Initially, the malicious clients receive the global model w_t from the server, and then calculate the error e_t between global model and target model in Eq. (8) (line 6). By choosing the sliding surface in Eq. (9) (line 7), s_t will converge to 0 in a finite time, hence e_t will converge to $-C/k$ in an exponential rate. Then, by substituting Eq. (8) and Eq. (9) (lines 6 and 7) into the control input u_t in Eq. (10), the malicious model is obtained (line 10). After that, the malicious model is uploaded to the server to be aggregated with updates from other clients by different AGRs (line 13).

Fig. 3 illustrates the process of our FedSA. The derivative of s_t is $\dot{s}_t = \dot{e}_t + ke_t + C$, which is written at the top of the Fig. 3. By applying SMC, e_t can be forced under control law u_t from the initial point C_1 to reach the sliding surface and slide along the designed sliding surface $s_t = \dot{s}_t = 0$, which are represented by two blue dash lines for $C = 0$ and $C \neq 0$, respectively. Theorem 1 guarantees that, in the presence of poisoned local models $w'_t = \{w'_{\{t,i\}}|i \in [1,m]\}$, the global model w_t will achieve the desired target accuracy $\tilde{w} = w_t - C/k$ chosen by the attacker by manipulating C.

Algorithm 1 Malicious Model Update

Input: Global model w_t, desired poisoning model $\tilde{w}$.
Function: Control input u_t in Eq. (10), sliding surface s_t in Eq. (9) and error function e_t in Eq. (8).
Output: Malicious model update vector w'_t.
1: **if** t=0 **then**
2:     $w_t \leftarrow w_0$
3:     Initialize $\tilde{w}$, C_1
4: **else**
5:     **for** malicious client $i = 1$ to m **do**
6:        Calculate e_t of $\tilde{w}$ and w_t in Eq. (8)
7:        Calculate s_t in Eq. (9)
8:     **end for**
9:     Calculate u_t from Eq. (10)   ▷ {Control Law}
10:     Calculate w'_t from Eq. (6)
11:     **Output** w'_t
12: **end if**
13: Upload malicious models w'_t to the FL server
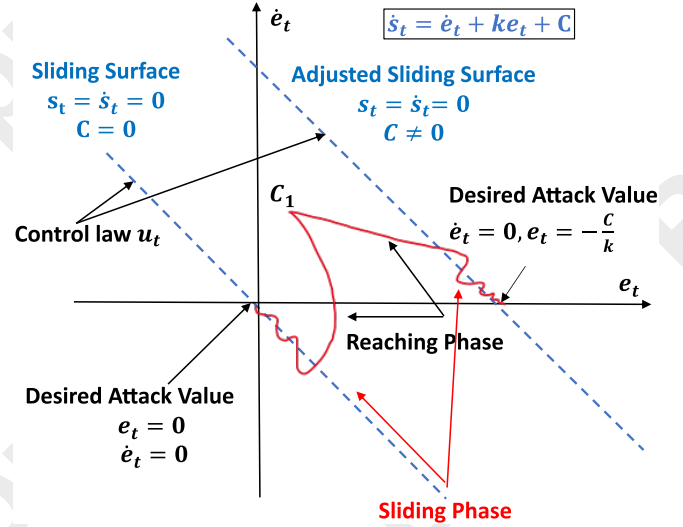

Figure 3: The mechanism of FedSA.

Theorem 1. *Consider a FL system governed by the dynamics described in Eq. (6) with the error function given by Eq. (8) and a sliding surface defined by Eq. (9). Let the control law u_t be designed as in Eq. (10) with constants $k > 0$, $\eta > 0$, and $C \in \mathbb{R}$, and the derivative of the aggregation function F_{AGR} with respect to the malicious model $w'_{\{t,i\}}$ is continuous. Then the SMC control law ensures that: 1): The sliding surface s_t reaches zero in a finite time and remains zero thereafter. 2): The error $e_t = \tilde{w} - w_t$ converges to $-C/k$ exponentially fast.*

Due to space limit, the proof of Theorem 1 is delayed to Supp.-1.1. Below, we highlight key remarks on the new features of FedSA.

Remark 1: Attack Speed. The adversary can control the convergence speed of e_t by designing the appropriate k. On the sliding mode surface (i.e., $s_t = \dot{s}_t = 0$), solving the ordinary differential equation $\dot{e}_t = -ke_t - C$, yields $e_t =$

$1/k \cdot e_0^{-kt} - C/k$. It is evidence that, k governs the exponential convergence rate of e_t: the larger the value of k, the faster e_t converges exponentially. Hence, FedSA allows the adversary to freely control the attack's convergence speed.

Remark 2: Adjustable Objectives. At any stage of FedSA, the adversary can flexibly adjust its objective by selecting a different value of C in e_t, as shown in Fig. 3 and Eq. (8). At equilibrium on the sliding surface, $\dot{s}_t = 0$, $\dot{e}_t = 0$. Using $\dot{s}_t = \dot{e}_t + ke_t + C$ again, we have the desired $e_t = -C/k$ or $w_t = \tilde{w} + C/k$. This phenomena is denoted by the red curves' ending points on the sliding surface for $C = 0$ and $C \neq 0$ in Fig. 3 where the adversary can achieve the desired attack objective.

4 Performance Evaluation

4.1 Experiment Setup

Datasets and Models

For the experiments of FedSA, the following models and datasets are considered: AlexNet [Yang *et al.*, 2017] is used for CIFAR10; FC (Fully Connected Network) for MNIST; and ResNet50 for Tiny ImageNet. The details of each dataset are demonstrated in Supp.-2.1. We implement our FedSA under both IID and Non-IID data split strategy. Taking CIFAR10 dataset as an example, for IID scenarios, each client receives 100 training images from each of the 10 classes, culminating in a total of 1000 training images per client. In addition, we also use Dirichlet distribution under different degrees of Non-IID scenarios including 0.1, 0.3, 0.5, 0.7 and 0.9. A smaller degree value leads to more skewed distributions, where clients predominantly receive data from a few classes. A larger degree value results in a more balanced distribution. For example, if Non-IID level is at 0.1, each client will receive 1000 images from only one or two classes.

FL System Settings

For CIFAR10 dataset with AlexNet, the global learning rate is set as 0.02, the global batch size is set as 128 and the global epochs is set as 100. For the local training process, the batch size is set as 10, and the epochs is set as 5. For MNIST dataset with fully connected network, the global learning rate is set as 0.01, the global batch size is set as 128 and the global epochs is set as 100. For the local training process, the batch size is set as 5, and the epochs is set as 3. As for Tiny ImageNet with ResNet50, the global training rate is set as 0.001, the global batch size is set as 128, the global epochs is set as 20. In the local training process, the batch size is set as 10, and the epochs is set as 3.

Attack Settings

We consider 50 clients participated and 10% malicious clients as default setting, and study the impact of the various proportion of malicious clients in Section 4.3. 10% of malicious proportion is considered as it is the benchmark setting in the literature of positioning attacks against FL [Zhang *et al.*, 2023; Shejwalkar and Houmansadr, 2021; Baruch *et al.*, 2019]. This setting is noticeably realistic in real-world scenarios, such as Sybil attacks and botnets, where a malicious attacker can compromise or simulate large numbers of users. As for setting various attack objective, we set 60%, 55%, 50%,

Dataset (Model)	AGRs	No Attack(%)	Test Acc. (Difference to the Targeted Acc. δ (%))				
			LIE	Min-Max	Min-Sum	FMPA	FedSA
			Target Acc 60%				
CIFAR10 (AlexNet)	FedAvg	66.59	40.3 (-32.83)	43.73 (-27.12)	31.57 (-47.38)	53.19 (-11.35)	59.76 (**-0.40**)
	Median	64.22	35.04 (-41.60)	42.98 (-28.37)	46.31 (-22.82)	49.03 (-18.28)	57.22 (**-4.63**)
	Trmean	66.32	41.57 (-30.72)	43.38 (-27.70)	43.86 (-26.9)	55.16 (-8.07)	62.18 (**3.63**)
	NB	66.75	37.32 (-37.80)	45.64 (-23.93)	46.07 (-23.22)	51.45 (-14.25)	60.06 (**0.10**)
	Bulyan	66.09	34.29 (-42.85)	44.32 (-26.13)	41.39 (-31.02)	62.59 (4.32)	61.93 (**3.22**)
	Mkrum	66.89	40.97 (-31.72)	33.04 (-24.31)	31.69 (-47.18)	50.94 (-15.10)	60.59 (**0.98**)
	Fltrust	66.70	52.35 (-12.75)	50.79 (-19.79)	52.56 (-12.4)	58.16 (-3.07)	60.62 (**1.03**)
	CC	66.84	51.63 (-13.95)	47.36 (-21.07)	51.80 (-13.67)	42.36 (-29.40)	62.05 (**3.42**)
	DNC	66.70	64.70 (7.83)	63.90 (6.50)	54.95 (-8.42)	50.52 (-15.80)	61.35 (**2.25**)
			Target Acc 55%				
	FedAvg	66.59	40.3 (-26.73)	43.73 (-20.49)	31.57 (-42.60)	58.52 (6.40)	56.37 (**2.49**)
	Median	64.22	35.04 (-36.29)	42.98 (-21.85)	46.31 (-15.80)	51.22 (-6.87)	51.47 (**-6.42**)
	Trmean	66.32	41.57 (-24.42)	43.38 (-21.13)	43.86 (-20.25)	53.12 (-21.60)	52.50 (**-4.55**)
	NB	66.75	37.32 (-32.15)	45.64 (-17.02)	46.07 (-16.24)	58.29 (5.98)	56.42 (**2.58**)
	Bulyan	66.09	34.29 (-37.65)	44.32 (-19.42)	41.39 (-24.75)	48.79 (-11.29)	54.31 (**-2.35**)
	Mkrum	66.89	40.97 (-25.51)	33.04 (-39.93)	31.69 (-42.38)	51.07 (-7.15)	54.92 (**-0.15**)
	Fltrust	66.70	52.35 (-4.82)	50.79 (-7.65)	52.56 (-4.44)	34.50 (-37.27)	55.16 (**0.29**)
	CC	66.84	51.63 (-6.13)	47.36 (-13.89)	51.80 (-5.82)	59.37 (7.95)	55.71 (**1.29**)
	DNC	66.70	64.70 (17.64)	63.90 (16.18)	54.95 (-0.09)	51.42 (-6.51)	53.14 (**-3.38**)
			Target Acc 50%				
	FedAvg	66.59	40.3 (-19.40)	43.73 (-12.54)	31.57 (-36.86)	57.70 (15.40)	50.87 (**1.74**)
	Median	64.22	35.04 (-29.92)	42.98 (-14.04)	46.31 (-7.38)	48.65 (-2.70)	50.99 (**1.98**)
	Trmean	66.32	41.57 (-16.86)	43.38 (-13.24)	43.86 (-12.28)	53.17 (6.34)	50.71 (**1.42**)
	NB	66.75	37.32 (-25.36)	45.64 (-8.72)	46.07 (-7.86)	56.02 (12.04)	50.26 (**0.52**)
	Bulyan	66.09	34.29 (-31.42)	44.32 (-33.92)	41.39 (-36.62)	59.03 (18.06)	49.83 (**-0.34**)
	Mkrum	66.89	40.97 (-18.06)	33.04 (-24.31)	31.69 (-47.18)	62.52 (25.04)	50.53 (**1.06**)
	Fltrust	66.70	52.35 (4.70)	50.79 (1.58)	52.56 (5.12)	52.26 (4.52)	51.87 (**3.74**)
	CC	66.84	51.63 (3.26)	47.36 (-5.28)	51.80 (3.60)	43.99 (12.02)	50.42 (**0.84**)
	DNC	66.70	64.70 (29.40)	63.90 (27.80)	54.95 (9.90)	52.44 (4.88)	51.46 (**2.92**)
			Target Acc 90%				
MNIST (FC)	FedAvg	97.97	96.73 (7.48)	93.18 (3.53)	92.84 (3.16)	95.22 (5.80)	90.45 (**0.50**)
	Median	97.78	97.73 (8.59)	92.76 (3.07)	92.84 (3.16)	44.21 (-50.88)	88.56 (**-1.60**)
	Trmean	97.98	96.02 (6.69)	92.88 (3.40)	92.43 (2.70)	97.29 (8.10)	90.24 (**0.27**)
	NB	97.97	92.82 (3.13)	92.78 (3.09)	93.02 (3.36)	59.8 (-33.56)	91.49 (**1.66**)
	Bulyan	97.94	92.35 (2.61)	92.90 (3.22)	92.29 (2.54)	44.24 (-50.84)	88.22 (**-1.98**)
	Mkrum	97.95	95.19 (5.77)	95.21 (5.76)	95.39 (5.99)	93.21 (3.57)	89.37 (**-0.70**)
	Fltrust	97.97	92.19 (2.43)	93.10 (3.44)	93.12 (3.47)	94.84 (5.38)	89.18 (**-0.91**)
	CC	97.97	94.48 (4.98)	94.66 (5.18)	94.54 (5.04)	97.07 (7.86)	93.80 (**4.22**)
	DNC	95.48	92.76 (3.07)	93.36 (3.73)	93.36 (3.73)	93.22 (3.58)	92.46 (**2.73**)
			Target Acc 85%				
	FedAvg	97.97	96.73 (13.80)	93.18 (9.62)	92.84 (9.22)	83.21 (-2.11)	85.70 (**0.82**)
	Median	97.78	97.73 (14.98)	92.76 (9.13)	92.84 (9.22)	51.74 (39.13)	88.37 (**3.96**)
	Trmean	97.98	96.02 (12.96)	92.88 (9.27)	92.43 (8.74)	95.84 (12.75)	84.45 (**-0.65**)
	NB	97.97	92.82 (9.20)	92.78 (9.15)	93.02 (9.44)	88.35 (3.94)	86.15 (**1.35**)
	Bulyan	97.94	92.35 (8.65)	92.90 (9.29)	92.29 (8.58)	98.68 (16.09)	87.94 (**3.46**)
	Mkrum	97.95	95.19 (11.99)	95.21 (12.01)	95.39 (12.22)	86.71 (2.01)	85.40 (**0.47**)
	Fltrust	97.97	92.19 (8.46)	93.10 (9.53)	93.12 (9.55)	93.00 (9.41)	87.94 (**3.46**)
	CC	97.97	94.48 (11.15)	94.66 (11.36)	94.54 (11.22)	94.86 (11.60)	83.72 (**-1.51**)
	DNC	97.96	92.76 (9.13)	93.36 (9.84)	93.36 (9.84)	99.47 (17.02)	86.38 (**1.62**)
			Target Acc 80%				
	FedAvg	97.97	96.73 (20.91)	93.18 (16.48)	92.84 (16.05)	92.39 (15.49)	80.68 (**0.85**)
	Median	97.78	97.73 (22.16)	92.76 (15.95)	92.84 (16.05)	35.52 (55.60)	69.70 (**-12.88**)
	Trmean	97.98	96.02 (20.03)	92.88 (16.10)	92.43 (15.54)	97.44 (21.80)	78.75 (**-1.56**)
	NB	97.97	92.82 (16.03)	92.78 (15.98)	93.02 (16.28)	46.36 (-42.05)	79.87 (**-0.16**)
	Bulyan	97.94	92.35 (15.44)	92.90 (16.13)	92.29 (15.36)	68.69 (-14.14)	77.84 (**-2.70**)
	Mkrum	97.95	95.19 (18.99)	95.21 (19.01)	95.39 (19.24)	25.52 (-68.10)	80.02 (**0.03**)
	Fltrust	97.97	92.19 (15.24)	93.10 (16.38)	93.12 (16.40)	95.11 (18.89)	77.47 (**-3.16**)
	CC	97.97	94.48 (18.10)	94.66 (18.32)	94.54 (18.18)	92.39 (15.49)	76.85 (**-3.94**)
	DNC	97.96	92.76 (15.95)	93.36 (16.70)	93.36 (16.70)	92.62 (15.77)	82.54 (**3.18**)
			Target Acc 45%				
Tiny ImageNet (ResNet50)	FedAvg	56.58	51.64 (14.76)	57.75 (28.33)	53.20 (18.22	56.37 (25.27)	47.88 (**6.40**)
	Median	52.55	31.05 (-31.00)	34.31 (-23.76)	34.24 (-23.91)	42.24 (-6.13)	46.96 (**4.36**)
	Trmean	55.80	40.58 (-9.82)	33.77 (-24.96)	39.82 (-11.51)	55.75 (23.89)	45.51 (**1.13**)
	NB	56.76	53.07 (17.93)	52.95 (17.67)	53.09 (17.98)	55.75 (23.89)	43.06 (**-4.31**)
	Bulyan	55.80	25.25 (-43.89)	35.56 (-20.98)	33.51 (-25.53)	3.72 (-91.73)	44.45 (**-1.22**)
	Mkrum	55.22	26.54 (-41.02)	18.37 (-59.18)	26.39 (-41.36)	36.7 (18.44)	45.70 (**1.56**)
	Fltrust	55.35	32.81 (-27.09)	47.08 (4.62)	53.45 (18.78)	55.48 (23.29)	45.31 (**0.69**)
	CC	53.09	31.02 (-31.07)	31.36 (-30.31)	31.26 (-30.53)	46.39 (3.09)	44.96 (**-0.09**)
	DNC	52.92	47.87 (6.38)	47.73 (6.07)	14.29 (68.24)	48.88 (8.62)	46.80 (**4.00**)
			Target Acc 40%				
	FedAvg	56.58	51.64 (29.10)	57.75 (44.38)	53.20 (33.00)	38.79 (-3.03)	40.42 (**1.05**)
	Median	52.55	31.05 (-22.38)	34.31 (-14.23)	34.24 (-14.40)	43.46 (8.65)	39.45 (**-1.37**)
	Trmean	55.80	40.58 (1.45)	33.77 (-15.58)	39.82 (-0.45)	46.95 (17.38)	40.12 (**0.30**)
	NB	56.76	53.07 (32.68)	52.95 (32.38)	53.09 (32.73)	34.09 (-14.78)	38.74 (**-3.15**)
	Bulyan	55.80	25.25 (-36.88)	35.56 (-11.10)	33.51 (-16.23)	38.9 (-2.75)	39.52 (**-1.20**)
	Mkrum	55.22	26.54 (-29.91)	18.37 (-54.08)	26.39 (-34.03)	35.61 (-10.98)	40.82 (**2.05**)
	Fltrust	55.35	32.81 (-15.98)	47.08 (17.70)	53.45 (33.63)	35.97 (-10.08)	37.86 (**-5.35**)
	CC	53.09	31.02 (-22.45)	31.36 (-21.60)	31.26 (-21.85)	41.76 (4.40)	40.91 (**2.27**)
	DNC	52.92	47.87 (19.68)	47.73 (19.33)	14.29 (-64.28)	40.64 (1.60)	39.31 (**-1.73**)
			Target Acc 35%				
	FedAvg	56.58	51.64 (47.54)	57.75 (65.00)	53.20 (52.00)	50.78 (45.09)	34.99 (**-0.03**)
	Median	52.55	31.05 (-11.29)	33.52 (-4.23)	33.78 (-3.49)	32.6 (-6.86)	33.88 (**-3.20**)
	Trmean	55.80	40.58 (15.94)	33.77 (-3.51)	39.82 (-13.77)	47.80 (36.57)	34.27 (**2.09**)
	NB	56.76	53.07 (51.63)	52.95 (51.29)	53.09 (51.69)	48.78 (39.37)	35.51 (**1.46**)
	Bulyan	55.80	25.25 (-27.86)	35.56 (1.60)	33.51 (-4.26)	3.08 (-91.20)	35.30 (**0.86**)
	Mkrum	55.22	26.54 (-24.17)	18.37 (-47.51)	26.39 (-24.60)	38.26 (9.31)	35.39 (**1.11**)
	Fltrust	55.35	32.81 (-6.26)	47.08 (34.51)	53.45 (36.54)	45.64 (30.40)	35.84 (**2.40**)
	CC	53.09	31.02 (-11.37)	31.36 (-10.40)	31.26 (-10.69)	43.40 (24.00)	36.43 (**4.09**)
	DNC	52.92	47.87 (36.77)	47.73 (36.37)	14.29 (59.17)	48.12 (37.49)	35.14 (**0.40**)

Table 1: The comparison of accuracy of global model between different attacks on CIFAR10, MNIST and Tiny ImageNet against different AGRs. More experimental results against different AGRs under various attack objectives are demonstrated in Supp.-2.2.

30% and 10% for CIFAR10, in which 60% is set for reference model and other objectives are set by adjusting C. For

MNIST, we set 90% for reference model while 85%, 80%, 50% and 10% are obtained by adjusting C. For Tiny ImageNet, we set 45% as the accuracy of the reference model and 40%, 35% and 0.5% are adjusted by C. The target accuracy of 10% for CIFAR10, 10% for MNIST and 0.5% for Tiny ImageNet are equivalent to random guessing among the classes of the dataset settings.

However, for FMPA we obtain multiple reference models for each attack objective. We compare our method with existing attacks including LIE [Baruch *et al.*, 2019], Min-Max [Shejwalkar and Houmansadr, 2021], Min-Sum [Shejwalkar and Houmansadr, 2021], and FMPA [Zhang *et al.*, 2023]. The introduction of each attack is illustrated in Supp.-2.1 and their computation costs are reported in Supp.-2.2.

Evaluation Defenses
In this paper, we consider various defenses such as FedAvg [McMahan *et al.*, 2017], Median [Yin *et al.*, 2021], Trmean [Yin *et al.*, 2021], Norm-Bounding (NB) [Sun *et al.*, 2019] Bulyan [Mhamdi *et al.*, 2018], Mkrum [Blanchard *et al.*, 2017], Fltrust [Cao *et al.*, 2022], CC [Karimireddy *et al.*, 2021], and DNC [Shejwalkar and Houmansadr, 2021]. The details of each defense is introduced in the Supp.-2.1.

Evaluation Metric
We set the accuracy of attack objective as A_0, if the final accuracy output is A_T, then $\delta = ((A_T - A_0)/A_0) \times 100\%$ demonstrates the difference between the attack objective and the real accuracy. When we compare the performance of different attacks, we use the absolute value of δ. The lower the $|\delta|$ is, the better the performance of attack achieves. We also define the evaluation metric $\theta = |\delta|_{\text{OA}}/|\delta|_{\text{FedSA}}$ to evaluate the performance of our attack ($|\delta|_{\text{FedSA}}$) comparing to other attacks ($|\delta|_{\text{OA}}$).

4.2 Experimental Results
The experiment results against various AGRs with different desired attack objectives are concluded in Table 1. More experimental results under various attack objectives are demonstrated in Supp-2.2. In general, our FedSA achieves the lowest $|\delta|$ and outperform all other attacks, demonstrating its ability in reaching to the attack objective within a small range of loss. The average values of $|\delta|$ across all the AGRs for CIFAR10 are 2.18% under attack objective 60%, 2.61% under attack objective 55%, 1.62% under attack objective 50%, 4.43% under attack objective 30% and 4.56% under attack objective 10.00%. As for MNIST dataset, the average values among all AGRs for MNIST are 1.62% under attack objective 90% and 1.92% under attack objective 85%, 3.16% under attack objective 80%, 7.41% under attack objective 50%, and 3.37% under attack objective 10.00%. The averages against all AGRs are 2.64% for attack objective 45%, 2.05% for attack objective 40%, 1.74% for attack objective 35%, and 4.00% under attack objective 0.50% on Tiny ImageNet.

In addition, our FedSA successfully evades all AGRs and outperforms all other attacks. As shown in Fig. 4, our FedSA can stably converge to it without being detected by AGRs. In contrast, FMPA is detected by AGRs in various scenarios, leading to the test accuracy in the presence of FMPA being close to the global optimal without attack. Moreover,

the outcomes in the main experiments demonstrates that our FedSA can realize precise control and improve the θ mostly by 5226.00× on FMPA, 4737.00× on LIE, 4723.00× on Min-Max and 4156.00× on Min-Sum across all the scenarios.

We now discuss the results on each dataset in detail. For experiments on CIFAR10 dataset, as for different attack objectives, the lowest accuracy difference $|\delta|$ can be realized against Norm-Bounding at 0.10%, Mkrum at 0.15%, Bulyan at 0.34%, CC at 0.10%, and Trmean at 0.30% for 60%, 55%, 50%, 30% and 10% respectively. Moreover, FMPA is found out by Bulyan and Mkrum for 50%.

For experiments on MNIST dataset, the lowest $|\delta|$ can be found 0.27% against Trmean for 90%, 0.47% against Mkrum for 85%, 0.03% against Mkrum for 80%, and 0.54% against Norm-Bounding for 50%. As for attack objective 85%, except the maximum accuracy difference $|\delta|$ of our attack 3.46% against Fltrust, all of the accuracy difference is reduced to under 2%. It is evident that FMPA is detected by Bulyan, CC and DNC for 85% due to the high accuracy, additionally, FMPA is detected by all AGRs except Median and Mkrum for 90% and Bulyan for 80%. The results are nearly reaching or has reached the accuracy with no attack, which means FMPA cannot precisely control the accuracy to attack objective.

For experiments on Tiny ImageNet, the accuracy difference $|\delta|$ has the lowest value against CC at 0.09%, Trmean at 0.30%, DNC at 0.40%, and Norm-Bounding, CC, and DNC at 2.00% for attack objectives 45%, 40%, 35% and 0.5%. According to results, FMPA is noticed by all AGRs except Median, CC and DNC for 45%, and Median, Bulyan and Mkrum for 35%.

However, the performance against Median is slightly worse than other AGRs, which might be due to the significant less amount of malicious clients.

4.3 Ablation Study

In this section, we evaluate the impact of various factors on FedSA's performance, including the attack speed under different k, the proportion of malicious clients, degree of Non-IID, number of clients and sampling rate of clients. More ablation study under different settings are demonstrated in Supp.-2.3.

Impact of k on the attack speed. We illustrated the effect of different k on CIFAR10 with targeted accuracy at 60% against Mkrum. According to the result shown in Figure 5a, as the increasing value of k, the attack speed of reaching the target objectives will be faster.

Impact of the proportion of malicious clients. We compared the effect of different proportion of malicious clients including 5%, 10%, 15% and 20% on CIFAR10 with targeted accuracy at 55% against various AGRs. According to the result shown in Figure 5b, as the increase of the proportion of malicious clients, the outcomes indicate that we can still achieve attack objective with 20% malicious clients without being detected by central server.

Impact of the different degree of Non-IID dataset. We have evaluated the performance under different degrees of
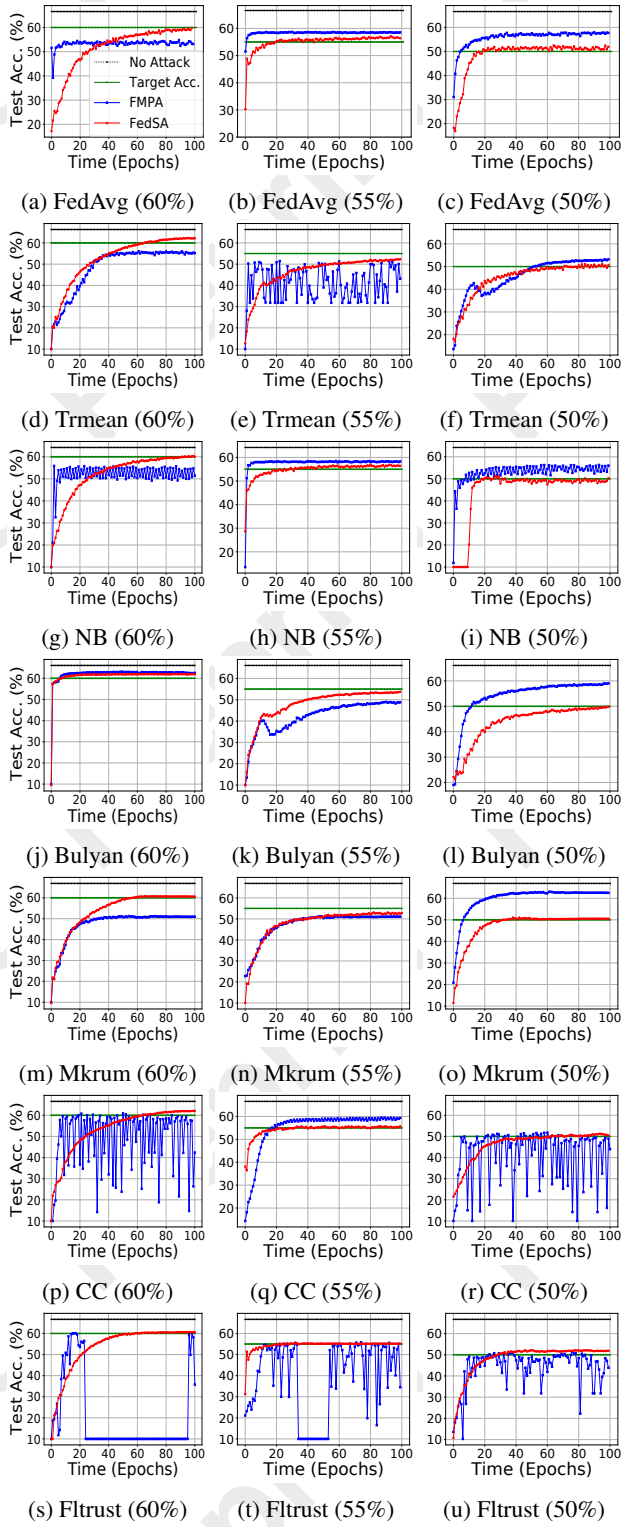
Figure 4: Comparison of FMPA and FedSA against various AGRs with different attack objectives on CIFAR10 with AlexNet. Comparison figures on MNIST and Tiny ImageNet are given in Supp.-2.2.

Non-IID scenarios include 0.1, 0.3, 0.5, 0.7 and 0.9 on CIFAR10 against Mkrum, and the attack objective is set as 60%.

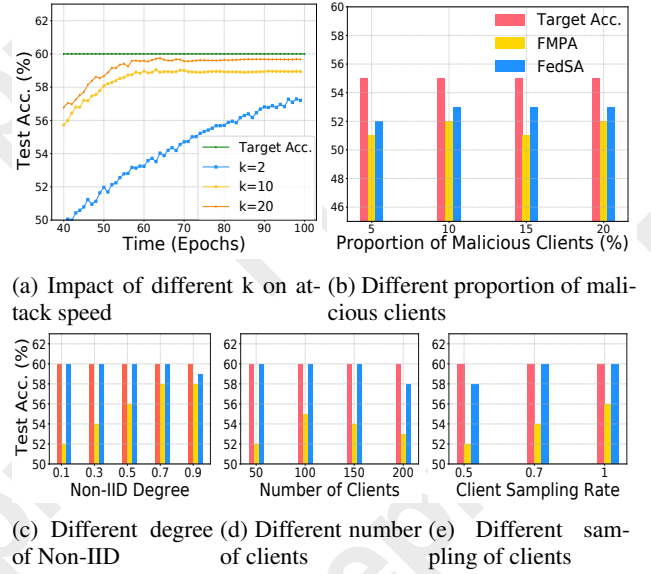The outcomes, compared to other attacks are shown in Fig. 5c, demonstrate that our attack not only achieves its objective but also outperforms other attacks across various Non-IID scenarios.

(a) Impact of different k on attack speed
(b) Different proportion of malicious clients

(c) Different degree of Non-IID
(d) Different number of clients
(e) Different sampling of clients

Figure 5: Ablation study results against Mkrum on CIFAR10 with AlexNet.

Impact of the number of clients. In our experiments, 50 clients are participated in federated learning. We also evaluated our model with an increasing number of clients, including 100, 150, and 200 on CIFAR10 dataset with target accuracy 60%. The outcomes is shown in Fig 5d, which demonstrates that our model can outperform other attacks in increasing number of clients scenario.

Impact of the sampling rate of clients We also implement our model with different sampling rate shown in Fig 5e. It is obvious that FedSA can achieve stabler performance with various sampling rates. Experiment results of the ablation studies on more datasets against various AGRs with different target accuracy are given in Supp.-2.3.

5 Conclusion

We have presented a novel controllable attack FedSA to realize precise control on FL. We demonstrated that our attack can bypass all the Byzantine-robust FL algorithms without being detected. We also provided the theoretical proof to guarantee the performance of the attack. We have proven that FedSA can converge to the attack objective with an adjustable speed. Our experiments on benchmarking datasets demonstrates that FedSA can achieve precise control to the attack objective. Our work highlights the need for specialized defense mechanisms to counteract the controllable poisoning attack, which subtly influences the model performance. Future work will include further investigation into dedicated defense mechanisms against FedSA. In addition, we also see new research opportunities in extending FedSA's mechanism to diverse FL scenarios, e.g., asynchronous learning.

Acknowledgements

References

[Alqumsan *et al.*, 2019] Ahmad Abu Alqumsan, Suiyang Khoo, and Michael Norton. Robust control of continuum robots using Cosserat rod theory. *Mechanism and Machine Theory*, 131:48–61, January 2019.

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How To Backdoor Federated Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, June 2020.

[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*, volume 32. Curran Associates, Inc., 2019.

[Bhagoji *et al.*, 2019] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing Federated Learning through an Adversarial Lens, November 2019. arXiv:1811.12470.

[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*, volume 30. Curran Associates, Inc., 2017.

[Cao *et al.*, 2022] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping, April 2022. arXiv:2012.13995.

[Fang *et al.*, 2020] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. pages 1605–1622, 2020.

[Fang *et al.*, 2021] Liandi Fang, Shihong Ding, Ju H. Park, and Li Ma. Adaptive Fuzzy Control for Stochastic High-Order Nonlinear Systems With Output Constraints. *IEEE Transactions on Fuzzy Systems*, 29(9):2635–2646, September 2021.

[Gong *et al.*, 2024] Zirui Gong, Liyue Shen, Yanjun Zhang, Leo Yu Zhang, Jingwei Wang, Guangdong Bai, and Yong Xiang. Agramplifier: Defending federated learning against poisoning attacks through local update amplification. *IEEE Transactions on Information Forensics and Security*, 19:1241–1250, 2024.

[Gong *et al.*, 2025] Zirui Gong, Yanjun Zhang, Leo Yu Zhang, Zhaoxi Zhang, Yong Xiang, and Shirui Pan. Not all edges are equally robust: Evaluating the robustness of ranking-based federated learning. In *2025 IEEE Symposium on Security and Privacy (SP)*, 2025.

[Jagielski *et al.*, 2018] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35, May 2018. ISSN: 2375-1207.

[Jebreel and Domingo-Ferrer, 2023] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. FL-defender: Combating targeted attacks in federated learning. *Knowledge-Based Systems*, 260:110178, 2023.

[Karimireddy *et al.*, 2021] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from History for Byzantine Robust Optimization, June 2021. arXiv:2012.13995.

[Khoo *et al.*, 2009] Suiyang Khoo, Lihua Xie, and Zhihong Man. Robust finite-time consensus tracking algorithm for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, 14(2):219–228, April 2009.

[Khoo *et al.*, 2013] Suiyang Khoo, Juliang Yin, Zhihong Man, and Xinghuo Yu. Finite-time stabilization of stochastic nonlinear systems in strict-feedback form. *Automatica*, 49(5):1403–1410, 2013.

[Lee *et al.*, 2013] Jeng-Dao Lee, Suiyang Khoo, and Zhi-Bin Wang. DSP-Based Sliding-Mode Control for Electromagnetic-Levitation Precise-Position System. *IEEE Transactions on Industrial Informatics*, 9(2):817–827, May 2013. Conference Name: IEEE Transactions on Industrial Informatics.

[Mahloujifar *et al.*, 2019] Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal Multi-Party Poisoning Attacks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4274–4283. PMLR, May 2019.

[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*, pages 1273–1282, April 2017.

[Mhamdi *et al.*, 2018] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The Hidden Vulnerability of Distributed Learning in Byzantium, July 2018. arXiv:1802.07927.

[Muñoz-González *et al.*, 2017] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38, 2017.

[Shejwalkar and Houmansadr, 2021] Virat Shejwalkar and Amir Houmansadr. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *Proceedings 2021 Network and Distributed System Security Symposium*, 2021.

[Shen *et al.*, 2022] Liyue Shen, Yanjun Zhang, Jingwei Wang, and Guangdong Bai. Better Together: Attaining the Triad of Byzantine-robust Federated Learning via Local Update Amplification. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 201–213, Austin TX USA, December 2022. ACM.

[Sun *et al.*, 2019] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can You Really Backdoor Federated Learning?, December 2019. arXiv:1911.07963.

[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. PMLR, August 2020.

[Yang *et al.*, 2017] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017.

[Yin *et al.*, 2011] Juliang Yin, Suiyang Khoo, Zhihong Man, and Xinghuo Yu. Finite-time stability and instability of stochastic nonlinear systems. *Automatica*, 47(12):2671–2677, 2011.

[Yin *et al.*, 2021] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates, February 2021. arXiv:1803.01498.

[Young *et al.*, 1999] K.D. Young, V.I. Utkin, and U. Ozguner. A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, 7(3):328–342, May 1999.

[Yu *et al.*, 2021] Xinghuo Yu, Yong Feng, and Zhihong Man. Terminal Sliding Mode Control – An Overview. *IEEE Open Journal of the Industrial Electronics Society*, 2:36–52, 2021.

[Zhang *et al.*, 2019] Yanjun Zhang, Xin Zhao, Xue Li, Mingyang Zhong, Caitlin Curtis, and Chen Chen. Enabling privacy-preserving sharing of genomic data for gwass in decentralized networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 204–212, 2019.

[Zhang *et al.*, 2020] Yanjun Zhang, Guangdong Bai, Xue Li, Caitlin Curtis, Chen Chen, and Ryan KL Ko. Privcoll: Practical privacy-preserving collaborative machine learning. In *European Symposium on Research in Computer Security*, pages 399–418. Springer, 2020.

[Zhang *et al.*, 2021] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, March 2021.

[Zhang *et al.*, 2023] Hangtao Zhang, Zeming Yao, Leo Zhang, Shengshan Hu, Chao Chen, Alan Liew, and Zhetao Li. Denial-of-service or fine-grained control: Towards flexible model poisoning attacks on federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, 2023.