Good Gradients Poison Your Model: Evading Defenses in Federated Learning via Boundary-adaptive Perturbation

Xiaojie Zhao[1], **Jinqiao Shi** [1] *, **Yi Li** [1] , **Junmin Huang** [1] , **Chongru Fan** [1]

[1] Key Laboratory of Trustworthy Distributed Computing and Service (MoE),
Beijing University of Posts and Telecommunications, Beijing, China
xiaojiezhao@bupt.edu.cn, shijinqiao@bupt.edu.cn, yili@bupt.edu.cn

Abstract

Federated learning (FL) allows for collaborative model training while preserving data privacy, but its distributed nature makes it vulnerable to poisoning attacks. Existing defense methods typically rely on using gradients from multiple clients to define a trusted region, selecting only the trustworthy update (good gradients) within this region for aggregation. Mainstream defense boundaries are categorized as hard boundaries, soft boundaries, and semi-soft boundaries. However, we argue that even good gradients within these boundaries can still be exploited by attackers to poison the model. To tackle this challenge, we introduce a boundary-adaptive attack method that leverages the directional properties of optimization techniques to derive baseline poisoned gradients. Through iterative perturbation, it generates seemingly innocent gradients that subtly deviate from the global model. Our extensive study on benchmark datasets and mainstream defensive mechanisms confirms that the proposed attack raises a significantly threat to the integrity and security of FL practices, regardless of the flourishing of robust FL methods.

Introduction

Federated learning (FL) (Konečnỳ et al. 2016) uses a distributed training approach to train and optimize models without sharing users' original data, fully protecting user privacy and solving the problem of data silos. In recent years, FL has found extensive applications in various fields, including user behavior analysis (Hard et al. 2018), smart healthcare (Antunes et al. 2022), wireless communications (Yang et al. 2022), signal recognition (Shi et al. 2020), and security detection (Khramtsova et al. 2020). It has also led to the development of several open-source frameworks, such as FATE (Liu et al. 2021), PySyft (Ryffel et al. 2018), PaddleFL (Ma et al. 2019), and TFF (Bonawitz 2019).

However, due to its large-scale, distributed nature and the strong autonomy of the training clients, attackers can launch poisoning attacks during the training process by modifying local training data or uploading gradients to reduce model accuracy (Fang et al. 2020). By poisoning local data, malicious clients can drive the global model to learn incorrect knowledge or embed backdoors (Wang et al. 2024; Tao et al. 2024; Liu et al. 2024). Alternatively, they may submit malicious models to the central server to disrupt the training process, commonly known as the model poisoning attack (MPA)(Tan et al. 2023; Zhang et al. 2023; Shan et al. 2024). Consequently, the global model suffers from high testing error indiscriminately and eventually causes model divergence.

These attacks can be classified as either untargeted attacks, aimed at reducing the overall quality of the learned model, or targeted, where the goal is to manipulate the model into misclassifying samples into an adversary's desired class. **Our work focuses on the untargeted adversarial model poisoning attack, which not only degrades model performance but also evades existing defenses, posing a significant threat to FL**.

To mitigate model poisoning attacks (MPAs) in FL, a growing number of defenses have been proposed (Yang et al. 2024), where the server employs a robust aggregation algorithm. We argue that the key factor in the success of existing defenses is the boundary of model updates available to clients for selection. Specifically, boundary patterns are introduced based on the characteristics of existing defenses. We introduce three kinds of boundaries, namely hard boundary, semi-soft boundary, and soft boundary. **The hard boundary** is the boundary that we believe is conducive to the training of the FL model, and the outside of the boundary does not participate in the aggregation in this round, such as Multi-Krum (Blanchard et al. 2017), SignGuard (Xu et al. 2022), DnC (Shejwalkar and Houmansadr 2021), FLDetector (Zhang et al. 2022), Median(Yin et al. 2018), Trmean (Yin et al. 2018), GeoMed (Chen, Su, and Xu 2018), Bulyan(Guerraoui, Rouault et al. 2018), etc. **The semi-soft boundary** uses some parameter characteristics to weaken the contribution of suspected malicious gradients to global model training, such as FoolsGold (Fung, Yoon, and Beschastnikh 2020), RoseAgg (Yang et al. 2024), etc. **The soft boundary** is a smoothing function to weaken the contribution of suspected malicious gradients to global model training, such as FedAvg (Li et al. 2019), FLAME (Nguyen et al. 2022), RLR (Ozdayi, Kantarcioglu, and Gel 2021), etc. With the development of defense algorithms, soft boundaries are the trend to optimize aggregation rules. In addition, considering the overall gradient information and fine-grained gradient information, the boundary is divided into

*Corresponding authors: Jinqiao Shi

Classification	Defenses	Principles	Boundary pattern
Gradients of the Overall Dimension	FedAvg'19	Select all local gradients for average aggregation	○
	Multi-Krum'17	Select k local gradients for aggregation	●
	FoolsGold'20	Similarity-based local gradients for aggregation	◐
	SignGuard'22	Select m local gradients for aggregation	●
	RoseAgg'24	Select k local gradients for aggregation	◐
	DnC'21	Select k local gradients for aggregation	●
	FLAME'22	Select k local gradients for aggregation	○
	FLDetector'22	Select k local gradients for aggregation	●
	RLR'21	Adjusting the learning rate for local clients	○
Gradients of Fine-grained Axis Dimensions	Median'18	Select median local gradients for aggregation	●
	Trmean'18	Select trimmed local gradients for aggregation	●
	GeoMed'18	Select geometric local gradients for aggregation	●
	Bulyan'18	Select k local gradients for aggregation	●

Table 1: Classification and principles of defense methods with corresponding boundary patterns. ● denotes hard boundary. ◐ denotes semi-soft boundary. ○ denotes soft boundary.

the overall boundary and the fine-grained boundary. Hard boundaries like the median defense algorithm belong to fine-grained boundaries. The specific classification table of defense algorithms is shown in Tab. 1.

However, current MPAs suffer from the following limitations. Although research uses optimized or adaptive attack methods to evade existing defense algorithms, the effect is suboptimal as new defense algorithms emerge. In this attack-defense game, attack methods are easier to detect.

To address the aforementioned challenges, we present a novel boundary-based model poisoning attack on FL by designing a new adversarial simulation optimization-based framework. The attacker defines the gradient boundary for each round. The boundary can be divided into the hard boundary, the semi-soft boundary, and the soft boundary. We design boundary functions to optimize the novel attack methods proposed. The attacker then adversarially extracts the features of the benign boundary gradient, e. g. the gradient size, direction. Fig. 1 intuitively illustrates the difference between the proposed attack and existing attacks through visualizing the loss landscape.

Our contributions are highlighted as follows:

- We propose a new model poisoning attack, namely BAPerturb. By finding the boundary gradient and constructing malicious updates based on features, the accuracy of the global model decreases after multiple iterations.
- To the best of our knowledge, we introduce a new concept, namely, boundary. This approach breaks conventional defense principles by asserting that good gradients can positively contribute to the accuracy of the trained global model.
- The dynamic boundaries of multiple rounds are first considered, focusing on their vertical and horizontal information in cross-device FL systems, and we explore their role in poisoning attacks.

Related Work

Model Poisoning Attack against Federated Learning. Model poisoning attack allows for arbitrary manipulation of the entire training process by changing parameters and loss functions. The purpose of a model poisoning attack is to set up carefully crafted gradient updates in a compromised or fake client so that the aggregated global gradient deviates from the original global model update, resulting in a degradation of the prediction or classification performance of the trained global model. Simply constructing carefully designed gradients and evading aggregation algorithm detection are two key elements for successfully executing a model poisoning attack.

Existing attack methods often appear random in early stages, rendering them readily detectable. Some researchers have started to study attacks based on optimization as well as adaptation, such as LIE (Baruch, Baruch, and Goldberg 2019), but the effect is not significant in cross-device FL scenarios, especially in MNIST and Fashion-MNIST. The main reason is the reduced probability of malicious clients being selected, as well as the less obvious attack stratagems. The proposed Fang attack, along with the improved Min-Max attack and the Min-Sum attack, were examined. Three types of perturbations were used, including the inverse unit vector, the inverse standard deviation vector, and the inverse sign vector. But the effect of multiple rounds of iteration on the global model accuracy is not taken into account. Jian Xu et al. (Yang et al. 2024) proposed a multi-target attack, but it is not very effective against mainstream defense failures such as roseagg. Tongsai Jin et al. (Jin et al. 2023) conducts partial perturbations of a small number of well-selected model parameters against a small number of parameters, but the effect is not very obvious with the state-of-the-art defense algorithms.

Byzantine-robust Aggregation Methods. These methods follow the principle of anomaly detection (Blanchard et al. 2017; Yin et al. 2018; Xu et al. 2022). The core assumption is that the parameters of all benign local models re-

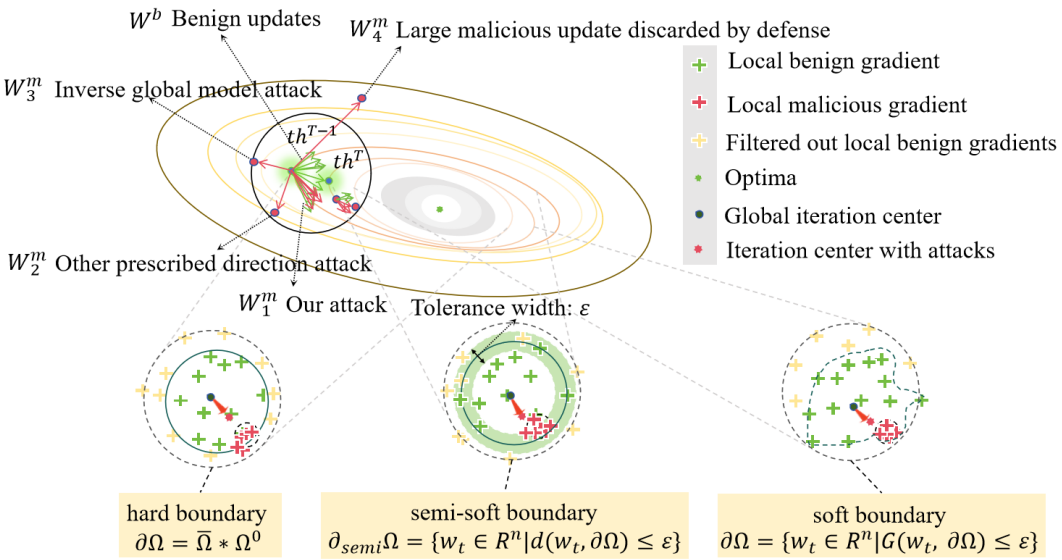

Figure 1: Schematics of our attack: the visualization process of the boundary is hard boundary, semi-soft boundary, and soft boundary. The black boundary shows an effective attack. The gradient information of the client outside the black boundary will be invalid. Our attack method (e.g., W_1^m)is based on the optimal poisoning attack on the boundary with perturbations. The red arrow represents a carefully designed malicious gradient update based on an optimization direction (e.g., W_2^m and W_3^m). The malicious local update of the red arrow will be invalid outside the boundary (e.g., W_4^m).

main within a bounded region centered on the global model. Therefore, the poisoned local model is considered an outlier and deviates from the benign local model. However, existing defense algorithms fail to recognize that the proposed attack method targets the boundaries of the benign region.

Our Method: BAPerturb

Federated learning. The global model parameters, w, are trained by combining the data from all clients to ensure optimal performance on the global dataset. Formally, this involves minimizing the mean of the loss function on the server side.

$$\underset{w \in R^d}{argmin} F(w) = \sum_{n=1}^{N} \frac{|D_n|}{|D|} F_n(w) \tag{1}$$

Where $F_n(w)$ is the loss function of the n^{th} client. Each client n uses its data D_n locally to update the model parameters. Stochastic gradient descent (SGD) is generally used for optimization.

$$w_n^{t+1} = w^t - \eta \bigtriangledown F_n(w^t) \tag{2}$$

The server calculates a summary of updates from all of the clients.

$$w^{t+1} = \sum_{n=1}^{N} \frac{|D_n|}{|D|} w_n^{t+1} \tag{3}$$

In the paper, different from previous work (Xu et al. 2022), we mainly consider cross-device FL, which is more commonly used and makes attacks more difficult.

Attacker's goals. Similar to most model poisoning attacks, the goal is to damage the accuracy of the global model for all targets, making it unable to converge or reducing the overall prediction (or classification) performance of the model.

Attacker's capabilities. Assume that the attacker controls m local malicious clients or that different attackers collude with each other. Specifically, like the Sybil attack on distributed systems (Douceur 2002) or the fake client proposed (Cao and Gong 2023), the attacker can inject m local malicious clients, or compromise m benign local clients, or control m zombie clients. However, assume that the number of working devices under the control of the attacker is less than 50% (otherwise, it is easy to manipulate the global model).

Problem Description

The attacker can directly control multiple compromised clients and manipulate their uploaded models to influence the behavior of the learning algorithm according to a preemptive goal. We assume that there are m clients compromised by the attacker, and it will directly change the output of these clients to bias the learning model towards the goal. We also define b as the number of benign clients, and we have $m + b = u$. We define m as the set of these compromised clients, B as the set of benign clients, and U as the set of all clients, where $B = U/M$. Specifically, in each round of communication, the benign client calculates a local parameter vector, but each compromised client provides an unreliable parameter vector. With a specific aggregation rule and all uploaded models, the server can update the global model.

Demarcation of Poisoning Area Boundary

Motivation We consider that the key to the effectiveness of the defense algorithm is geometrically the existence of a region in which the gradient information within the bounded region is selected for aggregation by the variant, and outside the boundaries is weakened or filtered out of the influence on the global model training process. Based on the above principles, we give definitions of hard, semi-soft, and soft boundaries in a geometric form.

Based on the definition of the boundary categories, the thresholding of the soft boundary based on the cosine recognition degree, we select the more centralized gradient information to filter the poisoning gradient information of the benchmark.

Threshold constraints Suppose we have a set of multidimensional gradients $\{w_1^t, w_2^t, ..., w_m^t\}$, the gradients w_i^t of the m clients being controlled are d-dimensional, i. e $w_i^t \in R^d$. The center point w_c^t is approximated by replacing it with the gradient information w^{t-1} sent by the server in the previous round.

For gradient w_i^t of the m clients being controlled , calculate its cosine similarity with the center gradient

$$cos(w_i^t, w_c^t) = \frac{w_i^t \cdot w_c^t}{\|w_i^t\| \, \|w_c^t\|} \tag{4}$$

Determine the boundary based on the value of cosine similarity, and determine whether each vector is inside or outside the boundary. We introduce a threshold τ as a boundary criterion to measure the directional consistency between each client gradient and the center gradient, and accordingly categorize updates as inside-boundary or outside-boundary.If $cos(w_i^t, w_c^t) \geq \tau$, then the gradient $w_i^t$is inside the boundary. If $cos(w_i^t, w_c^t) < \tau$, then the gradient w_i^t is outside the boundary. The value of τ was chosen because the direction of the gradient at this point lies in the middle of the center gradient, which is used to improve the stealthiness of the proposed attack.

Compared with the soft boundary, the Compertz Function is chosen as the adaptive boundary, and the boundary is dynamically adjusted by judging the contribution of the gradient information through the corresponding function value.

The Gompertz function is often used as a mathematical model to describe the update of soft boundaries. Its calculation formula is $C_i = ae^{be^{cr_i}}$, where a, b, and c are standard parameters, a controls the upper asymptote of the curve, b controls the offset along the x-axis, and c controls the growth rate of credibility. In this paper, r_i represents the soft tolerance of the gradient of client n_i, and C_i represents the tolerance of user node n_i. If the server receives the gradientW_t^i of n_i, it can aggregate the gradient into $W_t^i \cdot C_i$.

Optimization Algorithm for Malicious Gradient

Motivation Due to the different forms of boundaries, we specifically solve the poisoning benchmark gradient in three cases. Based on intuition, we select those gradients that are as close to the edge of the defense algorithm as possible. From a defense perspective, the information of the gradient will be aggregated into the global model parameters, increasing the concealment of the attack; from an attack perspective, the center point of the federated learning training model is continuously shifted through multiple rounds of accumulation, increasing the effectiveness of the attack.

Algorithm 1: Gradient Descent for BAPerturb Optimization

Input: Initial gradients $\{w_i^0\}$; Center gradient w_c^t; parameter γ; $\theta_{\min}$, M ;Learning rate α; T
Parameter: Each malicious client i randomly initialize w_i^0 satisfy constraints: $\cos(w_i^0, w_c^t) \geq \cos(\theta_{\min})$ and $\|w_i^0 - w_c^t\| \geq M$
Output: Optimized gradients $\{w_i^*\}$ for malicious clients

for $t = 1$ **to** T **do**
  for each malicious client i **do**
    Compute the gradient of the objective function:
    grad_cos $= -\frac{w_i^t \cdot w_c^t}{\|w_i^t\| \cdot \|w_c^t\|}$
    grad_norm $= \gamma \cdot \frac{w_i^t - w_c^t}{\|w_i^t - w_c^t\|}$
    grad_total = grad_cos + grad_norm
    Update the gradient using gradient descent:
    $w_i^{t+1} = w_i^t - \alpha \cdot$ grad_total
    Check and enforce constraints:
    if $\cos(w_i^{t+1}, w_c^t) < \cos(\theta_{\min})$ **then**
      Adjust w_i^{t+1} to satisfy $\cos(w_i^{t+1}, w_c^t) \geq \cos(\theta_{\min})$
    end if
    if $\|w_i^{t+1} - w_c^t\| < M$ **then**
      Adjust w_i^{t+1} to satisfy $\|w_i^{t+1} - w_c^t\| \geq M$
    end if
  end for
  if the change in w_i^{t+1} is below a certain threshold for all i, or $t == T$ **then**
    Break
  end if
end for
return optimized gradients $\{w_i^*\}$ for malicious clients

Based on the determined hard boundaries, we define the optimization function. Considering the impact of multiple iterations on the global model, the baseline poisoning gradient in a similar direction is selected in each round to make the attack more effective. The specific optimization problem is as follows

$$\begin{aligned} \min_{w_1^t, w_2^t, \ldots, w_m^t} \quad & \sum_{i=1}^{m} \left(1 - \cos(w_i^t, w_c^t) + \gamma \left\|w_i^t - w_c^t\right\|\right) \\ \text{s. t.} \quad & w_i^t \in \left\{P_1^t, P_2^t, \ldots, P_n^t\right\}, \\ & \cos(w_i^t, w_c^t) \geq \cos(\theta_{\min}), \\ & \left\|w_i^t - w_c^t\right\| \geq M \end{aligned} \tag{5}$$

where $P_1^t, P_2^t, ..., P_n^t$ is the gradient set consisting of local clients in each round. We select the gradient information sent by the server in the previous round as an approximation of the gradient at the center point w_c^t. $cos(w_i^t - w_c^t)$ is the cosine similarity between a single gradient and the center point. $\|w_i^t - w_c^t\|$ is the euclidean distance from the gradi-

ent to the mean.The specific optimization problem is solved as outlined in Algorithm 1.

Based on the determined semi-soft boundaries, we identify a gradient parameter whose cosine similarity $cos(w_i^t, w_c^t)$ is close to ζ and use it as the baseline gradient for poisoning. At this time, the gradient is in the middle of the entire federated learning gradient parameter distribution. After multiple rounds of iterations, the entire federated learning gradient parameter deviates from the center. For fine-grained gradient information based on the coordinate dimension, the gradient information of the coordinate dimension is further considered to find the poisoning reference gradient information, such as selecting the median of the i-th dimension as the poisoning reference gradient of the i-th dimension.

Based on the determined soft boundaries, mentioned above, we choose the gradient value with a Gompertz function value close to μ as the poisoning baseline gradient. Similarly, in the coordinate-wise setting, the gradient value whose function value is close to μ in the i-th dimension is selected as the poisoning baseline gradient for that dimension.

Directional Perturbations of Malicious Gradients

Motivation Given that the gradient is a crucial parameter for model training updates, its direction indicates the direction of the update in each dimension. Furthermore, to circumvent existing defense methods, the direction-based perturbation is divided into an overall angle shift and a fine-grained sign shift.

Direction perturbation Consider the most commonly used FedAvg algorithm $\bar{w}$. This can average the existing gradients to get. Use Euclidean distance to calculate the cosine similarity of the center value and each value. The corresponding formula is

$$c_i = cos(w_i, \bar{w}) = \frac{w_i \bar{w}}{\|w_i\| \, \|\bar{w}\|} \tag{6}$$

It is worth noting that the overall perspective shift does not influence the cosine identity between gradients.

The fine-grained symbolic perturbations are as follows. Take the positive flip sign rate $R_{flip}^{+}$and negative flip sign rate R_{flip}^{-}selected in each round, $\kappa_i = min(c_i)$ selects the corresponding vector $w^{'}$, and the angle of the cosine similarity between the later selected vector and the vector is less than $\theta \geq \zeta$. Therefore, the update of the malicious client is

$$\hat{w} = w^{'} \cdot R_{flip}^{+} \cdot R_{flip}^{-} \tag{7}$$

In which, R_{flip}^{+}satisfies the Gaussian distribution and dynamically adjusts the gradient of the global model.

Experimental Evaluation

We conduct experiments to evaluate the performance of BAPerturb. By comparision with previous attack, we demonstrate the effectiveness of our proposed attack scheme. We simulate multiple clients by Python, following previous work (Fang et al. 2020). Model training process with FL is running on one high-performance machine with an Intel Xeon Gold 6133 CPU (2. 50 GHz).

Experimental Setup

Datasets and model architectures. We evaluate the performance of BAPerturb on commonly used MNIST (LeCun et al. 1998), Fashion-MNIST (Cohen et al. 2017) and CIFAR-10 (Krizhevsky, Hinton et al. 2009) datasets. The convolutional neural network LeNet and the residual neural network CNN ResNet20 are used as global models for the MNIST dataset and the Fashion-MNIST dataset, respectively. To demonstrate the impact of attacks on the model, Conv8 and the residual neural network ResNet18 are used as global models for the CIFAR-10 dataset.

Baseline poisoning attack. Referring to related work, we adopt existing attacks with our proposed method for comparison. Such as, Random(Fang et al. 2020), LIE (Baruch, Baruch, and Goldberg 2019), Min-Max(Shejwalkar and Houmansadr 2021), Min-Sum(Shejwalkar and Houmansadr 2021), ByzMean (Xu et al. 2022), FedPerturb(Jin et al. 2023), etc.

Defense methods. We evaluate existing defenses in cross-device FL, such as FedAvg(Li et al. 2019), Multi-Krum (Blanchard et al. 2017), Median(Yin et al. 2018), DnC(Shejwalkar and Houmansadr 2021), TrMean(Yin et al. 2018), GeoMed(Chen, Su, and Xu 2018), Bulyan(Guerraoui, Rouault et al. 2018), FoolsGold(Fung, Yoon, and Beschastnikh 2020), RLR(Ozdayi, Kantarcioglu, and Gel 2021), FLAME(Nguyen et al. 2022), SignGuard(Xu et al. 2022), FLDetector(Zhang et al. 2022) and RoseAgg(Yang et al. 2024).

Measurement metrics. Model Test Accuracy is the proportion of samples correctly predicted by the model to the total number of samples on the test dataset, where ACC_{F_B} and ACC_{F_M} represent the accuracy of the benign model and the malicious model respectively. Obviously, the larger ACC is, the better the global model performance is. We define Attack success rate (Asr) $\varphi = 1 - ACC_{F_M}$ as the degree of model accuracy degradation. For a given attack, the larger φ is, the better the attack effect is.

Parameter and attack settings for FL. We consider a more practical cross-device FL setting. We set the number of clients to 1,000 for MNIST and CIFAR-10, and 3,400 for Fashion-MNIST. In each communication round, 10 % of clients are randomly sampled to participate. The global model is trained for 2,000 rounds. Each selected client performs 2 local epochs on MNIST and Fashion-MNIST, and 5 local epochs on CIFAR-10. To control data heterogeneity, we partition data using a Dirichlet distribution with concentration parameter $\beta = [0, 1]$. Specifically, $\beta = 1$ corresponds to a more non-IID setting, and smaller β yields a distribution closer to IID. For all three datasets, we partition the data into training and testing sets. Specifically, we allocate 80 % of the samples for training and the remaining 20 % for testing.

Attack Performance

We evaluate existing mainstream defenses, both for targeted and untargeted defenses. This includes existing attack methods, ranging from the most basic ones to those that apply fine-grained perturbations. As shown in Tab. 2, most of the

Datasets (Model)	Defenses	Ran -dom'18	LIE'19	Min -Max'21	Min -Sum'21	Byz -Mean'22	FedPer -turb'23	BAPer -turb (ours)
MNIST (LeNet)	FedAvg'19	0. 4409	0. 0365	0. 0462	0. 0432	**0. 8690**	0. 3132	0. 7235
	Multi-Krum'17	0. 0466	0. 0545	0. 0513	0. 0564	0. 0439	0. 0126	**0. 1354**
	Median'18	0. 0526	0. 0485	0. 0495	0. 0496	0. 0493	0. 0164	**0. 3822**
	Trmean'18	0. 0995	0. 0485	0. 0480	0. 0498	0. 0492	0. 0156	**0. 6887**
	GeoMed'18	0. 0438	0. 0390	0. 0402	0. 0391	0. 0415	0. 0127	**0. 8829**
	Bulyan'18	0. 1831	0. 0368	0. 0397	0. 0415	0. 0435	0. 0819	**0. 7218**
	DnC'21	0. 0417	0. 0363	0. 0408	0. 0387	0. 0411	0. 0355	**0. 4929**
	FoolsGold'20	0. 0115	0. 0064	0. 0071	0. 0065	0. 0074	0. 0201	**0. 8718**
	RLR'21	0. 0472	0. 0686	0. 0248	0. 0187	**0. 8865**	0. 6696	0. 7207
	FlAME'22	0. 0103	0. 0098	0. 0096	0. 0095	0. 0097	0. 0146	0. 0165
	FLDetector'22	0. 0400	0. 0125	0. 0171	0. 0231	0. 8865	0. 8724	**0. 9318**
	SignGuard'22	0. 0463	0. 0447	0. 0435	0. 0439	0. 0445	0. 0095	**0. 0792**
	RoseAgg'24	0. 0079	0. 0075	0. 0073	0. 0074	0. 0078	0. 0134	**0. 0188**
Fashion-MNIST (CNN)	FedAvg'19	0. 4810	0. 1287	0. 1574	0. 1427	**0. 9000**	0. 2998	0. 3181
	Multi-Krum'17	0. 1182	0. 1430	0. 1241	0. 1328	0. 1182	**0. 3448**	0. 1255
	Median'18	0. 1235	0. 1491	0. 1476	0. 1507	0. 1505	**0. 3184**	0. 2385
	Trmean'18	0. 2456	0. 1494	0. 1471	0. 1507	0. 1488	0. 2955	**0. 4976**
	GeoMed'18	0. 1180	0. 1314	0. 1313	0. 1305	0. 1156	0. 3216	**0. 7207**
	Bulyan'18	0. 2842	0. 1281	0. 1280	0. 1341	0. 1293	0. 3245	**0. 8470**
	DnC'21	0. 1876	0. 1270	0. 1166	0. 1177	0. 1177	0. 3116	**0. 4323**
	FoolsGold'20	0. 0959	0. 0905	0. 0884	0. 0885	0. 0880	0. 3192	**0. 9000**
	RLR'21	0. 2024	0. 2351	0. 1422	0. 1376	**0. 9000**	0. 2967	0. 3619
	FlAME'22	0. 0947	0. 0969	0. 0929	0. 0953	0. 0932	0. 0934	**0. 1055**
	FLDetector'22	0. 1922	0. 1214	0. 1458	0. 1549	**0. 9000**	**0. 9000**	**0. 9000**
	SignGuard'22	0. 1178	0. 1229	0. 1267	0. 1244	0. 1196	**0. 3085**	0. 1720
	RoseAgg'24	0. 1405	0. 1010	0. 0997	0. 0982	0. 1066	**0. 4860**	0. 1220
CIFAR-10 (ResNet18)	FedAvg'19	**0. 9000**	0. 6803	0. 5220	**0. 9000**	**0. 9000**	0. 7649	**0. 9000**
	Multi-Krum'17	0. 1220	0. 8346	**0. 9000**	0. 7572	**0. 9000**	0. 1385	0. 4559
	Median'18	0. 1136	0. 7363	0. 1135	0. 7458	0. 7845	0. 1290	**0. 8358**
	Trmean'18	0. 2051	0. 7302	0. 7168	0. 7341	0. 7385	0. 1288	**0. 9000**
	GeoMed'18	0. 3203	0. 7066	0. 8447	0. 6536	**0. 9000**	0. 1487	**0. 9000**
	Bulyan'18	0. 7020	0. 6446	0. 8501	0. 8349	0. 6576	0. 7205	**0. 9000**
	DnC'21	0. 2106	0. 4883	0. 2318	0. 1476	0. 4019	0. 5720	**0. 8462**
	FoolsGold'20	0. 1950	0. 1164	0. 1162	0. 1187	0. 1823	0. 2728	**0. 9000**
	RLR'21	0. 6359	0. 7763	0. 5588	0. 5719	**0. 9000**	0. 1525	**0. 9000**
	FlAME'22	0. 1449	0. 1477	0. 1520	0. 1514	0. 1449	**0. 7652**	0. 1786
	FLDetector'22	**0. 9000**	0. 1476	0. 1494	**0. 9000**	**0. 9000**	0. 1487	**0. 9000**
	SignGuard'22	0. 1097	0. 1122	**0. 8931**	0. 1214	0. 1353	0. 1279	0. 7930
	RoseAgg'24	0. 1768	0. 3902	0. 3220	0. 4219	**0. 4359**	0. 1492	0. 1991

Table 2: Comparison of the attack impact between MPAs and BAPerturb.**Bold** denotes optimal solutions. Underline denotes sub-optimal solutions. Gray highlight denotes non-convergence.

attack methods are ineffective in cross-device FL scenarios, and the adaptive attacks fail against existing defense algorithms, except for individual ones like those designed by ByzMean for FedAvg and RLR, which are effective. The main reason is that the probability of the malicious client being selected in each round is much lower compared to that of cross-silo FL, which makes the training accuracy of the attack on the global model much weaker.

In addition, different datasets as well as models also have small fluctuations in the convergence as well as accuracy of the global model. We are prompted to consider the dataset as well as the model in the display scenario when designing the attack algorithm, and choose a more appropriate attack method. Existing attack methods demonstrate their adversarial advantages in cross-device FL. For example, in the GeoMed and Bulyan defense algorithms on the MNIST dataset, the attack effectiveness is improved by 72× and 25×, respectively, compared to no attack, highlighting the effectiveness

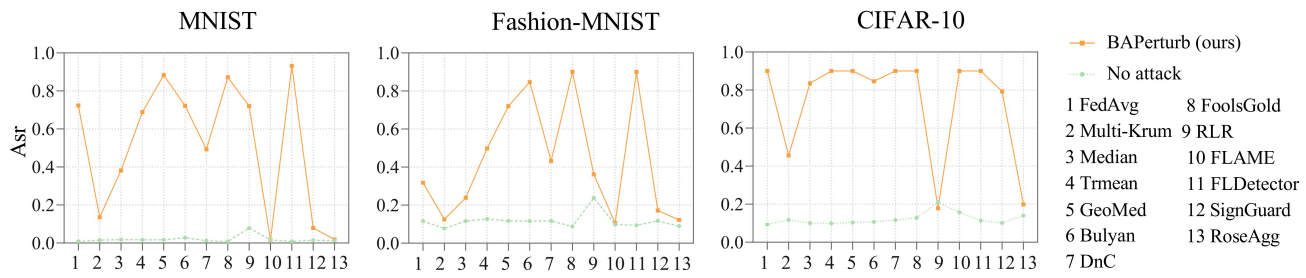

Figure 2: Comparison of the attack impact between no attack and BAPerturb under mainstream defenses.

of these methods, as shown in Fig. 2. However, existing attacks perform poorly under some defenses, and we will further improve them in future work.

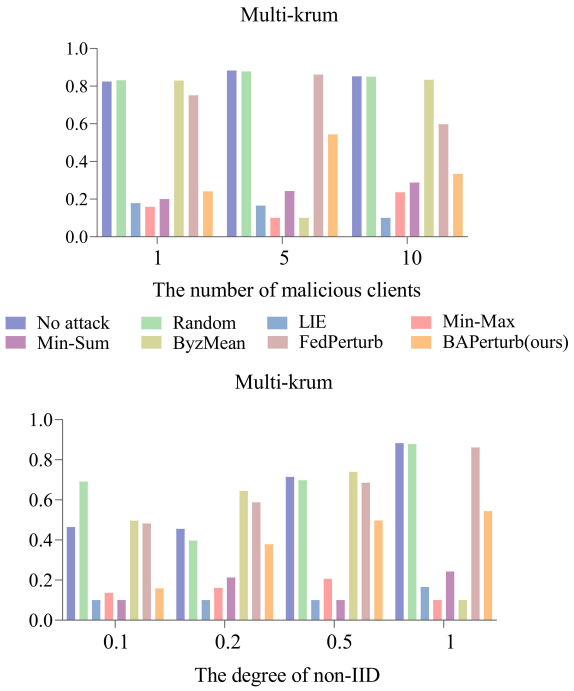

Figure 3: Impact of local training epochs and degree of non-IID on existing attacks and proposed attack.

Attack Effectiveness

Impact of the local training epochs. The epoch of local training also has an impact on the accuracy of the global model. Based on the above principle, we consider the effect of different locally trained epochs of FL on the accuracy of the global model. Fig. 3 shows the accuracy of different locally trained epochs on the global model. Under no-attack or random-perturbation settings, the global model accuracy remains consistently high, indicating that Multi-krum can stably train under benign conditions. However, when targeted poisoning attacks are present, its robustness varies significantly. Some attacks (e.g., LIE and Min-Max/Min-Sum) substantially reduce the accuracy, suggesting that Multi-krum is not effective against all attack types. Meanwhile, BAPerturb more consistently degrades model performance across different numbers of malicious clients and shows stronger attack effectiveness than FedPerturb.

Impact of the degree of non-IID. We consider more realistic data distributions, ranging from approximately even-numbered datasets to very extreme even-numbered datasets. In the experimental setup, we consider the CIFAR-10 using ResNet18 in FL. Fig. 3 shows the impact of varying degrees of non-IID malicious clients in FL on the global model accuracy. Multi-krum exhibits stronger baseline performance when the data is closer to IID, whereas severe non-IID heterogeneity leads to a substantial performance decline. Across all non-IID settings, BAPerturb consistently yields lower global accuracy than FedPerturb, suggesting that our method exerts a stronger attack impact and is more effective at evading Multi-krum's filtering mechanism.

Conclusion and Future Work

We propose a novel, boundary-based model poisoning attack on FL by designing a new adversarial simulation optimization-based framework. The attacker defines the gradient boundary for each round. Specifically, according to the existing principle of aggregation rules, the boundary can be divided into a hard boundary, a semi-soft boundary, and a soft boundary. We design boundary functions to optimize the novel attack methods proposed. From the defender's perspective, the breakdown of the perceptually benign gradient contributes to the accuracy of the global model. In the future, it is important to consider the iteration center of the global model, where the attack is controlled to achieve the targeted attack.

Existing defense methods mostly rely on static boundary definitions, and attacks are exploiting vulnerabilities in these static boundaries. Therefore, a dynamic boundary detection mechanism can be designed to monitor and adjust the boundary in real time to defend against attacks. In practical applications, it is also necessary to combine multi-dimensional monitoring tools to further strengthen the security protection of FL.

Acknowledgments

This work is supported by the Beijing Natural Science Foundation (L254017) and the Academician Fang Binxing Workstation in Hainan Province, China (Grant No. YSGZZ2023003) and the specific research fund of The Innovation Platform for Academicians of Hainan Province, China (Grant No. YSPTZX202506).

References

Antunes, R. S.; André da Costa, C.; Küderle, A.; Yari, I. A.; and Eskofier, B. 2022. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–23.

Baruch, G.; Baruch, M.; and Goldberg, Y. 2019. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.

Bonawitz, K. 2019. Towards federated learning at scale: Syste m design. *arXiv preprint arXiv:1902.01046*.

Cao, X.; and Gong, N. Z. 2023. MPAF: Model Poisoning Attacks to Federated Learning Based on Fake Clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Y.; Su, L.; and Xu, J. 2018. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, 96–96.

Cohen, G.; Afshar, S.; Tapson, J.; and Van Schaik, A. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, 2921–2926. IEEE.

Douceur, J. R. 2002. The sybil attack. In *International workshop on peer-to-peer systems*, 251–260. Springer.

Fang, M.; Cao, X.; Jia, J.; and Gong, N. 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, 1605–1622.

Fung, C.; Yoon, C. J.; and Beschastnikh, I. 2020. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 301–316.

Guerraoui, R.; Rouault, S.; et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, 3521–3530. PMLR.

Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Jin, T.; Fu, Z.; Meng, D.; Wang, J.; Qi, Y.; and Cao, G. 2023. FedPerturb: Covert Poisoning Attack on Federated Learning via Partial Perturbation. In *ECAI 2023*, 1172–1179. IOS Press.

Khramtsova, E.; Hammerschmidt, C.; Lagraa, S.; and State, R. 2020. Federated learning for cyber security: SOC collaboration for malicious URL detection. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 1316–1321. IEEE.

Konečnỳ, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

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

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Liu, Y.; Fan, T.; Chen, T.; Xu, Q.; and Yang, Q. 2021. Fate: An industrial grade platform for collaborative learning with data protection. *Journal of Machine Learning Research*, 22(226): 1–6.

Liu, Z.; Wang, T.; Huai, M.; and Miao, C. 2024. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14115–14123.

Ma, Y.; Yu, D.; Wu, T.; and Wang, H. 2019. PaddlePaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1): 105–115.

Nguyen, T. D.; Rieger, P.; De Viti, R.; Chen, H.; Brandenburg, B. B.; Yalame, H.; Möllering, H.; Fereidooni, H.; Marchal, S.; Miettinen, M.; et al. 2022. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, 1415–1432.

Ozdayi, M. S.; Kantarcioglu, M.; and Gel, Y. R. 2021. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9268–9276.

Ryffel, T.; Trask, A.; Dahl, M.; Wagner, B.; Mancuso, J.; Rueckert, D.; and Passerat-Palmbach, J. 2018. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*.

Shan, S.; Ding, W.; Passananti, J.; Wu, S.; Zheng, H.; and Zhao, B. Y. 2024. Nightshade: Prompt-Specific Poisoning Attacks on Text-to-Image Generative Models. In *2024 IEEE Symposium on Security and Privacy (SP)*, 212–212. IEEE Computer Society.

Shejwalkar, V.; and Houmansadr, A. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

Shi, J.; Zhao, H.; Wang, M.; and Tian, Q. 2020. Signal recognition based on federated learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 1105–1110. IEEE.

Tan, S.; Hao, F.; Gu, T.; Li, L.; and Liu, M. 2023. Collusive model poisoning attack in decentralized federated learning. *IEEE Transactions on Industrial Informatics*.

Tao, G.; Wang, Z.; Feng, S.; Shen, G.; Ma, S.; and Zhang, X. 2024. Distribution preserving backdoor attack in self-supervised learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2029–2047. IEEE.

Wang, H.; Xiang, Z.; Miller, D. J.; and Kesidis, G. 2024. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE Symposium on Security and Privacy (SP)*, 1994–2012. IEEE.

Xu, J.; Huang, S.-L.; Song, L.; and Lan, T. 2022. Byzantine-robust federated learning through collaborative malicious gradient filtering. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 1223–1235. IEEE.

Yang, H.; Xi, W.; Shen, Y.; Wu, C.; and Zhao, J. 2024. RoseAgg: Robust Defense Against Targeted Collusion Attacks in Federated Learning. *IEEE Transactions on Information Forensics and Security*.

Yang, Z.; Chen, M.; Wong, K.-K.; Poor, H. V.; and Cui, S. 2022. Federated learning for 6G: Applications, challenges, and opportunities. *Engineering*, 8: 33–41.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, 5650–5659. Pmlr.

Zhang, H.; Yao, Z.; Zhang, L. Y.; Hu, S.; Chen, C.; Liew, A.; and Li, Z. 2023. 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*, 4567–4575.

Zhang, Z.; Cao, X.; Jia, J.; and Gong, N. Z. 2022. 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*, 2545–2555.