ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230393

FedPerturb: Covert Poisoning Attack on Federated Learning via Partial Perturbation

Tongsai Jin[1,2], **Zhihui Fu**[3], **Dan Meng**[3], **Jun Wang**[3], **Yue Qi** [3] **and Guitao Cao** [1,2;*]

[1]MoE Engineering Research Center of SW/HW Co-design Technology and Application, East China Normal University, Shanghai, China
[2]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[3]OPPO Research Institute

Abstract. Federated learning breaks through the barrier of data owners by allowing them to collaboratively train a federated machine learning model without compromising the privacy of their own data. However, Federation Learning also faces the threat of poisoning attacks, especially from the client model updates, which may impair the accuracy of the global model. To defend against the poisoning attacks, previous work aims to identify the malicious updates in high dimensional spaces. However, we find that the distances in high dimensional spaces cannot identify the changes in a small subset of dimensions, and the small changes may affect the global models severely. Based on this finding, we propose an untargeted poisoning attack under the federated learning setting via the partial perturbations on a small subset of the carefully selected model parameters, and present two attack object selection strategies. We experimentally demonstrate that the proposed attack scheme achieves high attack success rate on five state-of-the-art defense schemes. Furthermore, the proposed attack scheme remains effective at low malicious client ratios and still circumvents three defense schemes with a malicious client ratio as low as 2%.

1 Introduction

Federated learning(FL) [14, 25] is a distributed machine learning paradigm that allows multiple participants to collaboratively train a global machine learning model while maintaining data privacy. A typical FL setup consists of a central server and several distributed clients, each of which keeps its own training data locally and uploads only the intermediate results of training (usually local model updates) to the central server. The central server updates the global model based on the results uploaded by each client, and obtaining a better global model than training on local data only. Federated learning protects the data privacy at the cost of losing control of the training data, exposing it to several security threats, one of which is the poisoning attacks. A poisoning attack is the act of malicious participants corrupting or manipulating the global model by submitting tampered local model updates.

Early defense schemes mostly remove outliers from a statistical perspective, such as selecting the median value of the updates or remove the extreme value in the updates [26], or identifying the centroid of the distribution of the updates through the Euclidean Distance [3, 7]. However, as the adversary drives the parameter shifts of the global model and forms new centers of updates distribution by conspiring to attack [5, 2, 18], the statistical-based defense schemes are circumvented.

To distinguish such malicious model updates that are very close to normal updates statistically, researchers have proposed defenses such as the sign statistical value of the updates [24], singular value decomposition [18], cosine similarity [16, 4] or direct prediction of the next round of updates [27]. These elaborate descriptions provide servers with more robust identification of malicious model updates, and in the experimental results they provide, these defense schemes can achieve more than 90% identification of malicious model updates [24, 27] or almost eliminate the impact of poisoning attacks [16]. These defense schemes all assume that the malicious update needs to be sufficiently offset from the center of its high dimensional space to affect the global model. After evaluating the updates from their high dimensional space, most of them only constrain the ℓ_2-norm of the overall update while ignoring constraints on the magnitude of individual update values. For instance, Signguard [24] chooses to directly reject those updates with too large or too small ℓ_2-norm , FLTrust [4] and FLAME [16] chooses to clip those updates with too large ℓ_2-norm, while DnC [18] and FLDetector [27] do not limit the updates ℓ_2-norm.

In this paper, we design an untargeted poisoning attack scheme called the **FedPerturb**.We perform a complicit scaling attack on only a constrained small set of parameters from the model. We restrict the rate of change of ℓ_2-norm of the malicious update before and after the scaling of the parameters. Since most neural networks contain a huge number of parameters, the scaling of a fraction of the parameters does not significantly change their scales and directions in the high dimensional spaces. Meanwhile, most state-of-the-art defense schemes detect the malicious updates depending on the distances and the differences of the directions of the uploaded parameters as a whole. Thus, our attack could circumvent that kind of defense schemes by modifying the small subset of parameters while keeping the malicious updates close to the benign updates. As to what kind of parameters are more easily attacked than the others, we find that the batch normalization layers are extremely sensitive to our attack. For those models without the batch normalization layers, the bias from either the convolutional layers or the fully connected layers are prone to the proposed attack. Furthermore, our attack only requires a low ratio of malicious clients. For example, on ResNet18, our attack is still effective when only 2% malicious clients exist. In addition, we

* Corresponding Author. Email: gtcao@sei.ecnu.edu.cn.

could perform an effective untargeted poisoning attack without the knowledge of the aggregation algorithm or the training results of the benign participants, i.e. our attack is non-omniscient.

We summarize our contributions below:

- We propose **FedPerturb**, a novel untargeted poisoning attack scheme for federated learning, which scales a tiny fraction of update parameters to create malicious updates that circumvent multiple state-of-the-art defense schemes.
- We analyze the sensitivity of different parts in the neural networks to **FedPerturb** and propose two attack targets to enhance the effect of the attack.
- We extensively evaluate **FedPerturb** with multiple datasets and models. We show that our attack scheme can circumvent multiple state-of-the-art defense schemes. Meanwhile, our attack scheme does not require careful selection of the attack parameters to perform an effective attack.

2 Background and Related Work

FL is an emerging learning paradigm that allows data owners to collaboratively train a common machine learning model without sharing their private data. In this setup, the server (e.g. the service provider) broadcasts the jointly trained model (the global model) to the data owners (called clients), who receive the global model, train it using their local private datasets according to the methods specified by the server and upload updates to the server. The server aggregates the collected updates into a new global model using *aggregation rule* (AGR) and then broadcasts the global model to the clients for the next round of training.

2.1 Poisoning Attacks on FL

Federated learning protects the privacy of participant data while also making it vulnerable to various poisoning attacks [2, 20, 15, 8, 22, 5, 1, 17, 18]. Malicious clients may exist during the training process, which intentionally send false or malicious updates to other devices or servers in an attempt to corrupt or control the global model. We can classify these attacks according to the objectives of the adversary and the attack method.

Depending on the adversary's objectives, poisoning attacks can be divided into two types: targeted and untargeted poisoning attacks. In targeted poisoning attacks [2, 20, 22, 1, 15, 8], the adversary's objective is to minimize the accuracy of the global model for a specific class while maintaining accuracy for other classes. Backdoor attack is a special kind of targeted attack, where the objectives is to implant a specific trigger in the global model such that under specific input conditions, the global model outputs an error or a predetermined result. In the untargeted poisoning attack [2, 5, 18, 17, 23], the adversary's objectives is to minimize the accuracy of the global model across all classes of inputs. In order to perform powerful untargeted attacks and circumvent the AGR, adversaries [2, 5, 18, 23] develop optimized target functions to create malicious updates by directly modifying the training updates.

Depending on the adversary's attack method or capability, poisoning attacks can be divided into data and model poisoning attack.In a data poisoning attack [15, 8], the adversary manipulates the client-generated update indirectly by poisoning the training datasets. While in model poisoning attack [2, 20, 5, 1, 18, 17, 22, 23], the adversary can directly manipulate the client and is able to directly modify the client's training updates to perform more powerful poisoning attacks. Therefore, in order to explore the threat level of poisoning attacks on FL, we focus on studying untargeted model poisoning attacks in FL.

2.2 Existing Defense Strategies

In non-adversarial FL settings, dimension-wise average [14] is an effective AGR to aggregate clients' updates. However it has been proved that for the federated averaging algorithm, adversary can simply upload noise as training updates to reduce the accuracy of the global model, or even perform a model replacement attack via update scaling to implant a high-accuracy backdoor into the global model while controlling only 1% of the clients [1]. To combat the growing security threats, many different AGRs have been proposed in the literature [18, 24, 16, 4, 27, 10, 26, 3, 7] to identify malicious clients or improve the robustness of model training. Traditional defense schemes use statistics-based AGR to obtain reliable gradient estimates, such as Median [26], Trimmed-mean [26], multkrum [3] and Bulyan [7].

Recently, researchers aim to evaluate the updates from high dimensional spaces and identify the correct update direction of the global model. DnC [18] downsamples the updates and then calculates the outlier scores by the Singular Value Decomposition (SVD) algorithm. SignGuard [24] believes that the sign of the update plays a crucial role in the model update, and identifies the malicious update through the sign statistics of the update; FLTrust [4] and FLAME [16] both take the cosine similarity algorithm as the core, FLTrust takes the credible training result of the server as the basis, clipping or rejecting the update far from the credible update, FLAME finds the direction of the majority of updates by HDBSCAN clustering algorithm; Centeredclipping(CC) [10] finds the direction of the vast majority of updates by iterative clipping and manual momentum constraint to update the update of the update changes dramatically; Fldetector [27] based on historical information with LBFGS approximation to calculate the Hessian matrix of model updates and predict the next round of updates to the model, identify and rejects the updated update that deviates from the predicted direction. Moreover, there are some recent works [9, 13, 6] dedicated to mitigate the performance degradation caused by noisy labels. Their proposed schemes improve the robustness of federated learning and are able to mitigate the effects of poisoning attacks to some extent.

2.3 Threat Model

Here, we explore the objective and the capabilities possessed by the adversary in this paper

Adversary's objective. The adversary aims to create malicious updates and upload it to the central server during the training process, thereby compromising the accuracy of the aggregated global model for all types of test inputs. This type of attack is also known as untargeted poisoning attack.

Adversary's capabilities. We assume that the adversary controls up to m out of n total clients, and these clients are referred to as malicious or corrupted clients. We also assume that the number of malicious clients m is less than the number of benign clients $n - m$, i.e., the ratio of malicious clients to total clients (m/n) is less than 0.5. Following the previous works [2, 1, 18], We also assume that the adversary has access to the global model parameters broadcast in each epoch and can manipulate updates on the malicious clients. In addition, we assume that the adversary is non-omniscient, i.e., the adversary does not know the aggregation rules and does not know the training results of benign clients. Therefore, the adversary can

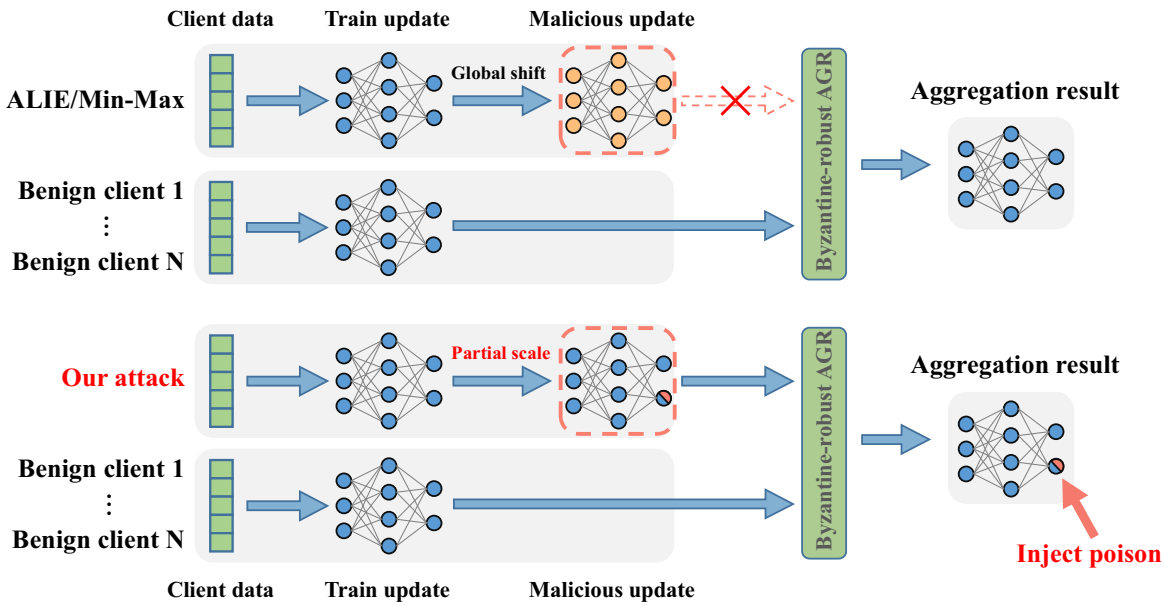

Figure 1: **Schematics of our attack scheme**: We change **only a small portion** of the updates(red area), the rest of the updates are the result of benign training while the ALIE[2]/Min-Max[18] attack drives global parameter shifts.

only use the training results of malicious clients to create malicious updates.

3 Our FedPerturb Framwork

The state-of-the-art FL defense schemes [18, 24, 16, 27, 4] have chosen to find a higher dimensional space to evaluate the updates, e.g. to find the benign updates that occupy the absolute majority. However, we find that these schemes neglect to detect the local anomalies of updates. Observing this, we propose a new untargeted poisoning attack called FedPerturb.

Intuitively, in order to escape detection by state-of-the-art robust AGRs, we scale only a very small fraction (i.e., P_{ind}) of the update (see Figure 1), rather than changing every value in the update. In the case of the ResNet18, we change less than 0.5% of the parameters. By continuously scaling the parameters of the target range P_{ind} significantly, the values of the P_{ind} regions parameters in the global model grow abnormally large, eventually leading to global model divergence.

3.1 Exploited Assumption

We find that all three current state-of-the-art attack schemes [2, 18, 5] have a similar approach, that is, in order to minimize the affection of the training accuracy, these schemes choose to widen the gap between the overall parameters of the model before and after poisoning when crafting the malicious updates. These malicious updates deviate as much as possible from the aggregation center under well-designed constraints, posing a significant challenge to statistics-based defense schemes [26, 3].

To counter these state-of-the-art attack methods that focus on achieving global parameter shifting, researchers have shifted the perspective of the defense schemes from local to global, and evaluated the updates from higher dimensional spaces. However, these defense schemes only consider the update as a whole and thus impose constraints, neglecting to detect and constrain the magnitude of the concrete values in the update. As the number of model parameters grows, changes to a very small range of parameters locally do not significantly change their appearance in the high dimensional spaces.

Compared to the change in update sign of the ALIE attack (>20%), we only change the update sign by less than 0.5%, and we experimentally demonstrate that Signgurad is unable to detect such a small difference. For both FLAME and FLTrusth, two cosine similarity-based defense schemes, the malicious updates generated by our scheme does not significantly change their vector direction (i.e., the cosine similarity between the malicious updates and the benign updates does not change significantly). The DnC only samples 1000 pieces of data and was not able to effectively detect a small range of parameter anomalies. Leveraging this vulnerability, we could circumvent these state-of-the-art defense schemes and cause global model divergence by scaling a very small range of parameters.

3.2 Our Propose Attack

Based on the above observations, we choose to scale only a small fraction of the updates, rather than trying to create malicious updates that would cause parameter shifts in the global model. In this case, the main question that needs to be answered is which parameters are more effective for us to attack. During training, the network updates parameters layer by layer from the output layer to the input layer according to the backpropagation algorithm. This allows parameters close to the output to influence a wider range of model parameters during training. Therefore, we choose to scale the parameters close to the output. Specifically, we have two strategies for selecting the target parameters of attack depending on whether the model contains batch normalization(BN) layers or not. For models containing BN layers, we select the weights of the convolutional layers close to the output as the target parameters of attack, while for models not containing BN layers, we select the bias of the operator close to the output or the bias of the last fully connected layer. We will discuss these two strategies in section 3.3 and 3.4.

Basic attack process.In the training process of each epoch of FL, we will first control our own malicious clients to perform the benign training tasks and obtain the base data for creating malicious updates,

Algorithm 1 FedPerturb Attack

Require: $\{g_i : i \in m\}$, P_{ind}, S
1: $\mu_{P_{ind}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} g_i[P_{ind}]$
2: $N_T \leftarrow \| \mu_{P_{ind}} \|$
3: **for** $j \in m$ **do**
4: $N \leftarrow \| p_j \|$
5: $N_R^2 \leftarrow N^2 - N_T^2$
6: $S_{dyn} \leftarrow \sqrt{\frac{(S\cdot N)^2 - N_R^2}{N_T^2}}$
7: $g_j[P_{ind}] \leftarrow -S_{dyn} \cdot \mu_{P_{ind}}$
8: **end for**

since we do not have the training results of the benign clients. Before uploading the updates, we collect the benign training update g of all malicious clients, calculates the mean $\mu_{P_{ind}}$ of the poisoning range P_{ind} in the updates and its ℓ_2-norm N_T. To ensure that the ℓ_2-norm of the update after scaling is not excessive, we control the magnitude of the change in update ℓ_2-norm after scaling by a hyperparameter S and calculate the target scaling factor S_{dyn} by lines 4 to 6 in Algorithm 1. The corresponding values in the updates of that malicious client are replaced with the back-scaled average (i.e., $-S_{dyn}\cdot\mu_{P_{ind}}$), and then the generated malicious update is finally uploaded. In this process, we only modify the updates in the P_{ind} region, and the rest of the updates are the result obtained from benign training.

After we determine the target parameters to be poisoned, we also need to determine how many parameters in that layer to poison (i.e., the poisoning ratio P_R) and the S. These parameters serve to constrain the variation of the malicious updates in the high dimensional spaces. Since we focuses on non-omniscient untargeted poisoning attacks, we experimentally demonstrate the effectiveness of our attack scheme, i.e., the ability to implement powerful untargeted poisoning attacks without careful selection of these two parameters. The experimental results of which are detailed in 5.5.

3.3 Attacks on Biases

The mathematical expressions of a convolutional layer can be expressed in the following form:

$$y = f(Wx + b) \tag{1}$$

where y is the output of the layer, x is the input, W is the weight matrix, b is the bias vector and f is the activation function. While a neural network typically has many fewer biases than weights, for those networks without the batch normalization layers, we can use the biases as the attack targets to perturb the model output directly.

In order to directly affect the output of the model while leveraging the backpropagation algorithm to amplify the impact of poisoning attack, we choose the bias vectors near the output layer as the target of our attack, thus affecting the model output more directly. In the case of convolutional neural networks for classification tasks, we choose to target the bias of the last fully connected layer, which is usually the same number as the number of classification categories. In other words, we attack the global model with very few parameters.

3.4 Attack on Weights that Before the BN Layers

The target scaling factor S_{dyn} in our attack scheme depends on the ℓ_2-norm of updates before and after scaling. With the scaling ratio S constant, the smaller the ratio of N_T to N, the larger S_{dyn} will be. We find that when a model contains BN layers, it is easier to achieve large scaling factor S_{dyn} by attacking the weights of the convolutional layers that before the BN layers.

Since the parameters running mean(μ) and running variance(δ^2) in the BN layers do not have gradients, the client needs to change the training update from parameter gradients to parameter difference(i.e., $g_i^j \leftarrow para_i^j - para_i^{j-1} : j \in GlobalRound$), in order to aggregate these two parameters. This change does not significantly change the amount of data, but drastically increases the ℓ_2-norm of the update. Taking the ResNet18 as an example, the ℓ_2-norm of the first training round update containing these two types of unlearnable parameters is around 400, while that does not contain them is only around 20. In this case, setting the scaling ratio S to 1.5 means that scaling the target range parameters would increase the training update ℓ_2-norm from about 400 to 600, which also means that the target scaling factor S_{dyn} becomes very large. In our experiments we find that this variation of the magnitude ℓ_2-norm does not deviate from the range of variation of the normal update ℓ_2-norm. While there is a probability that it is clipped by the defense scheme, this clipping ratio is between 0.25 to 1, and the S_{dyn} after clipping remains huge.

The Catastrophe of Variance. According to the formulae of $\mu : \mu \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$ and $\delta^2 : \delta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$, when we enlarge a small part of x by a factor of S_{dyn}, δ^2 is approximately scaled by a factor of S_{dyn}^2, and thus the next round of training will produce a greater δ^2 update, which further increases the scaling factor S_{dyn}, forming a positive feedback. Finally the model training diverges as the absolute value of the scaling attack target parameters in the global model are abnormally huge.

4 Evaluation Setup

4.1 Datasets and Models

Fashion-MNIST. Fashion-MNIST [21] is a 10-class clothing image classification dataset, which consists of 60,000 training samples and 10,000 test samples. Each sample is a 28 × 28 greyscale image, and each class of Fashion-MNIST has 7,000 images. To compare the impact of BN layers, we use two convolutional neural network models as global model architectures: CNN consists of two convolutional layers and two fully connected layers, and CNN-BN consists of three convolutional layers, three BN layers and one fully connected layer.

CIFAR10. CIFAR10 [11] is a 10-class color image classification task with 60,000 32 × 32 RGB images, including 50,000 training samples and 10,000 test samples, each class of CIFAR10 has 6,000 images. We use four models as the global model architecture: the two models containing the BN layer are ResNet18 and VGG11-BN [19], and the two models without the BN layer are AlexNet [12] and VGG11 [19].

4.2 Baseline Model Poisoning Attacks

We consider various popular model poisoning attack schemes and we only discuss non-omniscient attack settings, where the model attack schemes are:

Label-Flip(LF). In order to generate malicious gradients, the malicious clients flip the local sample labels from l to $C-1-l$, where C is the total label categories and $l \in \{0, 1, ..., C-1\}$.

Inner Product Manipulation(IPM). Following the approach in [10], in each round the adversary first estimate coordinate-wise mean (μ_j), and calculate malicious gradient by $(p_{mal})_j \leftarrow -\epsilon \cdot \mu_j$. We set the $\epsilon = 0.5$.

A Little is Enough(ALIE). Following the algorithm 3 in [2], we calculate the z^max by line 1 and line 2 in the algorithm, that is $s \leftarrow$

Table 1: Comparing **our attack on weights** and state-of-the-art model poisoning attacks reports test accuracy, where **D** indicates that the test loss of the global model is greater than 1000 or becomes NaN, making the global model diverge before the end of training.

Dataset (Model)	AGR	No Attack	LF	IPM	ALIE	Min-Max	Min-Sum	**Ours**
CIFAR10 (ResNet18)	Mean	87.86	86.42	86.23	85.69	83.01	83.98	**16.41(D)**
	Median	87.41	82.66	76.59	83.76	72.49	**71.88**	87.67
	Multikrum	87.33	87.27	80.58	86.11	79.98	81.37	**32.00(D)**
	Signguard	86.83	86.51	86.14	87.56	84.25	83.35	**10.00(D)**
	DnC	88.03	86.13	85.37	86.92	81.38	82.32	**10.14(D)**
	CC	88.06	84.62	85.75	86.95	82.81	83.67	**10.00(D)**
	FLAME	86.60	86.43	84.24	86.13	86.50	84.44	**34.39(D)**
	FLTrust	87.43	86.11	86.92	84.89	86.40	85.99	**10.00(D)**
CIFAR10 (VGG11-BN)	Mean	85.21	81.45	82.26	83.10	75.91	78.50	**10.34(D)**
	Median	84.47	83.65	63.31	**58.18**	62.87	63.32	84.70
	Multikrum	84.87	83.18	77.23	79.14	**69.54**	70.98	84.43
	Signguard	84.47	83.97	81.34	84.63	74.98	77.99	**16.33(D)**
	DnC	85.02	83.52	81.04	83.13	74.33	76.87	**19.86(D)**
	CC	85.22	82.23	81.62	83.19	76.07	78.40	**10.00(D)**
	FLAME	83.36	82.60	83.82	81.66	84.54	83.71	**28.27(D)**
	FLTrust(lr=0.1)	84.16	**10.00(D)**	**83.95(D)**	**10.54(D)**	83.82	84.44	**10.00(D)**
	FLTrust(lr=0.01)	83.21	82.6	82.73	80.77	82.65	82.67	**10.29(D)**
Fashion MNIST (CNN-BN)	Mean	90.28	88.23	90.04	90.29	89.75	89.72	**61.85(D)**
	Median	90.27	89.06	88.24	89.05	**88.20**	88.38	90.17
	Multikrum	89.56	89.30	89.45	90.15	**88.89(D)**	89.19	90.26
	Signguard	90.23	90.37	89.92	90.59	89.30	89.94	**51.62(D)**
	DnC	90.35	87.59	90.08	90.15	89.48	89.77	**62.07(D)**
	CC	90.34	84.85	89.95	90.31	89.47	89.74	**56.54(D)**
	FLAME	90.17	89.93	90.38	90.25	90.27	90.07	90.08
	FLTrust(lr=0.01)	**89.90(D)**	**86.89(D)**	**90.10(D)**	**86.26(D)**	90.44	**87.92(D)**	**85.43(D)**
	FLTrust(lr=0.001)	91.07	**90.32**	90.47	91.07	90.63	90.73	90.97

$\lfloor \frac{n}{2} + 1 \rfloor - m$ and $z^{max} \leftarrow max_z(\phi(z) < \frac{n-m-s}{n-m})$, in each round the adversary first estimate coordinate-wise mean (μ_j) and standard deviation (δ_j), and calculate malicious gradient by $(p_{mal})_j \leftarrow \mu_j - z^{max} \cdot \delta_j$.

Min-Max and Min-Sum. We use the source code provided in [18] to achieve the Min-Max and Min-Sum attack, whose version of the attacks is AGR agnostic and gradients of benign clients are unknown.

4.3 Training and Attack Settings

Unless specified otherwise, we consider a FL setup with a total of 50 participants, 20% of which are malicious participants (i.e. $n = 50, m = 10$). In both CIFAR10 and Fashion-MNIST datasets, the training data are independently and identically distributed (IID) among clients, because poisoning FL with IID data is the hardest [5]. To verify the effectiveness of our attack, we also evaluate the impact of different fractions of malicious clients on our attack scheme against different defense schemes. For reproducibility, our code is available at https://github.com/running-sheep/FedPerturb.

In the four test models in CIFAR10, we use the same training setup, we set the batch size to 64 and use the SGD optimizer with learning rate of 0.1 for a total of 100 training rounds.The number of local rounds is set to 2 and the weight decay is set to 0.0005, while momentum is employed with the parameter of 0.9. In the two test models in Fashion-MNIST, we set the batch size to 64 and use the Adam optimizer with learning rate of 0.01 for a total of 60 training rounds.The number of local rounds is set to 2 and the weight decay is set to 0.0005.

In the setting of attack FedPerturb Attack, for the three models without BN layers, Alexnet, VGG11 and CNN, we take the 10 biases of the last fully connected layer (i.e. the poisoning ratio $P_R = 1$) as the poisoning object P_{ind}, and the S is set to 1.5. For the three models with BN layers, ResNet18, VGG11-BN and CNN-BN, the S is set to 1.5, and set the weights of the convolutional layers as the poisoning object ϕ, and in each round, we randomly selects a convolutional layer in ϕ, calculates the ℓ_2-norm of each convolutional kernel, and selects the largest 2% kernel (i.e., the poisoning ratio $P_R = 0.02$) as the poisoning object P_{ind}. Specifically, we use the last four convolutional layers as the poisoning object of ResNet18, and set the the last two convolutional layers as the poisoning object of VGG11-BN and CNN-BN.

5 Evaluation Results

In this section, we conduct extensive experiments with various attack-defense pairs on IID data setting. We compare our methods with the following defense methods, including Median [26], Multi-Krum [3], Signguard [24], DnC [18], Centeredclipping(CC) [10], FLAME [16] and FLTrust [4]. For the experiments in the non-IID data setting, the results of the experiments can be found in our source code.

5.1 Results of Attacks on Weights that Before the BN Layers

For the model containing the BN layer, the results of our attack with the state-of-the-art poisoning attack under different datasets and different defense schemes are shown in the Table 1, the values in the table represent the global model test accuracy rates.

For the CIFAR10 dataset with ResNet18 and VGG11-BN, our attack circumvents all the five state-of-the-art defense schemes and achieves the best attack effect. These state-of-the-art defensive schemes basically provide better defense than the two traditional defensive schemes except for our attack. Since we do not create global offset malicious updates like the previous attacks, our attack has little effect on the median. For Multkrum, however, it is unable to remove all of the malicious gradients, so we still successfully attacked Multkrum in ResNet18. We find that FLTrust does not perform consistently in VGG11-BN, and in many cases its training even diverged. Considering that there are constraints on the learning rate of training in FLTrust, we retest the FLTrust scheme at a learning rate of 0.01 and show that FLTrust is better able to defend against other attacks at

Table 2: Comparing **our attack on biases** and state-of-the-art model poisoning attacks reports test accuracy, where **D** indicates that the test loss of the global model is greater than 1000 or becomes NaN, making the global model diverge before the end of training.

Dataset (Model)	AGR	No Attack	LF	IPM	ALIE	Min-Max	Min-Sum	**Ours**
CIFAR10 (AlexNet)	Mean	69.39	67.20	65.46	67.80	59.11	60.94	**37.26(D)**
	Median	68.31	66.46	**50.74**	62.12	51.42	50.79	68.12
	Multikrum	68.01	67.93	60.84	66.1	**48.72**	53.24	67.98
	Signguard	68.92	68.83	67.39	68.51	65.44	67.61	**47.25**
	DnC	69.01	67.53	63.98	67.38	56.41	57.67	**34.16(D)**
	CC	69.59	67.15	65.55	67.77	59.00	61.57	**11.35**
	FLAME	68.98	**60.18**	68.27	66.17	68.46	67.88	66.67
	FLTrust	69.09	67.04	68.98	**64.08**	69.21	68.20	67.69
CIFAR10 (VGG11)	Mean	80.44	76.59	75.32	77.89	63.84	68.10	**47.45(D)**
	Median	78.50	73.77	**44.39**	65.78	49.05	47.64	78.79
	Multikrum	78.72	78.51	67.24	74.44	**43.11**	48.22	79.60
	Signguard	80.47	80.24	76.80	79.42	73.37	70.93	**70.44**
	DnC	77.73	65.53	74.27	76.41	58.40	64.26	**9.33**
	CC	79.96	75.81	75.60	78.74	63.14	68.34	**14.02**
	FLAME	74.81	**68.18**	78.26	73.44	78.47	78.19	78.03
	FLTrust	80.11	**72.82**	79.84	76.44	80.48	79.35	79.89
Fashion MNIST (CNN)	Mean	84.79	82.30	84.00	84.04	84.01	84.29	**81.04**
	Median	85.10	84.48	**82.51**	82.56	83.46	83.72	85.27
	Multikrum	84.34	84.35	82.35	82.12	**80.19**	81.60	84.51
	Signguard	84.51	82.66	84.62	84.68	84.43	84.49	**81.94**
	DnC	84.28	83.40	84.66	84.88	83.84	83.81	**80.85**
	CC	85.07	82.53	84.82	84.08	84.42	84.34	**79.87**
	FLAME	84.36	84.32	84.91	**82.84**	84.91	84.91	84.37
	FLTrust	84.45	82.72	84.70	82.98	84.70	84.70	**82.84**

that learning rate, yet our attack still circumvents FLTrust and causes the global model to diverge.

For the Fashion-MNIST dataset with CNN-BN, our attack circumvents the three state-of-the-art defense schemes and causes the global model to diverge. In the attack on FLAME and FLTrust, our attack has a large impact on the cosine similarity due to the small number of parameters of the CNN-BN, which is only 37K (compared to 2.7M for the ResNet18), leading to our attack being defended. We carry out further comparative experiments on learning rate in Section 5.5.

5.2 Results of Attacks on Biases

For the model without the BN layer, the results of our attack with the state-of-the-art poisoning attack under different datasets and different defense schemes are shown in the Table 2, the values in the table represent the global model test accuracy rates.

For the CIFAR10 dataset, in the experiments with AlexNet as the global model, our attack achieves the best attack results in three state-of-the-art defense schemes. For the two cosine similarity-based defense schemes, FLAME and FLTrust, our attack has a high impact on the cosine similarity between malicious and normal updates, and the attack is ineffective. In the experiments with VGG11 as the global model, our attack scheme also achieves the best results in the experiments with the three defense schemes state-of-the-art Signguard, DnC and CC.

For the Fashion-MNIST dataset, in the experiments with CNN as the global model, our attack could not cause global model divergence because there is no batch normalisation layer in the model, and the number of model parameters is also smaller than in the first two models, so our attack could not be scaled to a larger ratio (scaling multiplier within 10), but compared to other attack schemes, our attack still achieves the most effective attack in four state-of-the-art defense schemes.

5.3 How Our Attack Works

To illustrate how our attack causes the global model to diverge, we tracked the changes in the ℓ_2-norm of each layer's parameters in the model during the training process. Figure 2 shows the results when we attack ResNet18 and AlexNet on the CIFAR10 dataset.

During the training of non-malicious clients, some parameters in the ResNet18 have significantly higher ℓ_2-norms than other parameters (shown as spikes in Figure 2), which are mainly the running variance parameters of the BN layer. After a few rounds of training, the running variance ℓ_2-norms of the last four BN layers are still higher than other parameters. When we attack ResNet18, the running variance ℓ_2-norm in the attacked area increases sharply, and the model update ℓ_2-norm also increases accordingly, which enables our attack to scale at a larger proportion in the next round of attack. The ℓ_2-norm of the attacked parameters (the convolutional layer weights for ResNet18) increases by nearly 20 times (from 8.43 to 160.06) after the attack, and the maximum change ratio of the running variance ℓ_2-norm of the corresponding BN layer approaches 100 (from 33.91 to 3389.15), which causes the model to diverge due to the abnormal increase of the model parameters.

Moreover, our attack process for ResNet18 is very fast, and the global models under the other four defense schemes diverge within 3 epochs, while the CC defense scheme diverges at the 10th epoch due to the limitation on the size of each round of model update. For AlexNet, the bias of the fully connected layer that is attacked also increases continuously under continuous attack, resulting in the global model diverging or becoming unusable, but because each layer's parameters have smaller ℓ_2-norms, and their updates also have smaller ℓ_2-norms, our attack requires more rounds of training to take effect.

5.4 Effect of the Percentage of Malicious Client

Table 3 shows the effectiveness of our attack on the three global models with five state-of-the-art defense schemes at different malicious client ratios. ResNet18 has more BN layers than the VGG11-BN and CNN-BN models, which makes the ℓ_2-norm change of the model update more drastic after the attack and thus more likely to cause the global model to diverge.

For ResNet18, our attack causes the global model to diverge under the three state-of-the-art defense schemes of Signguard, DnC and CC at only 2% of malicious clients (when there is only one mali-

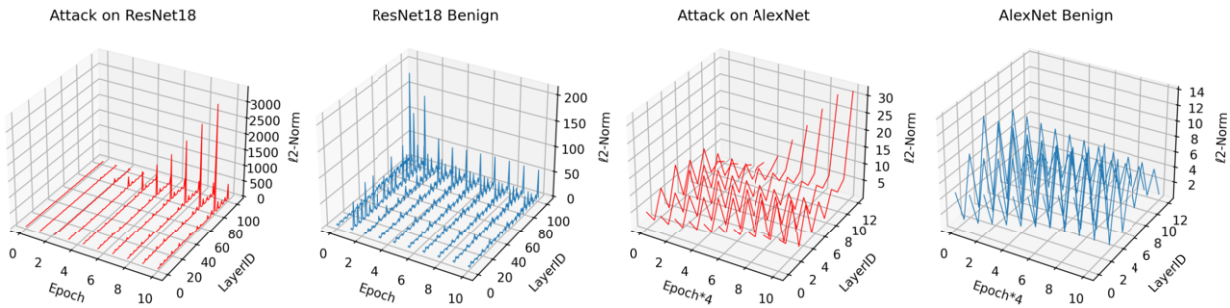

Figure 2: The ℓ_2-norm variation of the model parameters per layer with training progress for ResNet18 versus AlexNet under our attack(red line) and without attack(blue line).

Table 3: Accuracy of the global model under our attack with different malicious client percentages, where **D** indicates that the test loss of the global model is greater than 1000 or becomes NaN, making the global model diverge before the end of training.

Dataset (Model)	Ratio	Mean	Signguard	DnC	CC	FLAME	FLTrust
CIFAR10 (ResNet18)	2%	**12.96(D)**	**19.98(D)**	**60.46(D)**	**38.30(D)**	87.08	86.46
	4%	**10.00(D)**	**10.00(D)**	**10.60(D)**	**10.50(D)**	87.16	**11.44(D)**
	10%	**10.00(D)**	**10.00(D)**	**11.15(D)**	**10.00(D)**	86.89	**10.00(D)**
	20%	**16.41(D)**	**10.00(D)**	**10.14(D)**	**10.00(D)**	**34.39(D)**	**10.00(D)**
CIFAR10 (VGG11-BN)	2%	85.04	84.49	85.11	85.08	83.55	84.90
	4%	**31.84(D)**	83.78	**31.32(D)**	84.49	84.01	84.38
	10%	**10.70(D)**	**10.00(D)**	**10.00(D)**	**10.00(D)**	83.50	**20.69(D)**
	20%	**10.34(D)**	**16.33(D)**	**19.86(D)**	**10.00(D)**	**28.27(D)**	**10.00(D)**
Fashion MNIST (CNN-BN)	2%	90.27	90.50	90.54	90.41	89.99	90.28
	4%	90.39	89.92	90.26	90.15	90.10	**90.25(D)**
	10%	**87.47(D)**	90.03	**86.80(D)**	90.10	90.21	90.09
	20%	**61.85(D)**	**51.62(D)**	**62.07(D)**	**56.54(D)**	90.08	**85.43(D)**

cious client). The minimum percentage of malicious clients causing the global model to diverge is 4% for the VGG11-BN model and further increases to 10% for the CNN-BN model. The experimental results show that the impact of our attack on the model is not linear with the proportion of malicious parties, but segmented. That is, there is a critical value at which our attack causes the global model to diverge when the malicious client fraction is larger than this value, while the impact on the global model is weaker (less than 5%) when that is smaller than this value. Since we focus on non-omniscient untargeted poisoning attacks, we do not calculate the corresponding attack critical value for each defense scheme.

5.5 Impact of Attack Parameters and Learning Rate

Our attack scheme is a heuristic scheme and requires three hyperparameters to be set manually when performing the attack, i.e. which kinds (weights, biases or other parameters) of parameters to be attacked, the poisoning ratio P_R, and the scaling factor S. To better show the severity of the deficiencies we find in state-of-the-art defense schemes, we compare the attack effectiveness of our attack scheme under different attack hyperparameters and different learning rates, with ResNet18 as the global model.

Attack parameters. We experiment with five state-of-the-art defense schemes for FedPerturb with different attack parameters, where the poisoning ratio P_R is chosen from 0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2 and 0.5, and the choices of the scaling factor S were 1.1, 1.3, 1.7, 1.9, 2.5, 3 and 5. The other parameters remained the same as those set in section 4.3. Experimental results show that our attack is defended by FLAME when the scaling multiplier $S \geq 2.5$ and by Signguard when $P_R = 0.5$ and $S = 5$. In all other cases our attack circumvents the defense schemes and causes global model divergence within 20 epochs.

Learning rate. We conduct attack experiments on five state-of-the-art defense schemes at different learning rates, where the learning rates were chosen to be 0.1, 0.05, 0.01, 0.005, 0.001, and 0.0005, while the other parameters remained the same as those set in section 4.3. The experimental results show that all four defense schemes are unable to defend against our attack at all learning rates, i.e. they are unable to prevent global model divergence, except for the FLAME scheme, which is able to prevent model divergence at a learning rate of 0.01 or smaller.

With these two experiments, we demonstrate that our attack scheme does not require careful selection of attack parameters to circumvent the defense schemes and cause global model divergence when attacking ResNet18.

6 Conclusion

We find a weakness that is prevalent in current state-of-the-art defense schemes and propose a new attack scheme based on this vulnerability that scales only a small fraction of the parameters, rather than aiming to drive global parameter shifts like previous works. In the attack on ResNet18, our attack scheme circumvents five state-of-the-art defense schemes and causes global model divergence, whereas our attack scheme still circumvents three state-of-the-art defense schemes and causes global model divergence when controlling only 2% of the clients. We give specific reasons why the attack is effective, and future Byzantine robust FL algorithms should address these issues.

7 Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 61871186 and 61771322.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, 'How to backdoor federated learning', in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, (2020).

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg, 'A little is enough: Circumventing defenses for distributed learning', *Advances in Neural Information Processing Systems*, **32**, (2019).

[3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer, 'Machine learning with adversaries: Byzantine tolerant gradient descent', *Advances in neural information processing systems*, **30**, (2017).

[4] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong, 'Fltrust: Byzantine-robust federated learning via trust bootstrapping', in *Network and Distributed System Security Symposium(NDSS)*, (2021).

[5] 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*, pp. 1623–1640, (2020).

[6] Xiuwen Fang and Mang Ye, 'Robust federated learning with noisy and heterogeneous clients', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10072–10081, (2022).

[7] Rachid Guerraoui, Sébastien Rouault, et al., 'The hidden vulnerability of distributed learning in byzantium', in *International Conference on Machine Learning(ICLR)*, pp. 3521–3530. PMLR, (2018).

[8] 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)*, pp. 19–35. IEEE, (2018).

[9] Xuefeng Jiang, Sheng Sun, Yuwei Wang, and Min Liu, 'Towards federated learning against noisy labels via local self-regularization', in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 862–873, (2022).

[10] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi, 'Learning from history for byzantine robust optimization', in *International Conference on Machine Learning*, pp. 5311–5319. PMLR, (2021).

[11] Alex Krizhevsky, Geoffrey Hinton, et al., 'Learning multiple layers of features from tiny images', (2009).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', *Communications of the ACM*, **60**(6), 84–90, (2017).

[13] Junyi Li, Jian Pei, and Heng Huang, 'Communication-efficient robust federated learning with noisy labels', in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 914–924, (2022).

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, 'Communication-efficient learning of deep networks from decentralized data', in *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, (2017).

[15] 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*, pp. 27–38, (2017).

[16] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al., '{FLAME}: Taming backdoors in federated learning', in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, (2022).

[17] Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos, 'Detox: A redundancy-based framework for faster and more robust gradient aggregation', *Advances in Neural Information Processing Systems*, **32**, (2019).

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

[19] Karen Simonyan and Andrew Zisserman, 'Very deep convolutional networks for large-scale image recognition', in *International Conference on Machine Learning(ICLR)*, (2015).

[20] 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', *Advances in Neural Information Processing Systems*, **33**, 16070–16084, (2020).

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf, 'Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms', *arXiv preprint arXiv:1708.07747*, (2017).

[22] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li, 'Dba: Distributed backdoor attacks against federated learning', in *International Conference on Learning Representations*, (2020).

[23] 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*, volume 115, pp. 261–270. PMLR, (22–25 Jul 2020).

[24] Jian Xu, Shao-Lun Huang, Linqi Song, and Tian Lan, 'Byzantine-robust federated learning through collaborative malicious gradient filtering', in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 1223–1235. IEEE, (2022).

[25] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong, 'Federated machine learning: Concept and applications', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **10**(2), 1–19, (2019).

[26] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett, 'Byzantine-robust distributed learning: Towards optimal statistical rates', in *International Conference on Machine Learning(ICLR)*, pp. 5650–5659. PMLR, (2018).

[27] Zaixi Zhang, Xiaoyu Cao, Jinyuan 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*, pp. 2545–2555, (2022).