Advanced Free-rider Attacks in Federated Learning

Zhenqian Zhu
School of Computer Science and Technology
Harbin Institute of Technology
Shenzhen 518000,China
cszhuzhenqian@gmail.com

Jiangang Shu
Cyberspace Security Research Center
Peng Cheng Laboratory
Shenzhen 518000,China
shujg@pcl.ac.cn

Xing Zou
Cyberspace Security Research Center
Peng Cheng Laboratory
Shenzhen 518000,China
zoux02@pcl.ac.cn

Xiaohua Jia *
Department of Computer Science
City University of Hong Kong
Kowloon Tong, Hong Kong 999077, China
csjia@cityu.edu.hk

Abstract

Federated learning is a new machine learning technology that multiple clients collaboratively to train a global model without sharing their local data. Due to the fact that clients have the direct control over their local models and training data, federated learning is inherently vulnerable to free-rider attacks that the malicious client forges local model parameters to get reward without contributing sufficient local data and computation resources. Recently, many different free-rider attacks have been proposed. However, existing attacks haven't a good stealth property. The convergence property represents the convergent speed and final global model accuracy. The stealth property indicates the attacker's ability to hide its local update. In this work, we first utilize the Ornstein-Uhlenbeck (OU) process to formalize the evolution of local and global training processes, and analyze the geometrical relationship of all clients' local model updates. Then, we propose a scaled delta attack and an advanced free-rider attack. We also prove that advanced free-rider attack can not only ensure the convergence of the aggregated model, but also hold the stealth property. Expriment results demonstrate that our advanced free-rider attack is feasible and can escape from state-of-the-art defense mechanisms. Our results show that even a highly constrained adversary can carry out the advanced free-rider attack while simultaneously maintaining stealth under the defense strategies, which highlights the vulnerability of the federated learning setting and the need to develop effective defense strategies.

1 Introduction

Federated learning (FL) is a popular implementation of distributed stochastic optimization for large-scale deep neural network training [1, 2, 3]. It is a multi-round strategy where multiple clients work together to train a model under the orchestration of a central server, while preserving the confidentiality of the local training data. To ensure clients privacy, federated learning is designed to have no visibility into clients local data and training processes. FL systems can preserve data privacy and reduce the costs resulting from traditional centralized machine learning frameworks. Nowadays, more and more federated learning frameworks have been developed and deployed, such as TensorFlow Federated Framework and Webank's FATE.

*

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

In recent years, the security of FL has received significant interest from both research communities and the industry. There has been intensive work focusing on attacks and defenses for various attack vectors, such as evasion attack, data poisoning attack and model poisoning attacks [3]. In evasion attacks, the adversary alters the data used at inference-time. In data poisoning attacks, an adversary mainly injects malicious data into the training dataset before the learning process starts, while the learning process is assumed to maintain integrity [4, 5, 6]. In model poisoning attacks [7, 8], an adversary exploits the fact that a client can directly manipulate local model update sent back to the server. Free-rider attacks [8, 9] a new special attack method, where free-riders construct local updates through the sharing of opportune counterfeited parameters. The free-riders may submit fake parameters due to several reasons: 1) the adversary does not have any local data for training; 2) the adversary wants to save local storage spaces or computation resources [10].

In this paper, we propose advanced free-rider attack where the adversary can carefully construct local model update without being detected. We first formulate the local and global training processes of FL with OU process. Then, we propose our scaled delta attack that holds the convergence property for the aggregated global model but lacks a good stealth property. After that, we analyze the geometrical relationship of all clients' local model updates. Based on our theoretical analysis, our proposed advanced free-rider attack with a carefully constructed noise can blur the difference between free-riders' local updates and other honest updates. We also conduct experiments to demonstrate the feasibility of our attack.

Our contributions. We analyze the geometrical relationship of all local clients' model updates in an iid setting, that is, the evolution trend of cosine similarity for local model update, and propose an advanced free-rider attack where the free-rider can carefully construct local model update. The forged update has the convergence property and a good stealth property.

2 Related work

Defense strategies. Existing defense methods against training-time attacks can be summarized as follows: 1) Byzantine-tolerant aggregation mechanism. It is an alternative aggregation mechanism to ensure model convergence in the presence of Byzantine participants. Traditional techniques for secure aggregation include Krum [11], GeoMed [12] and Trimmed Mean [13]. 2) Norm difference clipping. The central sever in FL system can checks the *l2*-norm of each local model update returned by all clients [14]. And it clips the model updates that exceed a norm threshold. 3) Differential Privacy (DP). In recent research [15, 14], the central server can add a Gaussian noise with small standard deviations to the global model. In addition, participant-level differential privacy can reduce the effectiveness of the backdoor attack, but only at the cost of degrading the models performance on its main task. 4) Cosine Similarity. Another defense [16] targets sybil attacks by measuring the cosine similarity across the local updates and discards those that are very similar to each other. The variant version of this method is to compute the pairwise cosine similarity between all participants updates to distinguish the attackers local update. 5) Anomaly detection. The center server can detect local model updates submitted by all clients through anomaly detection methods and discards the outliers. Some cluster-based anomaly detection on multi- or high-dimensional data such as Autoencoder [17, 18], Deep Autoencoding Gaussian Mixture Model (DAGMM)[19] are often utilized.

3 Problem formulation

3.1 Federated learning

In a FL system, there is a central server which distributes its global model to all clients, and aggregates the local updates to generate a new global model at each round. We consider a typical SGD-based federated setting which consists of a central server and n clients $C_1, C_2, \cdots, C_n$, each with N_i local private training samples. The total number of training samples is N. At each round $t (t = 0, 1, 2...)$, the server distributes the current global model $\theta(t)$ to all clients. Each participant C_i trains a new local model based on global model $\theta(t)$, and then sends back the local update $U_i(\theta)$ to the server. Later, the server aggregates all local updates as follows:

$$\theta(t+1) = \theta(t) + \sum_{i=1}^{n} \frac{N_i}{N} U_i(\theta). \tag{1}$$

3.2 Formalizing FL training process

The Ornstein-Uhlenbeck process definition. In FL systems, SGD algorithm [20] is an important algorithm, which enables efficient optimization by following noisy gradients with a decreasing step size. In this work, we follow the typical FL setting equipped with SGD algorithm. According to recent researches [21, 22, 23, 24] SGD can be generally modeled as an OU process. The OU process is a stationary Gauss-Markov process where drifts towards its mean function over time. Specifically, the OU process can be described by the following stochastic differential equation:

$$d\theta_t = \lambda(\mu - \theta_t)dt + \sigma dW_t, \tag{2}$$

where λ denotes the magnitude of the OU process and W_t denotes the standard Wiener process. This equation means that the OU process drifts towards the mean μ with a velocity of λ and a volatility driven by a Brownian motion with a variance σ.

Local training process as an OU process. Follow the prior discussion in [21, 24], we formulate the local training process with SGD algorithm as an OU process. In our setting, at round t, client C_i receives the global model $\theta(t)$ and computes the local model update $U_i(\theta)$. In gradient descent, the loss function is $\mathcal{L}(\theta; \mathcal{S}) = \frac{1}{s}\sum_{j=1}^{s} \ell_j(\theta)$, where s is the size of a mini-batch dataset $\mathcal{S} \subseteq D_i$, and $\ell_j(\theta)$ is the loss of a point $s_j \in \mathcal{S}$. The stochastic gradient is $\nabla_\theta \mathcal{L}(\theta; \mathcal{S}) = \frac{1}{s}\sum_{j \in \mathcal{S}} g_{i,j}(\theta)$, where $g_{i,j}(\theta)$ is the stochastic gradient for a data point s_j. In local epoch k with a local model $\theta_i(k)$ and a local learning rate γ, we can get

$$\theta_i(k+1) = \theta_i(k) - \frac{\gamma}{s}\sum_{j \in \mathcal{S}} g_{i,j}(\theta). \tag{3}$$

The stochastic gradient is a sum of s independent, uniformly sampled contributions and the gradient noise is Gaussian with variance $\propto \frac{1}{s}$ [21]. According to the central limit theorem, we know that $\nabla_\theta \mathcal{L}(\theta; \mathcal{S}) \to \mathcal{N}(g_i(\theta), \frac{B(\theta)B(\theta)^T}{s})$, where $g_i(\theta)$ is a full gradient and $B(\theta)B(\theta)^T$ is the corresponding covarience matrix. According to previous discussion in [21], we assume that when θ_i approaches a stationary value, $B(\theta) = B$, which is a constant. Thus the following equation holds:

$$\triangle\theta_i = \theta_i(k+1) - \theta_i(k) \quad \approx -\gamma g_i(\theta) + \frac{\gamma}{\sqrt{s}} B\mathcal{N}(0, I). \tag{4}$$

This equation is a discretization equation of the continuous-time stochastic differential equation,

$$d\theta_i = -g_i(\theta)dt + \frac{\gamma}{\sqrt{s}} BdW_t. \tag{5}$$

Assumption 1. *We assume that, for the client i with a local model $\theta_i(t)$ that satisfies $\theta_i(t) \xrightarrow{t \to +\infty} \theta_i$. For convex model, the local model gradient at round t is close to*

$$g_i(\theta) \approx \lambda_i(\theta_i(t) - \theta_i). \tag{6}$$

Client C_i trains a local model initiated with $\theta(0)$ and finally the model converges to θ_i. We assume that the local model gradient $g_i(\theta)$ can be modeled by the OU process Eq.2, which is almost proportional to the distance between current local model $\theta(t)$ and convergent local model θ_i. Note that this assumption is an ideal and simple OU model for the evolution of local training process.

We can replace the parameter $g_i(\theta)$ in Eq. 5. Then, we can have

$$d\theta_i \approx \lambda_i(\theta_i - \theta_i(t))dt + \frac{\gamma}{\sqrt{s}} BdW_t. \tag{7}$$

Theorem 1. *The local training model's evolution can be represented by $\theta_i(t) \approx \theta(0)e^{-\lambda_i t} + (1 - e^{-\lambda_i t})\theta_i + \frac{\gamma}{\sqrt{s}} e^{-\lambda_i t} \int_0^t e^{\lambda_i t} BdW_t$, which satisfies $\mathbb{E}[\theta_i(t)] \xrightarrow{t \to +\infty} \theta_i$ and $Var[\theta_i(t)] \xrightarrow{t \to +\infty} \frac{\gamma^2 B^2}{2\lambda_i s}$.*

From the local training model's evolution $\theta_i(t)$, we can get that in the initial round, client C_i trained with the initial global model $\theta(0)$ converges to the local model θ_i with a speed of $O(e^{-\lambda_i t})$. The expectation of the local model weights is θ_i and the variation for the convergent local model is $\frac{\gamma^2 B^2}{2\lambda_i s}$, which shows the uncertainty of the process.

Global training process as an OU process. As defined in section 3.1, the local model updates for n clients are $U_1(\theta), U_2(\theta), ..., U_i(\theta), ..., U_n(\theta)$. According to [25], their observation is that when the learning rate is sufficiently small, the effect of E steps of local updates is similar to one step update with a larger learning rate. Therefore, we can assume that $U_i(\theta) \approx d\theta_i$, that is, $U_i(\theta) \approx d\theta_i = -g_i(\theta)dt + \frac{\gamma}{\sqrt{s}} B_i dW_t$. For the global training process defined in Eq.1, similar to the discretization equation and continuous-time stochastic differential equation illustrated in Eq.5, we can further get

$$d\theta(t) = \sum_{i=1}^{n} \frac{N_i}{N} \lambda_i (\theta_i - \theta_i(t))dt + \sum_{i=1}^{n} \frac{N_i}{N} \frac{\gamma}{\sqrt{s}} B_i dW_t. \tag{8}$$

Theorem 2. *The global training model's evolution can be represented by* $\theta(t) \approx \theta(0)e^{-\overline{\lambda}t} + (1 - e^{-\overline{\lambda}t})\overline{\theta} + \sum_{i=1}^{n} \frac{N_i}{N} \frac{\gamma}{\sqrt{s}} e^{-\overline{\lambda}t} \int_0^t B_i e^{\overline{\lambda}t} dW_t$*,which satisfies* $\mathbb{E}[\theta(t)] \xrightarrow{t\to+\infty} \overline{\theta}$ *and* $Var[\theta(t)] \xrightarrow{t\to+\infty} \sum_{i=1}^{n} \frac{\gamma^2 N_i^2 B_i^2}{2\overline{\lambda} N^2 s}$.

We define $\overline{\lambda} = \sum_{i=1}^{n} \frac{N_i \lambda_i}{N}$ and $\overline{\theta} = \frac{\sum_{i=1}^{n} N_i \lambda_i \theta_i}{\sum_{i=1}^{n} N_i \lambda_i}$. The global model $\theta(t)$, initiated with $\theta(0)$, finally converges to $\overline{\theta}$ with a speed of $O(e^{-\overline{\lambda}t})$. For iid setting in FL, the local data can be regarded as samples drawn from the overall distribution that represents the overall distribution. Therefore, we assume that the parameter λ_i is the same for all clients that equals to λ. We can infer that $\overline{\theta} = \sum_{i=1}^{n} \frac{N_i}{N} \theta_i$, that is, the expectation of the global model $\theta(t)$ is the weighted average of all local optimal model, and the variation for the convergent global model is $\sum_{i=1}^{n} \frac{\gamma^2 N_i^2 B_i^2}{2\lambda N^2 s}$.

4 Methods

4.1 Scaled delta free-rider attack

Motivation. When learning rate is small, the model update vectors in two adjacent rounds are almost the same. Thus, we use the weighted average value of all local model updates returned by honest clients $\theta(t) - \theta(t-1)$ as forged local update. As the model tends to converge, the local update gradually decays and tends to zero. We try to simulate honest clients' behavior by using the ratio of previous two rounds' *l2*-norm of local updates as the decay rate. To simulate a real global update $\theta(t+1) - \theta(t)$ in the next round, we can forge local update in our scaled delta attack as follows:

$$U_f(\theta) = \frac{\|\theta(t) - \theta(t-1)\|}{\|\theta(t-1) - \theta(t-2)\|} (\theta(t) - \theta(t-1)). \tag{9}$$

According to the disscussion in section 3.2, when the learning rate is sufficiently small, the effect of E steps of local update is similar to one step update with a larger learning rate. Thus, $U_f(\theta)$ can be viewed as the weighted average of all local model updates with just a one-step training. In FL system, $E = 1$ makes FedAvg equivalent to SGD[25]. Therefore, our fake update could be viewed as a global update over the whole training dataset with a one-step training.

Convergence proof. We analyze our scaled delta attack following the proof of convergence in [9]. We denote by J_1 the set of honest clients and by J_2 the set of m free-riders. The total number of training samples declared by free_riders is M. At round t, the aggregated model with free-riders could be calculated as follows:

$$d\theta(t) \approx \sum_{i \in J_1} \frac{N_i}{N-M} \lambda_i (\theta_i - \theta_i(t))dt + \sum_{i \in J_1} \frac{N_i}{N-M} \frac{\gamma}{\sqrt{s}} B_i dW_t. \tag{10}$$

Theorem 3. *The scaled delta attack has comparable performances in terms of convergence property compared to the training process involving only honest clients. We have* $\theta(t) \approx \theta(0)e^{-\overline{\lambda}t} + (1 - e^{-\overline{\lambda}t})\overline{\theta} + \sum_{i \in J_1} \frac{N_i}{(N-M)} \frac{\gamma}{\sqrt{s}} e^{-\overline{\lambda}t} \int_0^t B_i e^{\overline{\lambda}t} dW_t$*, which satisfies* $\mathbb{E}[\theta(t)] \xrightarrow{t\to+\infty} \overline{\theta}$ *and* $Var[\theta(t)] \xrightarrow{t\to+\infty} \sum_{i \in J_1} \frac{\gamma^2 N_i^2 B_i^2}{2\overline{\lambda}(N-M)^2 s}$.

We denote $\overline{\lambda} = \sum_{i \in J_1} \frac{N_i}{N-M} \lambda_i$ and $\overline{\theta} = \frac{\sum_{i \in J_1} N_i \lambda_i \theta_i}{\sum_{i \in J_1} N_i \lambda_i}$. Similar to Theorem 2, we know that the convergence rate, the final global model and the uncertainty of the training process with free-riders has the same form of expression as the training process with only honest clients. Our proposed scaled delta attack has comparable convergence performance with normal FL training.

4.2 Advanced free-rider attack

The scaled delta attack hasn't a good stealth property and could be detected easily. In this scaled delta attack, we can obtain the weighted average update of all clients as $\overline{U}(\theta) = \sum_{i=1}^{n} \frac{N_i}{N} U_i(\theta)$, which is a high-dimensional vector. If we can find the relationship (e.g, cosine similarity and *l2*-norm) between these local updates, we can construct a fake update similar to a real update.

Analysis for cosine similarity of local updates. In general, the expectation of local model weights for n clients $\theta_1, \theta_2, .., \theta_n$ could be viewed as a p-dimensional vector, p is the dimension of the local model parameters after being flattened. As shown in Theorem 2, we know that the global model weights converge to $\overline{\theta} = \sum_{i=1}^{n} \frac{N_i}{N}\theta_i$ in iid setting. And we denote $\epsilon_i = \theta_i - \overline{\theta}$, which demonstrates the deviation between local client model and global model, which is resulted from the difference between local datasets and whole dataset. We assume that the smallest space which contains all n vectors is Ω, and θ_i is a random vector which deviates from the vector $\overline{\theta}$. For iid setting in FL, the local data can be regarded as samples drawn from the overall distribution which represents the overall distribution. Therefore, these n vectors $\theta_1, \theta_2, .., \theta_n$ could be viewed as randomly distributed points around point $\overline{\theta}$. For client C_i, we set $\mathbb{E}(|\epsilon_i|) = \epsilon$. The following theory shows the relationship between the number of training rounds t and the cosine value of the angle generated from the local model update vectors for any two clients.

Lemma 1. *In iid setting, the cosine value between two local updates is* $\cos\beta$. *The relationship between* $\mathbb{E}(\cos\beta)$ *and the number of training rounds t satisfies the equation:* $\mathbb{E}(\cos\beta) \approx \frac{C^2}{C^2+e^{2\overline{\lambda}t}}$.

In Lemma 1, C is an introduced constant parameter that shows the multiple relationship between two expectations, that is, $\mathbb{E}(|\overline{\theta} - \theta(0)|) = C\mathbb{E}(|\epsilon_i|)$. When $t = 0$, the value of $\mathbb{E}(\cos\beta) \approx 1$, which means in the initial stage of FL training, random two local updates almost have the same direction with each other. With the training process moving forward, the value of $\mathbb{E}(\cos\beta)$ is decreasing with a rate of $O(e^{-\overline{\lambda}t})$. When the global model converges, that is, when $t \to \infty$,the value of $\mathbb{E}(\cos\beta)$ is close to 0, that is, random two local updates are almost orthogonal to each other, which means that when the model is close to convergence, differences in local model updates caused by the specificity of local datasets will gradually appear.

Lemma 2. *In iid setting, the l2-norm relationship between the local model update* $U_i(\theta)$ *and the weighted average value of all local model updates* $\overline{U}(\theta)$ *is* $\frac{|\mathbb{E}(U_i(\theta))|}{|\mathbb{E}(\overline{U}(\theta))|} = \sqrt{\frac{n^2}{n+(n^2-n)\mathbb{E}(\cos\beta)}}$. *And the vector difference between* $U_i(\theta)$ *and* $\overline{U}(\theta)$ *is orthogonal to vector* $\overline{U}(\theta)$.

We denote $\overline{U}(\theta) = \theta(t) - \theta(t-1) = \sum_{i=1}^{n} \frac{Ni}{N} U_i(\theta)$, which is a high-dimensional vector constructed by all local model updates. We also know how to simulate the expectation of $\cos\beta$ for random two vectors $U_i(\theta)$ and $U_j(\theta)$. Therefore, in the next section, we will show how to construct our advanced free-rider attack based on Lemma 1 and Lemma 2.

For iid setting, according to previous discussion in section 4.1, we know that our fake update $U_f(\theta) = \frac{\|\theta(t)-\theta(t-1)\|}{\|\theta(t-1)-\theta(t-2)\|}(\theta(t) - \theta(t-1))$ could be viewed as a global update over the whole training dataset with one-step training, and approximately equal to the weighted average value of all local model updates in the next round. Because of the difference between local datasets and whole dataset, there exists weight divergence between local client model and global model. Then, we analyze and quantize the weight divergence in Lemma 2. Thus, by exploiting the conclusion that in high-dimensional spaces, random vectors are orthogonal[26], our advanced strategy is to add Gaussian noise $\varphi(t)$ satisfying $|\varphi(t)|= \sqrt{\frac{n^2}{n+(n^2-n)E(\cos\beta)} - 1}|U_f(\theta)|$, which is a simulated deviation between $U_f(\theta)$ and a real local model update $U_i(\theta)$. The advanced free-rider model update is constructed by equation $\hat{U}_f(\theta) = U_f(\theta) + \varphi(t)$. The details of advanced free-rider attack are shown in Algorithm 1.The attacker firstly calculates the expectations of global updates to estimate parameters $\overline{\lambda}$ and $\cos\beta$. Then it chooses a random fraction of dimensions in the scaled delta update to add Gaussian noise, which simulate specificity of the training on local dataset.

Algorithm 1: AdvancedAttack. The malicious client C_f carries out the advanced free-rider attack based on global model $\theta(t)$ received at round t and $\theta(t-1)$ at round $t-1$.

```
Parameter estimation.
$l(t) \leftarrow |\mathbb{E}(\theta(t) - \theta(t-1))|$
$l(1) \leftarrow |\mathbb{E}(\theta(1) - \theta(0))|$
$\overline{\lambda} \leftarrow \ln \sqrt[t-1]{\frac{l(t)}{l(1)}}$
$\mathbb{E}(\cos\beta) \approx \frac{C^2}{C^2+e^{2\overline{\lambda}t}}$
Update generation.
$U_f(\theta) = \frac{\|\theta(t)-\theta(t-1)\|}{\|\theta(t-1)-\theta(t-2)\|}(\theta(t) - \theta(t-1))$
$|\varphi(t)|= \sqrt{\frac{n^2}{n+(n^2-n)\mathbb{E}(\cos\beta)} - 1}|\mathbb{E}(U_f(\theta))|$
Choose a fraction of parameter dimension as $d$
Add Gaussian noise $|\varphi(t)|\mathcal{N}(0, \frac{1}{d})$ to $U_f(\theta)$
Return the constructed model update $\hat{U}_f(\theta)$ to server
```

For highly skewed non-iid data, the local updates are naturally non-iid with each other. We cannot find certain geometric relationship between all local updates. Our Lemma 1 and Lemma 2 cannot be applied. However, due to the heterogeneity of local data, the local updates are spreaded in a large space which is beneficial for a free-rider to hide its fake update. We still use the advanced attack to carry out the experiments for non-iid setting.

Convergence proof. Similar to the proof of Eq. 10, we could have

$$d\theta(t) \approx \sum_{i\in J_1} \frac{N_i}{N-M}\lambda_i(\theta_i - \theta_i(t))dt + \sum_{i\in J_1} \frac{N_i}{N-M}\frac{\gamma}{\sqrt{s}}B_i dW_{t_1} + \frac{M}{N-M}\varphi(t)dW_{t_2}. \tag{11}$$

Theorem 4. *Our proposed advanced attack has comparable performance in terms of convergence property compared to the training process involving only honest clients. We have* $\theta(t) \approx \theta(0)e^{-\overline{\lambda}t} + (1 - e^{-\overline{\lambda}t})\overline{\theta} + \sum_{i\in J_1} \frac{N_i}{N-M}\frac{\gamma}{\sqrt{s}}e^{-\overline{\lambda}t}\int_0^t B_i e^{\overline{\lambda}t}dW_{t_1} + \frac{M}{N-M}e^{-\overline{\lambda}t}\int_0^t \varphi(t)e^{\lambda t}dW_{t_2}$*, which satisfies* $\mathbb{E}[\theta(t)] \xrightarrow{t\to+\infty} \overline{\theta}$ *and* $Var[\theta(t)] \xrightarrow{t\to+\infty} \sum_{i\in J_1} \frac{\gamma^2 N_i^2 B_i^2}{2\overline{\lambda}(N-M)^2 s}$.

We denote $\overline{\lambda} = \sum_{i\in J_1} \frac{N_i}{N-M}\lambda_i$ and $\overline{\theta} = \frac{\sum_{i\in J_1} N_i\lambda_i\theta_i}{\sum_{i\in J_1} N_i\lambda_i}$. Similar to Theorem 3, we get the same conclusion that our proposed advanced attack also have comparable convergence performance with normal FL training.

5 Performance evaluation

The goal of our experiment is to highlight that, compared with the state of the art free-rider attacks, our advanced free-rider attack has a good performance in terms of convergence and stealth property. We conduct experiments on real world datasets and simulate a FL environment. The resulting model under advanced attack has comparable performances and convergence rate with respect to the one of the model obtained with honest clients only. Besides, the results demonstrate that in both iid and non-iid settings, our advanced attack can stay stealthy under different indicative features. Furthermore, we utilize DAGMM as an anomaly detection method to defense and evaluate our advanced attack.

Tasks. We evaluate our advanced free-rider attack on two image classification tasks: **1) Task1.** MNIST[27] with multi-layer perceptron(MLP)[28], **2) Task2.** Fashion-MNIST[29] with convolutional neural network(CNN). For each task, we maintain the same experimental setting composed by 50 honest clients, and we set the number of free-riders to respectively 1, 50 and 150 in both iid and non-iid setting. Each client has the same number of training samples. Details of dataset and experimental setups are provided in the appendix.

Compared attacks. We compare latest free-rider attacks as follows: **1) Plain free-rider.** It is the simplest way to construct fake local update. The free-rider directly returns the received global model without any local training. Since this attack is trivial for detection, we use it as a baseline strategy for free-riders. **2) Disguised free-rider.** This attack is based on additive stochastic perturbations. Specifically, the free-rider utilizes GMM to fit a multimodal distribution forms for the local update. **3) Delta weights.** This attack subtracts the current global model to the previous global model.

Features. Byzantine-tolerant aggregation mechanisms and DP-based defense methods focus on mitigating the influence of the attacker, they don't detect the attacker and aren't suitable for free-rider attacks. Thus, we only consider following three indicative features which can distinguish free-riders: **1) Cosine similarity.** We compute cosine values between other clients' local model updates and a randomly chosen client's update vector. **2) L2-norm.** We compute *l2*-norm of every clients' local model updates. **3) Standard eviation(STD).** We compute the std value of each clients' local model vector.

DAGMM. This is a model for unsupervised anomaly detection. The model contains two main components: a compression network and an estimation network. the compression network performs dimensionality reduction for input samples by a deep autoencoder, prepares their low-dimensional representations from both the reduced space and the reconstruction error features, and feeds the representations to the subsequent estimation network; Then, the estimate the network obtains the feeds and predicts their energy in the Gaussian Mixture Model (GMM) framework.

Results. The experimental results for both non-iid setting of **Task1** and **Task2** can be found in the

Convergence and performances.

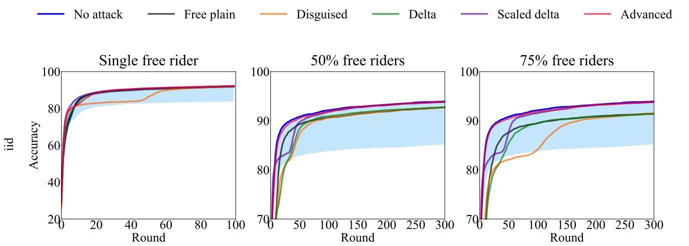

Figure 1: Accuracy performance in iid setting of MNIST dataset

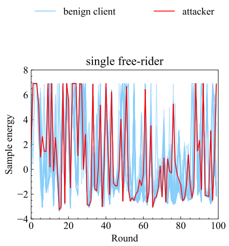

Figure 2: Sample energy in iid setting of MNIST dataset

As shown in Figure 1, we show the evolution of the global model's accuracy for iid setting in three scenarios where the number of free-riders is 1 and accounts for 50% and 75% of all clients. The shaded blue region indicates the variability of FL model with only honest clients which is estimated from 10 different training initialization [2].

The results indicate that, independently from the chosen free-rider attack, the resulting global models have comparable performances compared with the global model obtained with only honest clients. However, the convergence speed of different attacks is different. In 50% free-rider scenarios and 75% free-rider scenarios, the convergence rate of other attacks such as plain free-rider, disguised free-rider and delta weights is significantly lower than the normal convergence rate. However, our scaled delta and advanced free-rider attacks have comparable performances in terms of convergence rate compared to the training process involving only honest clients. This result is also in agreement with Theorem 3 and Theorem 4 which suggest that the accuracy and convergence rate of the final global model is not affected by the number of free-riders.

Stealth property. In this section, we study the stealth property of various attack methods by comparing indicative features. Since plain free-rider attack is easy to be detected, we only compare our attack methods with disguised free-rider and delta weights attacks.

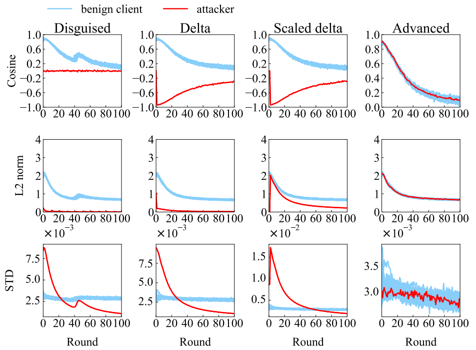

Figure 3: Features in iid setting of MNIST dataset

As shown in Figure 3, we observe that the local update of our advanced attack achieves stealth property in all indicators. The other attack methods do not have a good stealth property. Specifically, the evolution of cosine similarity and $l2$-norm of our advanced attack confirms Lemma 1 and Lemma 2. Note that in the initial several rounds, our advanced attack participate the training process honestly to compute core parameters such as C which can be used in the simulation of local update.

Anomaly detection. To defense our advanced attack, we also utilize a popular anomaly detection model DAGMM. The results of all sample energy values are shown in Figure 2. The sample energy values of free-riders are hidden in the energy value area of other honest clients in iid setting, which is difficult to distinguish by DAGMM. The results indicate that our advanced attack is powerful and can escape from the cluster-based anomaly detection techniques.

6 Conclusion and future work

In this paper, we propose scaled delta attack which can guarantee a convergent global model. Then we also present our advanced free-rider attack by improving the stealth property of scaled delta attack. Detailed experiments validate that our advanced free-rider attack has good performance and maintain stealth against existing defense strategies. For future work, we would like to study how to carry out our free-rider attacks in a real-world setting(such as partial client participation, secure aggregation[30], etc.), and investigate more defense strategies to protect FL systems against advanced free-rider attack.

[2] In our experiment for MNIST, the maximum global epoch for the training process is 300 due to our limited computation resources

Acknowledgements

This work is financially supported by National Natural Science foundation of China under Grant No. 61732022 and National Key R&D Program of China under Grant No. 2017YFB0803002.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[2] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[3] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019.

[4] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1885–1893, 2016.

[5] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3517–3529, 2017.

[6] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1596–1606. PMLR, 2019.

[7] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. *CoRR*, abs/1911.11815, 2019.

[8] Jierui Lin, Min Du, and Jian Liu. Free-riders in federated learning: Attacks and defenses. *CoRR*, abs/1911.12560, 2019.

[9] Yann Fraboni, Richard Vidal, and Marco Lorenzi. Free-rider attacks on model aggregation in federated learning. *CoRR*, abs/2006.11901, 2020.

[10] Qiang Yang, Lixin Fan, and Han Yu, editors. *Federated Learning - Privacy and Incentive*, volume 12500 of *Lecture Notes in Computer Science*. Springer, 2020.

[11] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett,

editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 119–129, 2017.

[12] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *POMACS*, 1(2):44:1–44:25, 2017.

[13] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR, 2018.

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

[15] Robin C. Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *CoRR*, abs/1712.07557, 2017.

[16] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *CoRR*, abs/1808.04866, 2018.

[17] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Ashfaqur Rahman, Jeremiah D. Deng, and Jiuyong Li, editors, *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Gold Coast, Australia, QLD, Australia, December 2, 2014*, page 4. ACM, 2014.

[18] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Anomaly detection using autoencoders in high performance computing systems. In Mario Alviano, Gianluigi Greco, Marco Maratea, and Francesco Scarcello, editors, *Discussion and Doctoral Consortium papers of AI*IA 2019 - 18th International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019*, volume 2495 of *CEUR Workshop Proceedings*, pages 24–32. CEUR-WS.org, 2019.

[19] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[20] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers*, pages 177–186. Physica-Verlag, 2010.

[21] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. A variational analysis of stochastic gradient algorithms. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 354–363. JMLR.org, 2016.

[22] Yazhen Wang and Shang Wu. Asymptotic analysis via stochastic differential equations of gradient descent algorithms in statistical and computational paradigms. *J. Mach. Learn. Res.*, 21:199:1–199:103, 2020.

[23] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 483–513. PMLR, 2020.

[24] Mónica Ribero and Haris Vikalo. Communication-efficient federated learning via optimal client sampling. *CoRR*, abs/2007.15197, 2020.

[25] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[26] T Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *J. Mach. Learn. Res.*, 14(1):1837–1864, 2013.

[27] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[28] H Taud and JF Mas. Multilayer perceptron (mlp). In *Geomatic Approaches for Modeling Land Change Scenarios*, pages 451–455. Springer, 2018.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[30] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.