Free-rider Attacks on Model Aggregation in Federated Learning

Yann Fraboni[1,2] **Richard Vidal**[2] **Marco Lorenzi**[1]
[1] Université Côte d'Azur, Inria Sophia Antipolis, Epione Research Group, France
and [2] Accenture Labs, Sophia Antipolis, France

Abstract

Free-rider attacks against federated learning consist in dissimulating participation to the federated learning process with the goal of obtaining the final aggregated model without actually contributing with any data. This kind of attacks is critical in sensitive applications of federated learning, where data is scarce and the model has high commercial value. We introduce here the first theoretical and experimental analysis of free-rider attacks on federated learning schemes based on iterative parameters aggregation, such as FedAvg or FedProx, and provide formal guarantees for these attacks to converge to the aggregated models of the fair participants. We first show that a straightforward implementation of this attack can be simply achieved by not updating the local parameters during the iterative federated optimization. As this attack can be detected by adopting simple countermeasures at the server level, we subsequently study more complex disguising schemes based on stochastic updates of the free-rider parameters. We demonstrate the proposed strategies on a number of experimental scenarios, in both iid and non-iid settings. We conclude by providing recommendations to avoid free-rider attacks in real world applications of federated learning, especially in sensitive domains where security of data and models is critical.

1 Introduction

Federated learning is a training paradigm that has gained popularity in the last years as it enables different clients to jointly learn a global model without sharing their respective data. It is particularly suited for Machine Learning applications in domains where data security is critical, such as healthcare [Brisimi et al., 2018, Silva et al., 2019]. The relevance of this approach is witnessed by current large scale federated learning initiatives under development in the medical domain, for instance for learning predictive models of breast cancer[1], or for drug discovery and development[2].

The participation to this kind of research initiatives is usually exclusive and typical of applications where data is scarce and unique in its kind. In these settings, aggregation results entail critical information beyond data itself, since a model trained on exclusive datasets may have very high commercial or intellectual value. For this reason, providers may not be interested in sharing the model: the commercialization of machine learning products would rather imply the availability of the model as a service through web- or cloud-based API. This is due to the need of preserving the intellectual property on the model components, as well as to avoid potential information leakage, for example by limiting the maximum number of queries allowed to the users [Carlini et al., 2019, Fredrikson et al., 2015, Ateniese et al., 2015].

This critical aspect can lead to the emergence of opportunistic behaviors in federated learning, where ill-intentioned clients may participate with the aim of obtaining the federated model, without actually contributing with any data during the training process. In particular, the attacker, or free-rider, aims at disguising its participation to federated learning while ensuring that the iterative training process ultimately converges to the wished target: the aggregated model of the fair participants. Free-riding attacks performed by ill-intentioned participants ultimately open federated learning initiatives to intellectual property loss and data privacy breaches, taking place for example in the form of model inversion [Fredrikson et al., 2014, Fredrikson et al., 2015].

The study of security and safety of federated learning is an active research domain, and several kind of attacks are matter of ongoing studies. For example, an attacker may interfere during the iterative federated learning procedure to degrade/modify mod-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

[1] blogs.nvidia.com/blog/2020/04/15/federated-learning-mammogram-assessment/

[2] www.imi.europa.eu/projects-results/project-factsheets/melloddy

els performances [Bhagoji et al., 2019, Li et al., 2016, Yin et al., 2018, Xie et al., 2019, Shen et al., 2016], or retrieve information about other clients' data [Wang et al., 2019, Hitaj et al., 2017]. Since currently available defence methods such as [Fung et al., 2020, Bhagoji et al., 2019] are generally based on outliers detection mechanisms, they are generally not suitable to prevent free-riding, as this kind of attack is explicitly conceived to stay undetected while not perturbing the FL process. Free-riding may become a critical aspect of future machine learning applications, as federated learning is rapidly emerging as the standard training scheme in current cooperative learning initiatives. To the best of our knowledge, the only investigation is in a preliminary work [Lin et al., 2019] focusing on attack strategies operated on federated learning based on gradient aggregation. However, no theoretical guarantees are provided for the effectiveness of this kind of attacks. Furthermore this setup is unpractical in many real world applications, where federated training schemes based on model averaging are instead more common, due to the reduced data exchange across the network. FedAvg [McMahan et al., 2017] is the most representative framework of this kind, as it is based on the iterative averaging of the clients models' parameters, after updating each client model for a given number of training epochs at the local level. To improve the robustness of FedAvg in non-iid and heterogeneous learning scenarios, FedProx [Li et al., 2018] extends FedAvg by including a regularization term penalizing local departures of clients' parameters from the global model.

The contribution of this work consists in the development of a theoretical framework for the study of free-rider attacks in federated learning schemes based on model averaging, such as in FedAvg and FedProx. The problem is here formalized via the reformulation of federated learning as a stochastic process describing the evolution of the aggregated parameters across iterations. To this end, we build upon previous works characterizing the evolution of model parameters in Stochastic Gradient Descent (SGD) as a continuous time process [Mandt et al., 2017, Orvieto and Lucchi, 2018, Li et al., 2017, He et al., 2018]. A critical requirement for opportunistic free-rider attacks is to ensure the convergence of the training process to the wished target represented by the aggregated model of the fair clients. We show that the proposed framework allows to derive explicit conditions to guarantee the success of the attack. This is an important theoretical feature as it is of primary interest for the attacker to not interfere with the learning process.

We first derive in Section 2.4 a basic free-riding strategy to guarantee the convergence of federated learning to the model of the fair participants. This strategy simply consists in returning at each iteration the received global parameters. As this behavior could easily be detected by the server, we build more complex strategies to disguise the free-rider contribution to the optimization process, based on opportune stochastic perturbations of the parameters. We demonstrate in Section 2.5 that this strategy does not alter the global model convergence, and in Section 3 we experimentally demonstrate our theory on a number of learning scenarios in both iid and non-iid settings. All proofs and additional material are provided in the Appendix.

2 Methods

Before introducing in Section 2.2 the core idea of free-rider attacks, we first recapitulate in Section 2.1 the general context of parameter aggregation in federated learning.

2.1 Federated learning through model aggregation: FedAvg and FedProx

In federated learning, we consider a set I of participating clients respectively owning datasets $\mathcal{D}_i$ composed of M_i samples. During optimization, it is generally assumed that the D elements of the clients' parameters vector $\boldsymbol{\theta}_i^t = (\theta_{i,0}^t, \theta_{i,1}^t, ..., \theta_{i,D}^t)$, and the global parameters $\boldsymbol{\theta}^t = (\theta_0^t, \theta_1^t, ..., \theta_D^t)$ are aggregated independently at each iteration round t. Following this assumption, and for simplicity of notation, in what follows we restrict our analysis to a single parameter entry, that will be generally denoted by θ_i^t and θ^t for clients and server respectively.

In this setting, to estimate a global model across clients, FedAvg [McMahan et al., 2017] is an iterative training strategy based on the aggregation of local model parameters θ_i^t. At each iteration step t, the server sends the current global model parameters θ^t to the clients. Each client updates the model by minimizing over E epochs the local cost function $\mathcal{L}(\theta_i^{t+1}, \mathcal{D}_i)$ initialized with θ^t, and subsequently returns the updated local parameters θ_i^{t+1} to the server. The global model parameters θ^{t+1} at the iteration step $t+1$ are then estimated as a weighted average:

$$\theta^{t+1} = \sum_{i \in I} \frac{M_i}{N} \theta_i^{t+1}, \tag{1}$$

where $N = \sum_{i \in I} M_i$ represents the total number of samples across distributed datasets. FedProx [Li et al., 2018] builds upon FedAvg by adding to the cost function a L2 regularization term penalizing the deviation of the local parameters θ_i^{t+1} from the global parameters θ^t. The new cost function is $\mathcal{L}_{Prox}(\theta_i^{t+1}, \mathcal{D}_i, \theta^t) = \mathcal{L}(\theta_i^{t+1}, \mathcal{D}_i) + \frac{\mu}{2} \left\| {\theta_i}^{t+1} - \theta^t \right\|^2$ where μ is the hyperparameter monitoring the regularization by enforcing proximity between local update ${\theta_i}^{t+1}$ and reference model θ^t.

Algorithm 1: Free-riding in federated learning

```
Input: learning rate $\lambda$, epochs $E$, initial model $\theta^0$,
       batch size $S$
$\tilde{\theta}^0 = \theta^0$;
for each round t=0,...,T-1 do
    Send the global model $\tilde{\theta}^t$ to all the clients;
    for each fair client $j \in J$ do
        $\tilde{\theta}_j^{t+1} = ClientUpdate(\tilde{\theta}^t, E, \lambda)$;
        Send $\tilde{\theta}_j^{t+1}$ to the server;
    for each free-rider $k \in K$ do
        if disguised free-rider then
            $\tilde{\theta}_k^{t+1} = \tilde{\theta}^t + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_k^2)$;
        else
            $\tilde{\theta}_k^{t+1} = \tilde{\theta}^t$
        Send $\tilde{\theta}_k^{t+1}$ to the server;
    $\tilde{\theta}^{t+1} = \sum_{j \in J} \frac{M_j}{N} \tilde{\theta}_j^{t+1} + \sum_{k \in K} \frac{M_k}{N} \tilde{\theta}_k^{t+1}$;
```

2.2 Formalizing Free-rider attacks

Aiming at obtaining the aggregated model of the fair clients, the strategy of a free-rider consists in participating to federated learning by dissimulating local updating through the sharing of opportune counterfeited parameters. The free-riding attacks investigated in this work are illustrated in Algorithm 1, and analysed in the following sections from both theoretical and experimental standpoints.

We denote by J the set of fair clients, i.e. clients following the federated learning strategy of Section 2.1 and by K the set of free-riders, i.e. malicious clients pretending to participate to the learning process, such that $I = J \cup K$ and $J \neq \emptyset$. We denote by M_K the number of samples declared by the free-riders.

2.3 SGD perturbation of the fair clients local model

To describe the clients' parameters observed during federated learning, we rely on the modeling of Stochastic Gradient Descent (SGD) as a continuous time stochastic process [Mandt et al., 2017, Orvieto and Lucchi, 2018, Li et al., 2017, He et al., 2018].

For a client j, let us consider the following form for the loss function:

$$\mathcal{L}_j(\theta_j) = \frac{1}{M_j} \sum_{n=1}^{M_j} l_{n,j}(\theta_j), \tag{2}$$

where M_j is the number of samples owned by the client, and $l_{n,j}$ is the contribution to the overall loss from a single observation $\{x_{n,j}; y_{n,j}\}$. The gradient of the loss function is defined as $g_j(\theta_j) \equiv \nabla \mathcal{L}_j(\theta_j)$.

We represent SGD by considering a minibatch $\mathcal{S}_{j,k}$, composed of a set of S different indices drawn uniformly at random from the set $\{1, \dots, M_j\}$, each of them indexing a function $l_{n,j}(\theta_j)$ and where k is the index of the minibatch. Based on $\mathcal{S}_{j,k}$, we form a stochastic estimate of the loss,

$$\mathcal{L}_{\mathcal{S}_{j,k}}(\theta_j) = \frac{1}{S} \sum_{n \in \mathcal{S}_{j,k}} l_{n,j}(\theta_j), \tag{3}$$

where the corresponding stochastic gradient is defined as $g_{\mathcal{S}_{j,k}}(\theta_j) \equiv \nabla \mathcal{L}_{\mathcal{S}_{j,k}}(\theta_j)$.

By observing that gradient descent is a sum of S independent and uniformly distributed samples, thanks to the central limit theorem, gradients at the client level can thus be modeled by a Gaussian distribution

$$g_{\mathcal{S}_{j,k}}(\theta_j) \sim \mathcal{N}(g_j(\theta_j), \frac{1}{S}\sigma_j^2(\theta_j)), \tag{4}$$

where $g_j(\theta_j) = \mathbb{E}_s\left[g_{\mathcal{S}_{j,k}}(\theta_j)\right]$ is the full gradient of the loss function in equation (2) and $\sigma_j^2(\theta_j)$ is the variance associated with the loss function in equation (3).

SGD updates are expressed as:

$$\theta_j(u_j + 1) = \theta_j(u_j) - \lambda g_{\mathcal{S}_{j,k}}(\theta_j(u_j)), \tag{5}$$

where u_j is the SGD iteration index and λ is the learning rate set by the server.

By defining $\Delta\theta_j(u_j) = \theta_j(u_j+1) - \theta_j(u_j)$, we can rewrite the update process as

$$\Delta\theta_j(u_j) = -\lambda g_j(\theta_j(u_j)) + \frac{\lambda}{\sqrt{S}} \sigma_j(\theta_j) \Delta W_j, \tag{6}$$

where $\Delta W_j \sim \mathcal{N}(0, 1)$. The resulting continuous-time model [Mandt et al., 2017, Orvieto and Lucchi, 2018, Li et al., 2017, He et al., 2018] is

$$\mathrm{d}\theta_j = -\lambda g_j(\theta_j) \mathrm{d}u_j + \frac{\lambda}{\sqrt{S}} \sigma_j(\theta_j) \mathrm{d}W_j. \tag{7}$$

where W_j is a continuous time Wiener Process.

Similarly as in [Mandt et al., 2017], we assume that $\sigma_j(\theta_j)$ is approximately constant with respect to θ_j for the client's stochastic gradient updates between t and $t + 1$, and will therefore denote $\sigma_j(\theta_j) = \sigma_j^t$. Following [Mandt et al., 2017], we consider a local quadratic approximation for the client's loss, leading to a linear form for the gradient $g_j(\theta_j) \simeq r_j[\theta_j - \theta_j^*]$, where $r_j \in \mathbb{R}^+$ depends on the approximation of the cost function around the local minimum θ_j^*. This assumption enables rewriting equation (7) as an Ornstein-Uhlenbeck process [Uhlenbeck and Ornstein, 1930]. Starting from the initial condition represented by θ^t, the global model received at the iteration t, we characterize the local updating of the parameters through equation (7), and we follow the evolution up to the time $\frac{EM_j}{S}$, where E is the number of epochs, and M_j is the number of samples owned by the client. Assuming that M_j is a multiple of S, the number of samples per minibatch, the quantity $\frac{EM_j}{S}$ represents the total number of SGD steps run by the client. The updated model θ_j^{t+1} uploaded to the server therefore takes the form:

$$\theta_j^{t+1} = \underbrace{e^{-\lambda r_j \frac{EM_j}{S}}[\theta^t - \theta_j^*] + \theta_j^*}_{\hat{\theta}_j^{t+1}} + \frac{\lambda}{\sqrt{S}} \int_{u=0}^{\frac{EM_j}{S}} e^{-\lambda r_j \left(\frac{EM_j}{S} - u\right)} \sigma_j^t dW_u. \tag{8}$$

We note that the relative number of SGD updates for the fair clients, $\frac{EM_j}{S}$, influences the parameter $\eta_j = e^{-\lambda r_j \frac{EM_j}{S}}$, which becomes negligible for large values of E.

The variance introduced by SGD can be rewritten as

$$\mathrm{Var}\left[\theta_j^{t+1} | \theta^t\right] = \underbrace{\frac{\lambda}{S} {\sigma_j^t}^2 \frac{1}{2 r_j} \left[1 - e^{-2\lambda r_j \frac{EM_j}{S}}\right]}_{{\rho_j^t}^2}, \tag{9}$$

where we can see that the higher $\frac{EM_j}{S}$, the lower the overall SGD noise. The noise depends on the local loss function r_j, on the server parameters (number of epochs E, learning rate λ, and number of samples per minibatch S), and on the clients' data specific parameters (SGD variance ${\sigma_j^t}^2$).

Equation (8) shows that clients' parameters observed during federated learning can be expressed as $\theta_j^t = \hat{\theta}_j^t + \rho_j^t \zeta_{j,t}$, where, given θ^t, $\hat{\theta}_j^t$ is a deterministic component corresponding to the model obtained with $\frac{EM_j}{S}$ steps of gradient descents, and $\zeta_{j,t}$ is a delta-correlated Gaussian white noise. We consider in what follows a constant local noise variance σ_j^2 (this assumption will be relaxed in Section 2.5.3 to consider instead time-varying noise functions ρ_j^t).

Based on this formalism, in the next Section we study a basic free-rider strategy simply consisting in returning at each iteration the received global parameters. We call this type of attack *plain free-riding*.

2.4 Plain free-riding

We denote by $\tilde{\theta}$ and $\tilde{\theta}_j$ respectively the global and local model parameters obtained in presence of free-riders. The plain free-rider returns the same model parameters as the received ones, i.e. $\forall k \in K$, $\tilde{\theta}_k^{t+1} = \tilde{\theta}^t$. In this setting, the server aggregation process (1) can be rewritten as:

$$\tilde{\theta}^{t+1} = \sum_{j \in J} \frac{M_j}{N} \tilde{\theta}_j^{t+1} + \frac{M_K}{N} \tilde{\theta}^t \ , \tag{10}$$

where $\tilde{\theta}^t$ is the global model and $\tilde{\theta}_j^t$ are the fair clients' local models uploaded to the server for free-riding.

2.4.1 Free-riders perturbation of the fair clients local model

In this section, we investigate the effect of the free-riders on the local optimization performed by the fair clients at every server iteration. The participation of the free-riders to federated learning implies that the processes of the fair clients are being perturbed by the attacks throughout training. In particular, the initial conditions of the local optimization problems are modified according to the perturbed aggregation of equation (10).

Back to the assumptions of Section 2.3 , the initial condition $\tilde{\theta}^t$ of the local optimization includes now the aggregated model of the fair clients and a perturbation coming from the free-riders. Thus, equation (8) in presence of free-riding can be written as

$$\tilde{\theta}_j^{t+1} = \eta_j [\tilde{\theta}^t - \theta_j^*] + \theta_j^* + \frac{\lambda}{\sqrt{S}} \int_{u=0}^{\frac{EM_j}{S}} e^{-\lambda r_j \left(\frac{EM_j}{S} - u\right)} \tilde{\sigma}_j^t dW_u, \tag{11}$$

where $\tilde{\sigma}_j^t = \sigma_j^t(\tilde{\theta}_j)$ is the SGD variance for free-riding. We consider that $\tilde{\sigma}_j^t = \sigma_j^t = \sigma_j$. This assumption will be relaxed in Section 2.5.3 to consider instead time-varying noise functions. With analogous considerations to those made in Section 2.3, the updated parameters take the form:

$$\tilde{\theta}_j^{t+1} = \eta_j [\tilde{\theta}^t - \theta_j^*] + \theta_j^* + \rho_j \tilde{\zeta}_{j,t}, \tag{12}$$

where $\tilde{\zeta}_{j,t}$ is a delta-correlated Gaussian white noise. Similarly as for federated learning, $\mathbb{E}\left[\tilde{\theta}_j^{t+1} | \tilde{\theta}^t\right] = \eta_j [\tilde{\theta}^t - \theta_j^*] + \theta_j^*$, and $\mathrm{Var}\left[\tilde{\theta}_j^{t+1} | \tilde{\theta}^t\right] = \rho_j^2$.

We want to express the global optimization process $\tilde{\theta}^t$ due to free-riders in terms of a a perturbation of the equivalent stochastic process θ^t obtained with fair clients only. Theorem 1 provides a recurrent form for the difference between these two processes.

Theorem 1. Under the assumptions of Section 2.3 and 2.4 for the local optimization processes resulting from federated learning with respectively only fair clients and with free-riders, the difference between the aggregation processes of formulas (1) and (10) takes the following recurrent form:

$$\tilde{\theta}^t - \theta^t = \sum_{i=0}^{t-1} \left(\epsilon + \frac{M_K}{N}\right)^{t-i-1} f(\theta^i) + \sum_{i=0}^{t-1} \left(\epsilon + \frac{M_K}{N}\right)^{t-i-1} (\tilde{\nu}_i - \nu_i), \tag{13}$$

with $f(\theta^t) = \frac{M_K}{N}\left[\theta^t - \sum_{j \in J} \frac{M_j}{N - M_K}[\eta_j(\theta^t - \theta_j^*) + \theta_j^*]\right]$, $\epsilon = \sum_{j \in J} \frac{M_j}{N} \eta_j$, $\nu_t = \sum_{j \in J} \frac{M_j}{N - M_K} \rho_j \zeta_{j,t}$ and $\tilde{\nu}_t = \sum_{j \in J} \frac{M_j}{N} \rho_j \tilde{\zeta}_{j,t}$.

We note that in the special case with no free-riders (i.e. $M_K = 0$), the quantity $\tilde{\theta}^t - \theta^t$ depends on the second term of equation (13) only, and represents the comparison between two different realizations of the stochastic process associated to the federated global model. Theorem 1 shows that in this case the variance across optimization results is non-zero, and depends on the intrinsic variability of the local optimization processes quantified by the variable ν_t. We also note that in presence of free-riders the convergence to the model obtained with fair clients depends on the relative sample size declared by the free-riders $\frac{M_K}{N}$.

2.4.2 Convergence analysis of plain free-riding

Based on the relationship between the learning processes established in Theorem 1, we are now able to prove that federated learning with plain free-riders defined in equation (10) converges in expectation to the aggregated model of the fair clients of equation (1).

Theorem 2 (Plain free-riding)**.** Assuming FedAvg converges in expectation, and based on the assumption of Theorem 1, the following asymptotic properties hold:

$$\mathbb{E}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} 0, \tag{14}$$

$$\operatorname{Var}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} \frac{[\frac{1}{N^2} + \frac{1}{(N-M_K)^2}] \sum_{j \in J} \left(M_j \rho_j\right)^2}{1 - \left(\epsilon + \frac{M_K}{N}\right)^2}. \tag{15}$$

As a corollary of Theorem 2, in Proof A.2 it is shown that the asymptotic variance is strictly increasing with the sample size M_K declared by the free-riders. In practice, the smaller the total number of data points declared by the free-riders, the closer the final aggregation result approaches the model obtained with fair clients only. On the contrary, when the the sample size of the fair clients is negligible with respect to the the one declared by the free-riders, i.e. $N \simeq M_K$, the variance tends to infinity. This is due to the ratio approaching to 1 in the geometric sum of the second term of equation (13). In the limit case when only free-riders participate to federated learning ($J = \emptyset$), we obtain instead the trivial result $\tilde{\theta}^t = \theta^0$ and $\operatorname{Var}\left[\tilde{\theta}^t\right] = 0$. In this case there is no learning throughout the training process. Finally, with no free-riders ($M_K = 0$), we obtain $\operatorname{Var}\left[\tilde{\theta}_1^t - \theta_2^t\right] \xrightarrow{t \to +\infty} \frac{2}{N^2} \frac{1}{1-\epsilon^2} \sum_{j \in J} \left(M_j \rho_j\right)^2$, reflecting the variability of the fair aggregation process due to the stochasticity of the local optimization processes.

2.5 Disguised free-riding

Plain free-riders can be easily detected by the server, since for each iteration the condition $[\tilde{\theta}_k^{t+1} - \tilde{\theta}^t = 0]$ is true. In what follows, we study improved attack strategies based on the sharing of opportunely disguised parameters, and investigate sufficient conditions on the disguising models to obtain the desired convergence behavior of free-rider attacks.

2.5.1 Additive noise to mimic SGD updates

A disguised free-rider with additive noise generalizes the plain one, and uploads parameters $\tilde{\theta}_k^{t+1} = \tilde{\theta}^t + \varphi_k(t)\epsilon_t$. Here, the perturbation ϵ_t is assumed to be Gaussian white noise, and $\varphi_k(t) > 0$ is a suitable time-varying perturbation compatible with the free-rider attack. As shown in equation (8), the parameters uploaded by the fair clients take the general form composed of an expected model corrupted by a stochastic perturbation due to SGD. Free-riders can mimic this update form by adopting a noise structure similar to the one of the fair clients:

$$\varphi_k^2(t) = \frac{\lambda}{S} {\sigma_k^t}^2 \frac{1}{2r_k} \left[1 - e^{-2\lambda r_k \frac{EM_k}{S}}\right], \tag{16}$$

where r_k and σ_k^t would ideally depend on the (non-existing) free-rider data distribution and thus need to be determined, while M_k is the declared number of samples. Compatibly with the assumptions of constant SGD variance σ_j^2 for the fair clients, we here assume that the free-riders noise is constant and compatible with the SGD form:

$$\varphi_k^2 = \frac{\lambda}{S} \sigma_k^2 \frac{1}{2r_k} \left[1 - e^{-2\lambda r_k \frac{EM_k}{S}}\right]. \tag{17}$$

The parameters r_k and σ_k affect the noise level and decay of the update, and thus the ability of the free-rider of mimicking a realistic client. These parameters can be ideally estimated by computing a plausible quadratic approximation of the local loss function (Section 2.3). While the estimation may require the availability of some form of data for the free-rider, in Section 2.5.2 we prove that, for any combination of r_k and σ_k, federated learning still converges to the desired aggregated target.

Analogously as for the fair clients, this assumption will be relaxed in Section 2.5.3.

2.5.2 Attacks based on fixed additive stochastic perturbations

In this new setting, we can rewrite the FedAvg aggregation process (1) for an attack with a single free-rider with perturbation φ:

$$\tilde{\theta}^{t+1} = \sum_{j \in J} \frac{M_j}{N} \tilde{\theta}_j^{t+1} + \frac{M_K}{N} \tilde{\theta}^t + \frac{M_K}{N} \varphi \epsilon_t. \tag{18}$$

Theorem 3 extends the results previously obtained for federated learning with plain free-riders to our new case with additive perturbations.

Theorem 3 (Single disguised free-rider)**.** Analogously to Theorem 2, the aggregation process under free-riding described in equation (18) converges in expectation to the aggregated model of the fair clients of equation (1) :

$$\mathbb{E}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} 0, \tag{19}$$

$$\begin{aligned}\mathrm{Var}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} & \frac{[\frac{1}{N^2} + \frac{1}{(N-M_K)^2}] \sum_{j \in J} (M_j \rho_j)^2}{1 - \left(\epsilon + \frac{M_K}{N}\right)^2} \\ & + \frac{1}{1 - \left(\epsilon + \frac{M_K}{N}\right)^2} \frac{M_K^2}{N^2} \varphi^2. \end{aligned} \tag{20}$$

Theorem 3 shows that disguised free-riding converges to the final model of federated learning with fair clients, although with a higher variance resulting from the free-rider's perturbations injected at every iteration. The perturbation is proportional to $\frac{M_K}{N}$, the relative number of samples declared by the free-rider.

The extension of this result to the case of multiple free-riders requires to account in equation (18) for an attack of the form $\sum_{k \in K} \frac{M_k}{N} \varphi_k \epsilon_{k,t}$, where M_k is the total sample size declared by free-rider k. Corollary 1 follows from the linearity of this form.

Corollary 1 (Multiple disguised free-riders)**.** Assuming a constant perturbation factor φ_k for each free-rider k, the asymptotic expectation of Theorem 3 still holds, while the variance reduces to

$$\begin{aligned}\mathrm{Var}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} & \frac{[\frac{1}{N^2} + \frac{1}{(N-M_K)^2}] \sum_{j \in J} (M_j \rho_j)^2}{1 - \left(\epsilon + \frac{M_K}{N}\right)^2} \\ & + \frac{1}{1 - \left(\epsilon + \frac{M_K}{N}\right)^2} \sum_{k \in K} \frac{M_k^2}{N^2} \varphi_k^2. \end{aligned} \tag{21}$$

2.5.3 Time-varying noise model of fair-clients evolution

To investigate more plausible parameters evolution in federated learning, in this section we relax the assumption made in Section 2.3 about the constant noise perturbation of the SGD process across iteration rounds.

We assume here that the standard deviation σ_j^t of SGD decreases at each server iteration t, approaching to zero over iteration rounds: $\sigma_j^t \xrightarrow{t \to +\infty} 0$. This assumption reflects the improvement of the fit of the global model $\tilde{\theta}^t$ to the local datasets over server iterations, and implies that the stochastic process of the local optimization of Section 2.3 has noise parameter $\rho_j^t \xrightarrow{t \to +\infty} 0$. We thus hypothesize that, to mimic the behavior of the fair clients, a suitable time-varying perturbation of the free-riders should follow a similar asymptotic behavior: $\varphi_k(t) \xrightarrow{t \to +\infty} 0$. Under these assumptions, Corollary 2 shows that the asymptotic variance of model aggregation under free-rider attacks is zero, and that it is thus still possible to retrieve the fair client's model.

Corollary 2. Assuming that fair clients and free-riders evolve according to Section 2.3 to 2.5, if the conditions $\rho_j^t \xrightarrow{t \to +\infty} 0$ and $\varphi_k(t) \xrightarrow{t \to +\infty} 0$ are met, the aggregation process of federated learning is such that the asymptotic variance of Theorems 2 and 3 reduce to

$$\mathrm{Var}\left[\tilde{\theta}^t - \theta^t\right] \xrightarrow{t \to +\infty} 0. \tag{22}$$

We assumed in Corollary 2 that the SGD noise σ_j^t decreases at each server iteration and eventually converges to 0. In practice, the global model may not fit perfectly the dataset of the different clients $\mathcal{D}_j$ and, after a sufficient number of optimization rounds, may keep oscillating around a local minima. We could therefore assume that $\sigma_j^t \xrightarrow{t \to +\infty} \sigma_j$ leading to $\rho_j^t \xrightarrow{t \to +\infty} \rho_j$. In this case, to mimic the behavior of the fair clients, a suitable time-varying perturbation compatible with the free-rider attacks should converge to a fixed noise level such that $\varphi_k(t) \xrightarrow{t \to +\infty} \varphi_k$. Similarly as for Corollary 2, it can be shown that under these hypothesis federated learning follows the asymptotic behaviors of Theorem 2 and 3 for respectively plain and disguised free-riders.

2.6 FedProx

FedProx includes a regularization term for the local loss functions of the different clients ensuring the proximity between the updated models θ_j^{t+1} and θ^t. This regularization is usually defined as an additional L2 penalty term, and leads to the following form for the local gradient $g_j(\theta_j) \simeq r_j[\theta_j - \theta_j^*] + \mu[\theta_j - \theta^t]$ where μ is a trade-off parameter. Since the considerations in Section 2.3 still hold in this setting, we can express the local model contribution for FedProx with a formulation analogous to the one of equation (8). Hence, for FedProx, we obtain similar conclusions for Theorem 2 and 3, as well as for Corollary 1 and 2, proving that the convergence behavior with free-riders is equivalent to the one obtained with fair clients only, although with a different asymptotic variance (Appendix B).

Theorem 4. Assuming convergence in expectation for federated learning with fair clients only, under the assumptions of Theorem 1 the asymptotic properties of plain and disguised free-riding of Theorem 2, 3, and Corollary 1, 2, still hold with FedProx. In this case we have parameters:

$${\rho_j}^2 = \frac{\lambda}{S} {\sigma_j}^2 \frac{1}{2(r_j + \mu)} \left[1 - e^{-2\lambda(r_j + \mu)\frac{EM_j}{S}}\right], \tag{23}$$

$$\epsilon = \sum_{j \in J} \frac{M_j}{N} [\gamma_j + \mu \frac{1 - \gamma_j}{r_j + \mu}], \tag{24}$$

$$\text{and } \gamma_j = e^{-\lambda(r_j + \mu)\frac{EM_j}{S}}. \tag{25}$$

We note that the asymptotic variance is still strictly increasing with the total number of free-riders samples. Moreover,

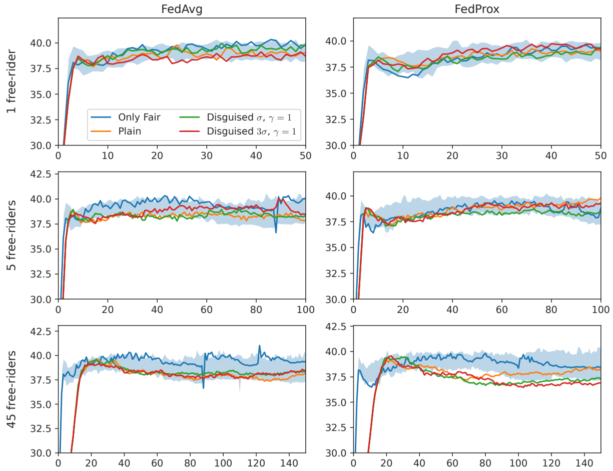

Figure 1: Plots for Shakespeare and $E = 20$. Accuracy performances for FedAvg and FedProx according to the number of free-riders participating in the learning process: 15% (top), 50% (middle), and 90% (bottom) of the total amount of clients. The shaded blue region indicates the variability of federated learning model with fair clients only, estimated from 30 different training initialization.

the regularization term monitors the asymptotic variance: a higher regularization leads to a smaller noise parameter ρ_j^2 and to a smaller ϵ, thus decreasing the asymptotic variances of Theorem 2, 3, and Corollary 1, 2.

3 Experiments

This experimental section focuses on a series of benchmarks for the proposed free-rider attacks. The methods being of general application, the focus here is to empirically demonstrate our theory on diverse experimental setups and model specifications. All code, data and experiments are available at `https://github.com/Accenture/Labs-Federated-Learning/tree/free-rider_attacks`.

3.1 Experimental Details

We consider 5 fair clients for each of the following scenarios, investigated in previous works on federated learning [McMahan et al., 2017, Li et al., 2018]:

MNIST (classification in iid and non-iid settings). We study a standard classification problem on MNIST [LeCun et al., 1998] and create two benchmarks: an iid dataset (MNIST iid) where we assign 600 training digits and 300 testing digits to each client, and a non-iid dataset (MNIST non-iid), where for each digit we create two shards with 150 training samples and 75 testing samples, and allocate 4 shards for each client. For each scenario, we use a logistic regression predictor.

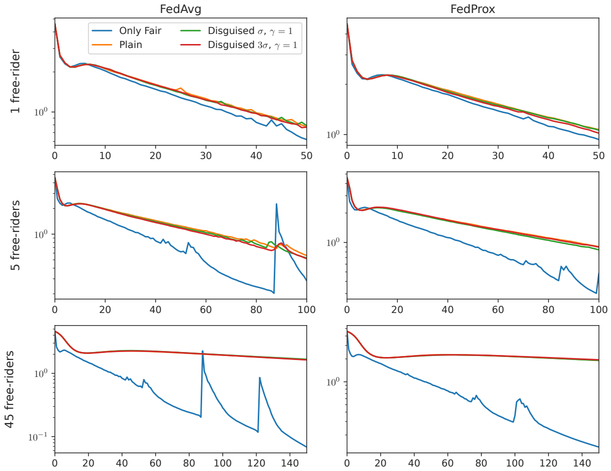

Figure 2: Plots for Shakespeare and $E = 20$. Loss performances for FedAvg and FedProx according to the number of free-riders participating in the learning process: 15% (top), 50% (middle), and 90% (bottom) of the total amount of clients.

CIFAR-10[Krizhevsky et al.,] (image classification). The dataset consists of 10 classes of 32x32 images with three RGB channels. There are 50000 training examples and 10000 testing examples which we partitioned into 5 clients each containing 10000 training and 2000 testing samples. The model architecture was taken from [McMahan et al., 2017] which consists of two convolutional layers and a linear transformation layer to produce logits.

Shakespeare (LSTM prediction). We study a LSTM model for next character prediction on the dataset of *The Complete Works of William Shakespeare* [McMahan et al., 2017]. We randomly chose 5 clients with more than 3000 samples, and assign 70% of the dataset to training and 30% to testing. Each client has on average 6415.4 samples (± 1835.6) . We use a two-layer LSTM classifier containing 100 hidden units with an 8 dimensional embedding layer. The model takes as an input a sequence of 80 characters, embeds each of the characters into a learned 8-dimensional space and outputs one character per training sample after 2 LSTM layers and a fully connected one.

We train federated models following FedAvg and FedProx aggregation processes. In FedProx, the hyperparameter μ monitoring the regularization is chosen according to the best performing scenario reported in [Li et al., 2018]: $\mu = 1$ for MNIST (iid and non-iid), and $\mu = 0.001$ for Shakespeare. For the free-rider we declare a number of samples equal to the average sample size across fair clients. We test federated learning with 5 and 20 local epochs using SGD optimization with learning rate $\lambda = 0.001$ for MNIST (iid and non-iid), $\lambda = 0.001$ for CIFAR-10, and $\lambda = 0.5$ for Shakespeare, and batch size of 100. We evaluate the success of the free-rider attacks by quantifying testing accuracy and training loss of the resulting model, as indicators of the effect of the perturbation induced by free-riders on the final model performances. Resulting figures for associated accuracy and loss can be found in Figure 1, Figure 2 and Appendix C.

3.2 Free-rider attacks: convergence and performances

In the following experiments, we assume that free-riders do not have any data, which means that they cannot estimate the noise level by computing a plausible quadratic approximation of the local loss function (Section 2.5). Therefore, we investigate free-rider attacks taking the simple form $\varphi(t) = \sigma t^{-\gamma}$. The parameter γ is chosen among a panel of testing parameters $\gamma \in \{0.5, 1, 2\}$, while additional experimental material on the influence of γ on the convergence is presented in Appendix C. While the optimal tuning of disguised free-rider attacks is out of the scope of this study, in what follows the perturbations parameter σ is defined according to practical hypotheses on the parameters evolution during federated learning. After random initialization at the initial federated learning step, the parameter σ is opportunely estimated to mimic the extent of the distribution of the update $\Delta\tilde{\theta}^0 = \tilde{\theta}^1 - \tilde{\theta}^0$ observed between consecutive rounds of federated learning. We can simply model these increments as a zero-centered univariate Gaussian distribution, and assign the parameter σ to the value of the fitted standard deviation. According to this strategy, the free-rider would return parameters $\tilde{\theta}_k^t$ with perturbations distributed as the ones observed between two consecutive optimization rounds. Figure 1, top row, exemplifies the evolution of the models obtained with FedAvg (20 local training epochs) on the Shakespeare dataset with respect to different scenarios: 1) fair clients only, 2) plain free-rider, 3) disguised free-rider with decay parameter $\gamma = 1$, and estimated noise level σ, and 4) disguised free-rider with noise level increased to 3σ. For each scenario, we compare the federated model obtained under free-rider attacks with respect to the equivalent model obtained with the participation of the fair clients only. For this latter setting, to assess the model training variability, we repeated the training 30 times with different parameter initializations. The results show that, independently from the chosen free-riding strategy, the resulting models attains comparable performances with respect to the one of the model obtained with fair clients only (Figure 1, top row). Similar results are obtained for the setup with 5 local training epochs and different values of γ, as well as for FedProx with 5 and 20 local epochs (Appendix C).

We also investigate the same training setup under the influence of multiple free-riders (Figure 1, mid and bottom rows). In particular, we test the scenarios where the free-riders declare respectively 50% and 90% of the total training sample size. In practice, we maintain the same experimental setting composed of 5 fair clients, and we increase the number of free-riders to respectively 5 and 45, while declaring for each free-rider a sample size equal to the average number of samples of the fair clients. Independently from the magnitude of the perturbation function, the number of free-riders does not seem to affect the performance of the final aggregated model. However, the convergence speed is greatly decreased. Figure 2 shows that the convergence in these different settings is not identically affected by the free-riders. When the size of free-riders is moderate, e.g. up to 50% of the total sample size, the convergence speed of the loss is slightly slower than for federated learning with fair clients. The attacks can be still considered successful, as convergence is achieved within the pre-defined iteration budget. However, when the size of free-riders reaches 90%, convergence to the optimum is extremely slow and cannot be achieved anymore in a reasonable amount of iterations. This result is in agreement with our theory, for which the convergence speed inversely proportional to the relative size of the free-riders. Interestingly, we note that the final accuracy obtained in all the scenarios is similar (though a bit slower with 90% of free-riders), and falls within the variability observed in federated learning with fair-clients only (Figure 1). This result is achieved in spite of the incomplete convergence during training. This effect can be explained by observing that this accuracy level is already reached at the early training stages of federated learning with fair clients, while further training does not seem to improve the predictions. This result suggests that, in spite of the very low convergence speed, the averaging process with 90% of free-riders still achieves a reasonable minima compatible with the training path of the fair clients aggregation.

We note that the "peaks" observed in the loss of Figure 2 are common in FL, especially in the considered application when the number of clients is low. It is important to notice that our experiments are performed by using vanilla SGD. As such, the peaks for only fair clients are to be expected in both loss and performances. We also notice that the peaks are smaller for free-riding because of the “regularization” effect of free-riders, which regresses the update towards the global model of the previous iteration.

Analogous results and considerations can be derived from the set of experiments on the remaining datasets, training parameters and FedProx as an aggregation scheme (Appendix C).

4 Conclusion and discussion

We introduced a theoretical framework for the study of free-riding attacks on model aggregation in federated learning. Based on the proposed methodology, we proved that simple strategies based on returning the global model at each iteration already lead to successful free-rider attacks (plain free-riding), and we investigated more sophisticated disguising techniques relying on stochastic perturbations of the parameters (disguised free-riding). The convergence of each attack was demonstrated through theoretical developments and experimental results. The threat of free-rider attacks is still under-investigated in machine learning. For example, current defence schemes in federated learning are mainly based on outliers detection mechanisms, to detect malicious attackers providing abnormal updates. These schemes would be therefore unsuccessful in detecting a free-rider update which is, by design, equivalent to the global federated model.

This work opens the way to the investigation of optimal disguising and defense strategies for free-rider attacks, beyond the proposed heuristics. Our experiments show that inspection of the client's distribution should be established as a routine practice for the detection of free-rider attacks in federated learning. Further research directions are represented by the improvement of detection at the server level, through better modeling of the heterogeneity of the incoming clients' parameters. This study provides also the theoretical basis for the study of effective free-riding strategies, based on different noise model distributions and perturbation schemes. Finally, in this work we relied on a number of hypothesis concerning the evolution of the clients' parameters during federated learning. This choice provides us with a convenient theoretical setup for the formalization of the proposed theory which may be modified in the future, for example, for investigating more complex forms of variability and schemes for parameters aggregation.

Acknowledgments and Disclosure of Funding

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002, and by the ANR JCJC project Fed-BioMed 19-CE45-0006-01. The project was also supported by Accenture. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

[Ateniese et al., 2015] Ateniese, G., Mancini, L. V., Spognardi, A., Villani, A., Vitali, D., and Felici, G. (2015). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150.

[Bhagoji et al., 2019] Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. (2019). Analyzing federated learning through an adversarial lens. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:1012–1021.

[Brisimi et al., 2018] Brisimi, T., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I., and Shi, W. (2018). Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 112.

[Carlini et al., 2019] Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. (2019). The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, Santa Clara, CA. USENIX Association.

[Fredrikson et al., 2015] Fredrikson, M., Jha, S., and Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA. Association for Computing Machinery.

[Fredrikson et al., 2014] Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. (2014). Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 17–32, San Diego, CA. USENIX Association.

[Fung et al., 2020] Fung, C., Yoon, C. J. M., 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)*, pages 301–316, San Sebastian. USENIX Association.

[He et al., 2018] He, L., Meng, Q., Chen, W., Ma, Z. M., and Liu, T. Y. (2018). Differential equations for modeling asynchronous algorithms. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July(1):2220–2226.

[Hitaj et al., 2017] Hitaj, B., Ateniese, G., and Perez-Cruz, F. (2017). Deep Models under the GAN: Information leakage from collaborative deep learning. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 603–618.

[Krizhevsky et al.,] Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research).

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Ha, P. (1998). LeNet. *Proceedings of the IEEE*, (November):1–46.

[Li et al., 2016] Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. (2016). Data poisoning attacks on factorization-based collaborative filtering. *Advances in Neural Information Processing Systems*, (Nips):1893–1901.

[Li et al., 2017] Li, Q., Tai, C., and Weinan, E. (2017). Stochastic modified equations and adaptive stochastic gradient algorithms. *34th International Conference on Machine Learning, ICML 2017*, 5:3306–3340.

[Li et al., 2018] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2018). Federated Optimization in Heterogeneous Networks. *Proceedings of the 1 st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019*, pages 1–28.

[Lin et al., 2019] Lin, J., Du, M., and Liu, J. (2019). Free-riders in Federated Learning: Attacks and Defenses. *http://arxiv.org/abs/1911.12560*.

[Mandt et al., 2017] Mandt, S., Hof Fman, M. D., and Blei, D. M. (2017). Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18:1–35.

[McMahan et al., 2017] McMahan, H., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 54.

[Orvieto and Lucchi, 2018] Orvieto, A. and Lucchi, A. (2018). Continuous-time Models for Stochastic Optimization Algorithms. (NeurIPS).

[Shen et al., 2016] Shen, S., Tople, S., and Saxena, P. (2016). AUROR: Defending against poisoning attacks in collaborative deep learning systems. In *ACM International Conference Proceeding Series*, volume 5-9-Decemb, pages 508–519.

[Silva et al., 2019] Silva, S., Gutman, B. A., Romero, E., Thompson, P. M., Altmann, A., and Lorenzi, M. (2019). Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 270–274. IEEE.

[Uhlenbeck and Ornstein, 1930] Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841.

[Wang et al., 2019] Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., and Qi, H. (2019). Beyond Inferring Class Representatives: User-Level Privacy Leakage from Federated Learning. *Proceedings - IEEE INFOCOM*, 2019-April:2512–2520.

[Xie et al., 2019] Xie, C., Huang, K., Chen, P.-Y., and Li, B. (2019). Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*.

[Yin et al., 2018] Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *35th International Conference on Machine Learning, ICML 2018*, 13:8947–8956.