What else is leaked when eavesdropping Federated Learning?

Chuan Xu, Giovanni Neglia

What else is leaked when eavesdropping Federated Learning?

Chuan Xu
chuan.xu@inria.fr
Univ. Côte d'Azur, Inria, CNRS, I3S

Giovanni Neglia
giovanni.neglia@inria.fr
Inria, Univ. Côte d'Azur

Abstract

In this paper, we initiate the study of *local model reconstruction attacks* for federated learning, where a honest-but-curious adversary eavesdrops the messages exchanged between the client and the server and reconstructs the local model of the client. The success of this attack enables better performance of other known attacks, such as the membership attack, attribute inference attacks, etc. We provide analytical guarantees for the success of this attack when training a linear least squares problem with full batch size and arbitrary number of local steps. One heuristic is proposed to generalize the attack to other machine learning problems. Experiments are conducted on logistic regression tasks, showing high reconstruction quality, especially when clients' datasets are highly heterogeneous (as it is common in federated learning).

1 Introduction

Federated learning (FL) [1–3] offers naturally a certain level of privacy, as clients' data is not collected at a third party. However, maintaining the data locally does not provide itself formal privacy guarantees. An (honest-but-curious) adversary can still infer some sensitive client information just by eavesdropping the exchanged messages (e.g., gradients). In fact, multiple recent works have shown the possibility to reconstruct private data samples (e.g., images) by inverting the gradients [4–8]. This attack works well when gradients are calculated on extremely small batches or data points belonging to the same class are similar, e.g., personal images of the same person or images of the same digit in MNIST dataset. However, this attack may not apply to FL settings, especially when clients are allowed to do multiple stochastic gradient updates locally to save significant communication costs. In [7, Table 6a] the authors show that the success rate of a reconstruction attack degrades from 100% to 39% when the number of local steps increases from 1 to 9, and degrades from 100% to 13% when the batch size increases from 1 to 16. Similarly, anther study [8] shows that the best attack by inverting the gradients can identify just 28% of ImageNet images with batch size equal to 48. In addition to the reconstruction attack, other attacks like membership attack and property inference attack (the adversary can infer when a property appears or disappears in the data during training) are also studied for FL [9]. Furthermore, these attacks are less effective as batch size increases [9, Table II], and the application of differential private techniques does not change the result [10, Figure 1,25].

In this paper, we initiate the study of a new attack, called the *local model reconstruction attack*, where the adversary seeks to reconstruct the model a client would have trained using only its local dataset. Allowing the adversary to have such information is dangerous, as the adversary can then target various types of personal information using model inversion attacks [11, 12], membership attacks [13], attribute inference attacks [14–16] [1], etc. Again, we assume a *weak* adversary, who is honest-but-curious (i.e., it does not interfere with the training process) and only eavesdrops the exchanged messages between the client and the server. The adversary knows the structure of the trained model and the loss function, as well as the training algorithm, which is common in the attacks for FL [6–9, 17]. Differently from the attacks proposed

[1] These attacks aim to learn hidden sensitive attributes of a data instance for which non-sensitive attributes are known to the public.

in literature, a local model reconstruction attack benefits from a larger batch size and is less sensitive to the number of local steps in FL.

First, we show analytically that, when training a linear least squares problem with full batch size and arbitrary number of local steps in FL, the adversary can reconstruct the *exact* local model of every client, just by eavesdropping the exchanged messages for $\Theta(d)$ number of rounds, where d is the number of the features in the data sample (Sec. 3.1). Second, we propose a heuristic to perform this attack on any machine learning problem (Sec. 3.2). Empirical results show that our heuristic works well for logistic regression problems (Sec. 4).

2 Motivation

We denote by $\mathcal{C}$ the set of all clients participating to FL. Let $\mathcal{D}_c$ be the local dataset of client $c \in \mathcal{C}$ drawn from a universe $\mathcal{X}$ and $|\mathcal{D}_c|$ be the size of $\mathcal{D}_c$. In FL, clients cooperate to learn a global model, which minimizes the following (weighted) empirical risk over all the data owned by clients:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta), \quad \text{where} \quad \mathcal{L}(\theta) = \sum_{c \in \mathcal{C}} p_c \mathcal{L}_c(\theta) = \sum_{c \in \mathcal{C}} p_c \left(\frac{1}{|\mathcal{D}_c|} \sum_{x \in \mathcal{D}_c} l(\theta, x) \right), \tag{1}$$

where $l(\theta, x) : \mathbb{R}^d, \mathcal{X} \to \mathbb{R}_+$ measures the loss of the model θ on the sample $x \in \mathcal{X}$ and p_c is the positive weight of client c, s.t. $\sum_{c \in \mathcal{C}} p_c = 1$.

Let $\theta^* = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta)$ be the global optimal model, i.e., the true minimizer of problem (1). The global model is the objective of FL training, and could be the target from where the adversary starts different attacks, such as membership attacks [13], model inversion attacks [12], and attribute inference attacks [14–16]. However, applying these attacks on the global model suffers from the plausible deniability of the clients, as the global model does not capture identity information of data instances (i.e., to which client each data belongs). Moreover, due to the heterogeneity of the data distribution, the global model may not perform well on individual local data, which makes these attacks less accurate.

In this paper, we propose the idea that the adversary can perform the attacks mentioned above not on the global model learned through FL, but on the local optimal model θ_c^*, i.e., $\theta_c^* = \arg\min \mathcal{L}_c(\theta)$. Then, the adversary needs first to estimate such a local model; this first step is the *local model reconstruction attack* we focus on in this paper. The local model reconstruction attack benefits from the local model overfitting the client's dataset and is more effective under non-i.i.d. clients' data distributions, which is exactly the case in FL. In FL, the degree of non-i.i.d can be quantified by the dissimilarity between the global optimal model and the local optimal model. In [18], the authors define the degree as $\sum_{c \in \mathcal{C}} p_c(\mathcal{L}_c(\theta^*) - \mathcal{L}_c(\theta_c^*))$. In [19, 20], the authors define the dissimilarity degree as $\sum_{c \in \mathcal{C}} p_c(\nabla \mathcal{L}_c(\theta^*) - \nabla \mathcal{L}_c(\theta_c^*))$.

3 Local model reconstruction

To start, we present a general framework for FL in Algo. 1, which generalizes a majority of the FL algorithms, including FedAvg [1], FedProx [3], and FL with different client sampling techniques [21–23]. The model $\tilde{\theta} = \theta(T)$ in the FL framework is the final output for problem (1). Its performance depends on the applied FL algorithm, which precises how the clients are selected in line 2, how the updated local models are aggregated in line 5 and how the local update rule works in line 8.

Our attack targets any FL algorithm falling into the framework presented in Algo. 1. Notice that this FL framework does **not** consider the possible use of secure aggregation protocols [24, 25], which allow the server to aggregate the local updates without having access to each individual update. Secure aggregation protocols prevent our adversary from decoding the local model of a specific client, as the identity information is lost after secure aggregation. However, they require significant additional computation, which makes these protocols hard to scale to a large system when training a large neural network. In this paper, we consider the case where no secure aggregation protocol is applied. Remember that the known attacks for FL where

Algorithm 1 Framework for cross-device federated learning

Output: $\theta(T)$

Server: *// global model* $\theta \in \mathbb{R}^d$*, local models* $\{\theta_c \in \mathbb{R}^d, \forall c \in \mathcal{C}\}$.

for $t \in \{0, ..., T-1\}$ **do**
  Server selects a subset of the clients $\mathcal{C}_s(t) \subseteq \mathcal{C}$,
  Server broadcasts the current global model $\theta(t)$ to $\mathcal{C}_s(t)$,
  Server waits for the updated local models θ_c from every client $c \in \mathcal{C}_s(t)$,
  Server updates $\theta(t+1)$ by aggregating the received updated local models.

Client $c \in \mathcal{C}$: *// global model* θ*, local model* θ_c*, local dataset* $\mathcal{D}_c$

while FL training is not completed **do**
  Client listens for the arrival of new global model θ,
  Client updates its local model: $\theta_c \leftarrow \text{Local\_Update}^c(\theta, \mathcal{D}_c)$
  Client sends back θ_c to the server.

the clients are honest, such as gradient reconstruction attack [4–8] and inference attack [17], are all against FL without secure aggregation protocols.[2]

3.1 Exact model reconstruction for linear least squares regression in FedAvg

Here we show the possibility to reconstruct the exact local model in polynomial time when training a least squares linear regression through FedAvg [1] with full batch size and arbitrary number of local steps. The local update rule of FedAvg is given in Algo. 2. The procedure for the exact reconstruction is shown in the proof of Observation 1.

Algorithm 2 Client c's local update rule in FedAvg [1]

$\text{Local\_Update}^c(\theta, \mathcal{D}_c)$ *// θ: server model, $\mathcal{D}_c$: local dataset, B: batch size, E: the number of local epochs, η: learning rate.*

$\theta_c \leftarrow \theta$, $\mathcal{B} \leftarrow$ (split $\mathcal{D}_c$ into batches of size B)
for each local epoch e from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $\theta_c \leftarrow \theta_c - \eta \times \mathbf{g}(\theta_c, b)$, where $\mathbf{g}(\theta_c, b) = \frac{1}{B} \sum_{x \in b} \nabla l(\theta_c, x)$
Return θ_c

Observation 1. *Consider training a least squares linear regression through FedAvg with full batch size and assume that a client's design matrix has rank d equal to the number of features. Once the client has communicated with the server $d+1$ times, the adversary can recover the client's local optimal model in $O(d^3)$ operations.*

Proof. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ be the design matrix with rank d and $\mathbf{y} \in \mathbb{R}^m$ be the labels in the local dataset $\mathcal{D}_c$ of the client c. Let m be the number of the local samples, i.e., $m = |\mathcal{D}_c|$. The local loss function of client c is:

$$\mathcal{L}_c(\theta) = \frac{\|\mathbf{X}\theta - \mathbf{y}\|^2}{m} \tag{2}$$

Let $\mathbf{H} = \mathbf{X}^T\mathbf{X}$, we know that $\theta_c^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. When the batch size is set to m in FedAvg:

$$\mathbf{g}(\theta) = \frac{2}{m}\left(\mathbf{H}\theta - \mathbf{H}\theta_c^*\right). \tag{3}$$

[2] In [9], the authors claim that their property attack can succeed even in presence of secure aggregation protocols, but the attack can detect when a certain property appears during training, not at at which client.

At round t, if client c is selected, it receives the server model and executes Algo. 2. Let $\theta_c^t(e)$ be the model after the e-th local epoch's update. Replacing (3) with line 4 in Algo. 2, we have

$$\begin{aligned}\theta_c^t(E) &= \theta_c^t(E-1) - \frac{2\eta}{m}\left(\mathbf{H}\theta_c^t(E-1) - \mathbf{H}\theta_c^*\right) \\ &= (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})\theta_c^t(E-1) + \frac{2\eta}{m}\mathbf{H}\theta_c^* \\ &= (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})^E\theta_c^t(0) + \left[\mathbf{I} - (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})^E\right]\theta_c^*.\end{aligned}$$

Let $\mathbf{W} = \left[\mathbf{I} - (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})^E\right]$ and $\mathbf{v} = \mathbf{W}\theta_c^*$. We have

$$\theta_c^t(0) - \theta_c^t(E) = \left[\mathbf{I} - (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})^E\right]\theta_c^t(0) - \left[\mathbf{I} - (\mathbf{I} - \frac{2\eta}{m}\mathbf{H})^E\right]\theta_c^* \tag{4}$$

$$= \mathbf{W}\theta_c^t(0) - \mathbf{v}. \tag{5}$$

Note that $\theta_c^t(0)$ is the server model and $\theta_c^t(E)$ is model returned from client c to server in Algo. 2. The adversary has access to both of them as it can eavesdrop messages exchanged between the server and the client. Since $\mathbf{W} \in \mathbb{R}^{d\times d}$ and $\mathbf{v} \in \mathbb{R}^d$, once the adversary gets $d+1$ exchanged messages, it can reconstruct the exact matrix $\mathbf{W}$ and the vector $\mathbf{v}$ by solving d systems (one for each row of $\mathbf{W}$ and the corresponding element of $\mathbf{v}$ in (5)), each with $d+1$ linear equations (one for each communication round). Solving a system of $d+1$ linear equations requires $O(d^3)$ operations, thus solving d systems in parallel requires only $O(d^3)$ operations. Since $\mathbf{H}$ is positive definite, $\mathbf{W}$ is invertible. Then, we can compute $\theta_c^* = \mathbf{W}^{-1}\mathbf{v}$. Note that in this reconstruction process, the adversary does not require knowledge of the parameters in FedAvg, such as the learning rate and the number of local steps. □

Observation 2. *Consider training a least squares linear regression through FedAvg and assume that a client's design matrix has rank d equal to the number of features. For any FL algorithm where a client's local update rule can be seen as a first-order black box* [3]*, at least one client is required to communicate with the server $\Omega(d)$ times for the convergence to the optimum. In addition, to reconstruct the local optimal model of this client, the adversary must eavesdrop $\Omega(d)$ times.*

Proof. To prove the lower bound for the communications, we construct the specific "hard" scenario. This scenario is inspired by the work for studying the lower bound for convergence of smooth convex function [26, Sect. 2.1.2].

Let $\mathbf{c}$ be the client having the local dataset $(\mathbf{X}, \mathbf{y})$ such that $\mathbf{H} = \mathbf{X}^T\mathbf{X}$ is a tridiagonal matrix with $\mathbf{H}_{i,i+1} = \mathbf{H}_{i+1,i} = -1/2$ and $\mathbf{H}_{i,i} = 1, \forall i \in \{1, 2, ..., d\}$, and $\mathbf{X}^T\mathbf{y} = [1/2, 0, 0, ..., 0]^T$. By substituting $\mathbf{H}$ and $\mathbf{X}^T\mathbf{y}$ into (2), we have

$$\begin{aligned}\mathcal{L}_\mathbf{c}(\theta) &= \frac{1}{m}\left(\theta^T\mathbf{H}\theta - 2(\mathbf{X}^T\mathbf{y})^T\theta - \mathbf{y}^T\mathbf{y}\right) \\ &= \frac{1}{m}\left(\sum_{i=1}^{d}\theta_{[i]}^2 - \sum_{i=1}^{d-1}\theta_{[i]}\theta_{[i+1]} - \theta_{[1]} - C\right),\end{aligned} \tag{6}$$

where C is a constant and $\theta_\mathbf{c}^* = \arg\min \mathcal{L}_\mathbf{c}(\theta) = \mathbf{H}^{-1}\mathbf{X}^T\mathbf{y} = [1 - \frac{1}{d+1}, 1 - \frac{2}{d+1}, ..., 1 - \frac{d}{d+1}]^T$. According to (6), we know that if $\theta_{[i]} = \theta_{[i+1]} = \ldots = \theta_{[d]} = 0$, then $\nabla\mathcal{L}_\mathbf{c}(\theta)_{[i+1]} = \ldots = \nabla\mathcal{L}_\mathbf{c}(\theta)_{[d]} = 0$.

At the same time, for the other clients $\forall \bar{c} \in \mathcal{C} \setminus \mathbf{c}$, we assume their local optimum are zeros and their $\mathbf{H} \in \mathbb{R}^{d\times d}$ are identity matrices, i.e., $\theta_{\bar{c}}^* = \mathbf{0}$ and $\mathbf{H} = \mathbf{I}$. Under this setting, according to Observation 1, we

[3] For every client c, its local model $\theta_c(t)$ can be expressed as

$$\theta_c(t) = \theta_c(0) + \mathbf{span}\{\nabla\mathcal{L}_c(\theta_c(0)), \nabla\mathcal{L}_c(\theta_c(1)), ..., \nabla\mathcal{L}_c(\theta_c(t-1))\}.$$

know that if the i^{th} element of the global model is zero, i.e., $\theta(t)_{[i]} = 0$, then $\theta_c^t(E)_{[i]} = 0$. Moreover, it is reasonable to assume that every element of the global optimum is non-zero, i.e., $\theta^*_{[i]} \neq 0, \forall i \in \{1, .., d\}$.

Now, suppose that we run the FL algorithm (Algorithm 1) under the above scenario, with initial global model $\theta(0) = \mathbf{0}$, i.e., $\theta_c^0(0) = \mathbf{0}, \forall c \in \mathcal{C}$. According to the previous analysis, we know that $\forall t \in \{0, 1..., d-1\}$:

$$\theta_{\mathbf{c}}^t(E)_{[t+1]} = \theta_{\mathbf{c}}^t(E)_{[t+2]} = \ldots = \theta_{\mathbf{c}}^t(E)_{[d]} = 0, \tag{7}$$

and

$$\theta(t)_{[t+1]} = \theta(t)_{[t+2]} = \ldots = \theta(t)_{[d]} = 0. \tag{8}$$

Therefore, to reach the non-zeros global optimum, the client $\mathbf{c}$ needs to communicate with the server (be selected by the server) for at least d times.

Moreover, to recover the local optimum of client $\mathbf{c}$, the adversary must listen on the communication channel for at least d times. Suppose that the client $\mathbf{c}$ holds another local data set which gives $\mathbf{H}'$ equals to $\mathbf{H}$ except that the last row and the last column are zeros. Under this case, the adversary will have the same observation as for the case of $\mathbf{H}$ till round $d-1$. □

Combining Observation 1 and Observation 2, we can conclude that the reconstruction algorithm presented in Observation 1 for least squares linear regression training through FedAvg, is **optimal** for local model reconstruction attack, in terms of the number of the communications that the adversary eavesdrops.

3.2 Heuristic for local model reconstruction

In this section, we propose a heuristic (Algo. 3) for a local model reconstruction attack suited for any algorithm falling into the FL framework (Algo. 1) and any machine learning problem.

Let $\mathcal{T}^c$ be the indexes of the rounds at which client c has been selected by the server, i.e., $\mathcal{T}^c = \{t | c \in \mathcal{C}_s(t), \forall t \in \{0, \ldots, T\}\}$. We denote by $\mathcal{M}^c$ the messages exchanged in Algo. 1 between client c and the server, i.e., $\mathcal{M}^c = \{(\theta(t), \theta_c(t)), \forall t \in \mathcal{T}^c\}$. Our approach to (approximately) reconstruct the local model of client c consists of two steps. First, the adversary learns a mapping function $\mathcal{G}^c$ to mimic the local update rule of the client, by exploring the messages in $\mathcal{M}^c$ (see (9) in Algo. 3). More precisely, given the server model θ as input, $\mathcal{G}^c$ predicts the update difference $\Delta\theta = \theta - \theta_c$. Second, the adversary estimates the local model of client c as the one which minimizes $\|\mathcal{G}^c(\theta)\|^2$ (see (10) in Algo. 3). The intuition behind this step is that the client does not update the model ($\Delta\theta = 0$) once it has reached the local optimal model.

When executing FedAvg using full batch size, the exact mapping function $\mathcal{G}^c$ can be correctly estimated for least squares linear regression, and is shown to be linear (see (5) in the proof of Observation 1). In this case, minimizing $\|\mathcal{G}^c\|^2$ corresponds to solving the linear systems mentioned in the proof. We can then conclude that the decoding algorithm in Observation 1 can be seen as a particular instance of Algo. 3 and, in particular, leads to an exact local model reconstruction as proved above.

Generally speaking, the performance of our heuristic depends on two effects: 1) How well the mapping function $\mathcal{G}^c$ mimics the behavior of the local update rule integrated in FL algorithm, 2) How close is the estimated $\hat{\theta}$ to the true minimizer of problem (10), when $\mathcal{G}^c$ is non-convex.

The first effect depends on the complexity of the learning task, the randomness introduced by the local update rule in the FL algorithm (line 8 in Algo. 1), and the number of messages observed by the adversary. More randomness is introduced, less accurate is the estimation of $\mathcal{G}^c$, and then the final local model reconstruction. This corresponds to the fact that differentially private algorithms, which amplify the randomness in the original algorithm by adding noise, sub-sampling [27, 28], reshuffling [29, 30], etc., can better protect private information from attacks. The second effect depends on the convexity of problem (10), and then on the structure of $\mathcal{G}^c$.

Algorithm 3 Local model reconstruction attack

// Input: $\mathcal{M}^c$, the messages exchanged between client c and the server

1: $\Delta\mathcal{M}^c = \{(\theta, \Delta\theta = \theta - \theta_c), \forall(\theta, \theta_c) \in \mathcal{M}^c\}$

2: Define a mapping function $\mathcal{G}^c$ with parameters $w \in \mathbb{R}^m$, taking the server model $\theta \in \mathbb{R}^d$ as input and predicting the difference between server model and local model, i.e., $\mathcal{G}^c : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^d$.

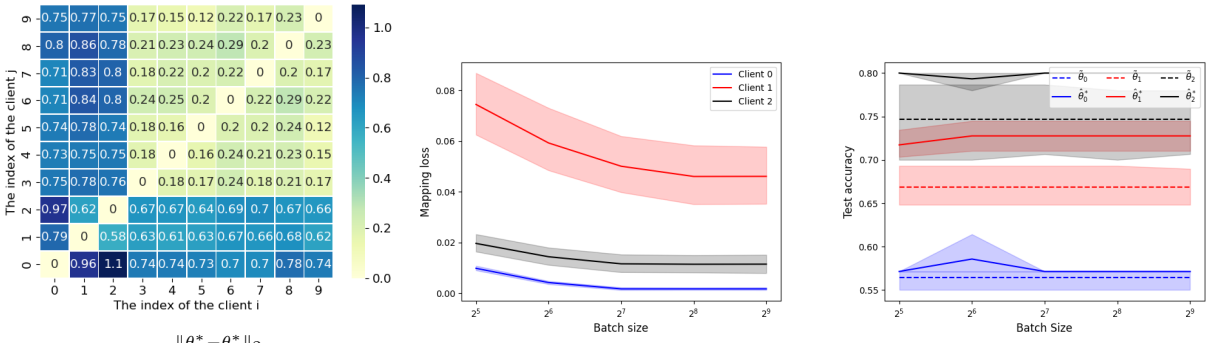
3: Estimate $\mathcal{G}^c$ parameters, denoted by $\hat{w}$, by minimizing the empirical risk:

$$\min_{w \in \mathbb{R}^m} \sum_{(\theta, \Delta\theta) \in \Delta\mathcal{M}^c} \|\mathcal{G}^c(w, \theta) - \Delta\theta\|^2 \tag{9}$$

4: Estimate the model $\hat{\theta}_c$, by minimizing the local update difference $\mathcal{G}^c$ for fixed parameters $\hat{w}$:

$$\min_{\theta \in \mathbb{R}^d} \|\mathcal{G}^c(\hat{w}, \theta)\|^2 \tag{10}$$

5: Return $\hat{\theta}^*_c$ as the estimator for the local model of client c



(a) $\frac{\|\theta^*_i - \theta^*_j\|_2}{\|\theta^*_j\|_2}$ (b) Mapping loss vs batch size (c) Test accuracy vs batch size

Figure 1: Adult, one local step and $\eta = 0.001$. Figs 1b and 1c show the mean and the 95% confidential interval over 10 runs.

4 Experiment

We have evaluated our proposed heuristic (Algo. 3) on two logistic regression tasks in the federated learning setting: LEAF synthetic [31] and Adult [32] (detailed below). For the structure of the mapping function $\mathcal{G}^c$, we use a simple neural network with one hidden layer of size 1000 followed by ReLu activation. Adam optimizer is used for solving the problems (9) and (10).

LEAF synthetic This dataset is designed for classification, where clients' data sets are highly heterogeneous both in terms of number of samples and underlying statistical distribution. The detailed generation procedure can be found in [31, Appendix A]. We set the number of features to 10, the number of the classes to 2, the number of clusters (determining statistical heterogeneity) to 5, and the number of the clients to 5. The batch size is set to 256. The numbers of samples for each client are 280, 184, 1536, 256 and 208 respectively.

Adult This dataset contains individual information such as sex, age, education level, family situation, working class, etc. This information is used to predict whether a person has an income higher than 50k$, which can be seen as sensitive information. We perform our attack on a subset of the data where the

individual's education level is at least "bachelor". [4] There are 10 clients. To simulate a non-iid data distribution scenario, we distribute the records of people with a PhD degree among the first three clients according to their age. The first client owns the data of young PhDs less than 38 years old, the second client owns the data of PhDs aging between 38 and 52 years old, and the third client owns the data of PhDs elder than 52 years old. The numbers of training samples for the first three clients are 126, 258 and 134, respectively. The rest of the data is uniformly distributed among the remaining clients. To show the dissimilarity between clients, the relative Euclidean distance between each client's local optimum model is evaluated (See Fig. 1a). We can observe that, due to our specific non-iid data distribution, the local models of the first three clients are quite far from the rest of the local models, which is reasonable as people with PhD degree are more likely to have a different salary prediction pattern.

Local steps	Model θ_c	$Acc_c(\theta_c)$ of client c (%)					$\frac{\sum_c Acc_c(\theta_c)}{5}$
		c=0	c=1	c=2	c=3	c=4	
1	$\tilde{\theta}_c$	53.9	49.7	85.1	28.3	75.8	58.6
	$\hat{\theta}_c^*$	67.0	74.3	80.6	77.9	91.0	**78.1**
5	$\tilde{\theta}_c$	60.1	54.6	87.6	33.6	79.2	63.0
	$\hat{\theta}_c^*$	65.6	72.9	76.9	72.5	86.2	**74.8**
10	$\tilde{\theta}_c$	69.4	60.4	90.6	42.3	83.9	69.3
	$\hat{\theta}_c^*$	67.5	79.5	78.1	75.8	88.9	**78.0**

Table 1: Average train accuracy on client c's local dataset of the final personalized model of FedAvg $\tilde{\theta}_c$ and the decoded model $\hat{\theta}_c^*$ over 10 independent runs. LEAF synthetic dataset with 5 clients, batch size 256 and learning rate 0.01.

Performance To evaluate the performance of the decoded local model $\hat{\theta}_c^*$ obtained from Algo. 3, we consider as a baseline the last model $\tilde{\theta}_c$ returned by the client c at the end of training (instead of the final global model $\tilde{\theta}$). The last returned model is potentially more susceptible to attacks than the final global model, as it contains more personal information. For every attack scenario, 10 independent runs were conducted with different seeds.

In Table 1, we show the average training accuracy of $\tilde{\theta}_c$ and $\hat{\theta}_c^*$ over 10 runs on each client's local dataset, under LEAF synthetic data distribution. Note that larger training accuracy values suggest the model is more suited to the local dataset and it may then leak more personal information. As the number of local steps increases, the accuracy of the last returned model increases as the model has been obtained through more local updates. We can see from the last column that, on average, our attack outperforms the baseline with 10%-20% improvement. The performance of the local model reconstruction attack is almost insensitive to the number of local steps. Moreover, this attack is more effective on client 3, whose local dataset is probably very different from the others as suggested by the low accuracy of the model $\tilde{\theta}_c$.

In Figure 1b, we show the performance of the mapping function $\mathcal{G}^c$ when attacking one of the first three client c, under different batch size scenarios in FL for Adult dataset. Small batch size would introduce more randomness into the data set $\Delta\mathcal{M}^c$. Thus, the mapping loss (Eq. 9) is higher for small batch size. Figure 1c shows that the decoded local model outperforms the baseline (dash line) for test accuracy as well.

5 Conclusion

In this paper, we initiate the study of a new attack for FL training: the adversary estimates a client's local/personalized model which may reveal private information and open the road to other classical model-based attacks. In comparison to state-of-the-art attacks on the global model, our attack reduces the plausible deniability and does not suffer the problem of large batch size (on the contrary, it works better the larger the

[4] The number of the data points are reduced from 48842 to 12300.

batch size). There is still space for the improvement of our local model reconstruction attack and it would be interesting to evaluate its performance on neural networks which is one of our future research directions.

Acknowledgements

This work has been supported by the French government, through the UCAJEDI and UCA DS4H Investments in the Future projects managed by the National Research Agency (ANR) with the reference number ANR-15-IDEX-0001 and ANR-17-EURE-0004.

References

[1] 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*, pages 1273–1282. PMLR, 2017.

[2] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5330–5340, 2017.

[3] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *MLSys*, 2020.

[4] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14774–14784, 2019.

[5] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *CoRR*, abs/2001.02610, 2020.

[6] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *NIPS*, 2020.

[7] Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*, 2020.

[8] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[9] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[10] Lixin Fan, KamWoh Ng, Ce Ju, Tianyu Zhang, Chang Liu, Chee Seng Chan, and Qiang Yang. Rethinking privacy preserving deep learning: How to evaluate and thwart privacy attacks. In Qiang Yang, Lixin Fan, and Han Yu, editors, *Federated Learning - Privacy and Incentive*, volume 12500 of *Lecture Notes in Computer Science*, pages 32–50. Springer, 2020.

[11] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

[12] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[13] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[14] Shiva Prasad Kasiviswanathan, Mark Rudelson, and Adam Smith. The power of linear reconstruction attacks. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1415–1433. SIAM, 2013.

[15] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[16] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1895–1912, 2019.

[17] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE, 2019.

[18] 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.

[19] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *Neurips Workshop*, 2019.

[20] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR, 2020.

[21] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[22] Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *Workshop in NeurIPS 2020: Privacy Preserving Machine Learning*, 2020.

[23] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

[24] 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*, CCS '17, page 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery.

[25] Swanand Kadhe, Nived Rajaraman, O Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020.

[26] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[27] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, pages 6277–6287, 2018.

[28] Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1226–1235. PMLR, 2019.

[29] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 332–349. IEEE, 2019.

[30] Sajin Sasy and Olga Ohrimenko. Oblivious sampling algorithms for private data analysis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 6495–6506. Curran Associates, Inc., 2019.

[31] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[32] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.