Attribute Inference Attacks for Federated Regression Tasks

Francesco Diana[1, 2], **Othmane Marfoq**[3], **Chuan Xu**[1, 2, 4, 5], **Giovanni Neglia**[1, 2], **Frédéric Giroire**[1, 2, 4, 5], **Eoin Thomas**[6]

[1]Université Côte d'Azur
[2]Inria
[3]Meta
[4]CNRS
[5]I3S
[6]Amadeus

{francesco.diana, chuan.xu, giovanni.neglia, frederic.giroire}@inria.fr, omarfoq@meta.com, eoin.thomas@amadeus.com

Abstract

Federated Learning (FL) enables multiple clients, such as mobile phones and IoT devices, to collaboratively train a global machine learning model while keeping their data localized. However, recent studies have revealed that the training phase of FL is vulnerable to reconstruction attacks, such as attribute inference attacks (AIA), where adversaries exploit exchanged messages and auxiliary public information to uncover sensitive attributes of targeted clients. While these attacks have been extensively studied in the context of classification tasks, their impact on regression tasks remains largely unexplored. In this paper, we address this gap by proposing novel model-based AIAs specifically designed for regression tasks in FL environments. Our approach considers scenarios where adversaries can either eavesdrop on exchanged messages or directly interfere with the training process. We benchmark our proposed attacks against state-of-the-art methods using real-world datasets. The results demonstrate a significant increase in reconstruction accuracy, particularly in heterogeneous client datasets, a common scenario in FL. The efficacy of our model-based AIAs makes them better candidates for empirically quantifying privacy leakage for federated regression tasks.

Code — https://github.com/francescodiana99/fedkit-learn

1 Introduction

Federated learning (FL) enables multiple clients to collaboratively train a global model (McMahan et al. 2017; Lian et al. 2017; Li et al. 2020). Since clients' data is not collected by a third party, FL naturally offers a certain level of privacy. Nevertheless, FL alone does not provide formal privacy guarantees, and recent works have demonstrated that clients' private information can still be easily leaked (Lyu et al. 2020; Liu, Xu, and Wang 2022). For instance, an adversary with access to the exchanged messages and knowledge of some public information (e.g., client's provided ratings) (Lyu and Chen 2021; Chen et al. 2022; Feng et al. 2021) can reconstruct a client's sensitive attributes (e.g., gender/religion) in an attack known as attribute inference attack (AIA). Additionally, the adversary can reconstruct client's training samples such as images (Geiping et al. 2020; Yin et al. 2021).

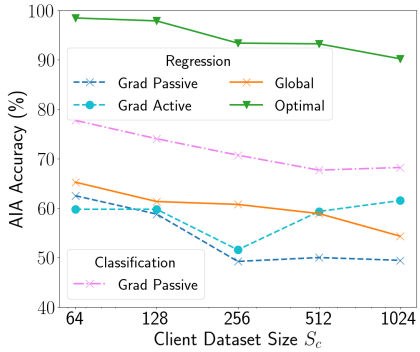

Figure 1: Average performance of different AIAs when four clients train a neural network through FedAvg with 1 local epoch and batch size 32. Each client stores S_c data points randomly sampled from ACS Income dataset (Ding et al. 2024). The adversary infers the gender attribute of every data sample held by the client given access to the released (public) information.

However, these reconstruction attacks for FL have primarily been tested on classification tasks and have not been explored for *regression* tasks, which are, needless to say, equally important for practical applications. Quite surprisingly, our experiments, as shown in Fig. 1, demonstrate that the accuracy of state-of-the-art gradient-based AIA under a honest-but-curious adversary (Lyu and Chen 2021; Chen et al. 2022) (referred to as passive) drops significantly from 71% on a classification task to 50% (random guess) on a regression task once the targeted client holds more than 256 data points. Furthermore, even a more powerful (active) adversary capable of forging the messages to the targeted client to extract more information (Lyu and Chen 2021; Chen et al. 2022) offers only limited improvement to the AIA performance on regression tasks. Detailed information about this experiment is in (Diana et al. 2024, Appendix B.4).

In this paper, we show that federated training of regression tasks does not inherently enjoy higher privacy, but it is simply more vulnerable to other forms of attacks. While existing

FL AIA attacks for classification tasks are gradient-based (see Sec. 2.3), we show that model-based AIAs—initially proposed for centralized training (Fredrikson et al. 2014; Kasiviswanathan, Rudelson, and Smith 2013; Yeom et al. 2018)—may be more effective for regression tasks. Figure 1 illustrates that a model-based attack on the server's global model (i.e., the final model trained through FL) already performs at least as well as the SOTA gradient-based passive attack. Moreover, it highlights that even more powerful attacks (up to 30 p.p. more accurate) could be launched if the adversary had access to the optimal local model of the targeted client (i.e., a model trained only on the client's dataset).

Motivated by these observations, we propose a new two-step model-based AIA for *federated regression tasks*. In this attack, the adversary first (approximately) reconstructs the client's optimal local model and then applies an existing model-based AIA to that model.

Our main contributions can be summarized as follows:

- We provide an analytical lower bound for model-based AIA accuracy in the least squares regression problem. This result motivates the adversary's strategy to approximate the client's optimal local model in federated regression tasks (Sec. 3).
- We propose methods for approximating optimal local models where adversaries can either eavesdrop on exchanged messages or directly interfere with the training process (Sec. 4).
- Our experiments show that our model-based AIAs are better candidates for empirically quantifying privacy leakage for federated regression tasks (Sec. 5).

2 Preliminaries

2.1 Federated Learning

We denote by $\mathcal{C}$ the set of all clients participating to FL. Let $\mathcal{D}_c = \{(\mathbf{x}_c(i), y_c(i)), i = 1, ..., S_c\}$ denote the local dataset of client $c \in \mathcal{C}$ with size $S_c \triangleq |\mathcal{D}_c|$. Each data sample $(\mathbf{x}_c(i), y_c(i))$ is a pair consisting of an input $\mathbf{x}_c(i)$ and of an associated target value $y_c(i)$. In FL, clients cooperate to learn a global model, which minimizes the following empirical risk over all the data owned by clients:

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) = \sum_{c \in \mathcal{C}} p_c \mathcal{L}_c(\theta, \mathcal{D}_c) = \sum_{c \in \mathcal{C}} p_c \left(\frac{1}{S_c} \sum_{i=1}^{S_c} \ell(\theta, \mathbf{x}_c(i), y_c(i)) \right). \quad (1)$$

where $\ell(\theta, \mathbf{x}_c(i), y_c(i))$ measures the loss of the model θ on the sample $(\mathbf{x}_c(i), y_c(i)) \in \mathcal{D}_c$ and p_c is the positive weight of client c, s.t., $\sum_{c \in \mathcal{C}} p_c = 1$. Common choices of weights are $p_c = \frac{1}{|\mathcal{C}|}$ or $p_c = \frac{S_c}{\sum_{c \in \mathcal{C}} S_c}$.

Let $\theta^* = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta)$ be a global optimal model, i.e., a minimizer of Problem (1). A general framework to learn such a global model in a federated way is shown in Alg. 1; it generalizes a large number of FL algorithms, including FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), and FL with different client sampling techniques (Nishio and Yonetani 2019; Chen, Horvath, and Richtarik 2020; Jee Cho, Wang, and Joshi 2022). The model θ^T—the output of Alg. 1—is the tentative solution of Problem (1). Its performance depends on the specific FL algorithm, which precises how 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. For example, in FedAvg, clients are selected uniformly at random among the available clients, the local models are averaged with constant weights, and the clients perform locally multiple stochastic gradient steps (McMahan et al. 2017).

Algorithm 1: FL Framework

Output: θ^T

Server s:

1: **for** $t \in \{0, ..., T-1\}$ **do**
2: s selects a subset of the clients $\mathcal{C}^t \subseteq \mathcal{C}$,
3: s broadcasts the global model θ^t to $\mathcal{C}^t$,
4: s waits for the updated models θ_c^t from every client $c \in \mathcal{C}^t$,
5: s computes θ^{t+1} by aggregating the received updated models.

Client $c \in \mathcal{C}$: Input θ, Output θ_c

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

2.2 Threat Model

Honest-but-Curious Adversary. We describe first an honest-but-curious adversary,[1] which is a standard threat model in existing literature (Melis et al. 2019; Geiping et al. 2020; Yin et al. 2021; Nasr, Shokri, and Houmansadr 2019), including the FL one (Kairouz et al. 2021, Table 7), (Lyu and Chen 2021; Chen et al. 2022). This *passive* adversary, who could be the server itself, is knowledgeable about the trained model structure, the loss function, and the training algorithm, and may eavesdrop on communication between the attacked client and the server but does not interfere with the training process. For instance, during training round t, the adversary can inspect the messages exchanged between the server and the attacked client (denoted by c), allowing him to recover the parameters of the global model θ^t and the updated client model $\theta_c^t(K)$. Let $\mathcal{T}_c \subseteq \{t | c \in \mathcal{C}^t, \forall t \in 0, \ldots, T-1\}$ denote the set of communication rounds during which the adversary inspects messages exchanged between the server and the attacked client and $\mathcal{M}_c = \{(\theta^t, \theta_c^t), \forall t \in \mathcal{T}_c\}$ denote the corresponding set of messages.

When it comes to defenses against such an adversary, it is crucial to understand that traditional measures like encrypted communications are ineffective if the attacker is the FL server. More sophisticated cryptographic techniques like secure aggregation protocols (Bonawitz et al.

[1] In what follows, we refer to the client using female pronouns and the adversary using male pronouns, respectively.

2017; Kadhe et al. 2020) allow the server to aggregate local updates without having access to each individual update and, then, do hide the client's updated model θ_c^t from the server. Nevertheless, they come with a significant computation overhead (Quoc et al. 2020) and are inefficient for sparse vector aggregation (Kairouz et al. 2021). Moreover, they are vulnerable to poisoning attacks, as they hinder the server from detecting (and removing) potentially harmful updates from malicious clients (Blanchard et al. 2017; Yin et al. 2018; El Mhamdi 2020). For instance, Tramèr et al. (2022, Sec. 4.4) introduce a new class of data poisoning attacks that succeed when training models with secure multiparty computation. Alternatively, Trusted Execution Environments (TEEs) (Sabt, Achemlal, and Bouabdallah 2015; Singh et al. 2021) provide an encrypted memory region to ensure the code has been executed faithfully and privately. They can then both conceal clients' updated models and defend against poisoning attacks. However, implementing a reliable TEE platform in FL remains an open challenge due to the infrastructure resource constraints and the required communication processes needed to connect verified codes (Kairouz et al. 2021).

Malicious Adversary. We also consider a stronger *active* adversary who can interfere with the training process. Specifically, this adversary can modify the messages sent to the clients and have the clients update models θ^t that have been concocted to reveal more private information. Let $\mathcal{T}_c^a$ be the set of rounds during which the adversary attacks client c by sending a malicious model θ^t. As above, the adversary could be the server itself. This adversary has been widely considered in the literature on reconstruction attacks (Wen et al. 2022; Boenisch et al. 2023) and membership inference attacks (Nguyen et al. 2023; Nasr, Shokri, and Houmansadr 2019). Some studies have also explored the possibility of a malicious adversary modifying the model architecture during training (Fowl et al. 2022; Zhao et al. 2023), even though such attacks appear to be easily detectable. In this paper, we do not allow the adversary to modify the model architecture. For simplicity, we will refer to these two adversaries as passive and active adversaries, respectively, throughout the rest of the paper.

2.3 Attribute Inference Attack (AIA) for FL

AIA leverages public information to deduce private or sensitive attributes (Kasiviswanathan, Rudelson, and Smith 2013; Fredrikson et al. 2014; Yeom et al. 2018; Lyu and Chen 2021; Chen et al. 2022). For example, an AIA could reconstruct a user's gender from a recommender model by having access to the user's provided ratings. Formally, each input $\mathbf{x}_c(i)$ of client c consists of public attributes $\mathbf{x}_c^p(i)$ and of a sensitive attribute $s_c(i)$. The target value, assumed to be public, is denoted by $y_c^p(i)$. The adversary, having access to $\{(\mathbf{x}_c^p(i), y_c^p(i)), i = 1, ..., S_c\}$ and $\mathcal{M}_c$, aims to recover the sensitive attributes $s_c(i)$.[2]

[2]In (Fredrikson et al. 2014; Yeom et al. 2018), the adversary possesses additional information, including estimates of the marginals or the joint distribution of the data samples. However, in this paper, we do not consider such a more powerful adversary.

Existing Gradient-Based AIA for FL. Lyu and Chen (2021) and Chen et al. (2022) present AIAs specifically designed for the FL context and both passive and active adversaries. The central idea involves identifying sensitive attribute values that yield virtual gradients closely resembling the client's model updates—referred to as *pseudo-gradients*—in terms of cosine similarity. Formally, the adversary solves the following optimization problem:

$$\underset{\{s_c(i)\}_{i=1}^{S_c}}{\operatorname{argmax}} \sum_{t \in \mathcal{T}} \textbf{CosSim}\left(\frac{\partial \ell(\theta^t, \{(\mathbf{x}_c^p(i), s_c(i), y_c^p(i))\})}{\partial \theta^t}, \theta^t - \theta_c^t\right), \quad (2)$$

where $\mathcal{T} \subseteq \mathcal{T}_c$ for a passive adversary and $\mathcal{T} \subseteq \mathcal{T}_c \cup \mathcal{T}_c^a$ for an active adversary. The active adversary simply sends back to the targeted client c her own model $\theta_c^{(t-1)}$ at each attack round in $\mathcal{T}_c^a$.

Lyu and Chen (2021) and Chen et al. (2022) assume that the sensitive attributes are categorical. Nevertheless, problem (2) can be solved efficiently using a gradient method with the reparameterization trick and the Gumbel softmax distribution (Jang, Gu, and Poole 2017). From (2), we observe that, since gradients incorporate information from all samples, the attack performance deteriorates in the presence of a large local dataset. For example, the attack accuracy almost halves on the Genome dataset for the classification task when the client's local dataset size increases from 50 to 1000 samples (Lyu and Chen 2021, Table 8). Our experiment (Figure 1) on a regression task corroborates this finding: when the local dataset size increases from 64 to 256, the attack accuracy drops from 60% to the level of random guessing.

Model-Based AIA. As an alternative, the AIA can be executed directly on the model (rather than on the model pseudo-gradients), as initially proposed for centralized training in (Kasiviswanathan, Rudelson, and Smith 2013; Fredrikson et al. 2014). Given a model θ, the adversary can infer the sensitive attributes by solving the following optimization problems:

$$\underset{s_c(i)}{\operatorname{argmin}}\, \ell(\theta, (\mathbf{x}_c^p(i), s_c(i), y_c^p(i))), \quad \forall i \in \{1, ..., S_c\}, \quad (3)$$

Below we provide theoretical guarantees (Prop. 1) for the accuracy of model-based AIAs to least squares regression problems. Our theoretical result corroborates that, in an FL setting, an adversary can benefit by first estimating the client's optimal local model—that is, the model that minimizes the client's empirical loss.

3 Model-Based AIA Guarantees for Least Squares Regression

In this section, we provide novel theoretical guarantees for the accuracy of the model-based AIA (Problem (3)) in the context of least squares regression. In particular, we show that the better the model θ fits the local data and the more the sensitive attribute affects the final prediction, the higher the AIA accuracy.

Proposition 1. *Let E_c be the mean square error of a given least squares regression model θ on the local dataset of client c and $\theta[s]$ be the model parameter corresponding to a*

binary sensitive attribute. The accuracy of the model-based AIA (3) *is larger than or equal to* $1 - \frac{4E_c}{\theta[s]^2}$.

Proof. Let $\mathbf{s}_c$ be the vector including all the unknown sensitive binary attributes $\{s_c(i), \forall i \in \{1, ..., S_c\}\}$ of client c. Let $\mathbf{x}_c \in \mathbb{R}^{S_c \times d}$ be the design matrix with rank d and $\mathbf{y}_c \in \mathbb{R}^{S_c}$ be the labels in the local dataset $\mathcal{D}_c$ of the client c. Let $\theta[:p] \in \mathbb{R}^{d-1}$ be the parameters corresponding to the public attributes. The adversary has access to partial data instances in $\mathcal{D}_c$ which consists of the public attributes $\mathbf{P} \in \mathbb{R}^{S_c \times (d-1)}$ and the corresponding labels $\mathbf{y}_c \in \mathbb{R}^{S_c}$.

The goal for the adversary is to decode the values of the binary sensitive attribute $\mathbf{s}_c \in \{0,1\}^{S_c}$ given $(\mathbf{P}, \mathbf{y}_c)$ by solving (3), i.e., checking for each point, which value for the sensitive attribute leads to a smaller loss.

It is easy to verify that the problem can be equivalently solved through the following two-step procedure. First, the adversary computes the vector of real values:

$$\tilde{\mathbf{s}}_c = \underset{\mathbf{s}_c \in \mathbb{R}^{S_c}}{\operatorname{argmin}} ||\mathbf{P}\theta[:p] + \mathbf{s}_c\theta[s] - \mathbf{y}_c||_2^2 = \frac{\mathbf{y}_c - \mathbf{P}\theta[:p]}{\theta[s]}.$$

Then, the adversary reconstruct the vector of sensitive features $\hat{\mathbf{s}}_c \in \{0,1\}^{S_c}$ as follows

$$\hat{s}_c(i) = \begin{cases} 0 & \text{if } \tilde{s}_c(i) < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}, \quad \forall i \in \{1, ..., S_c\}. \tag{4}$$

Let $\mathbf{e}$ be the vector of residuals for the local dataset, i.e., $\mathbf{e}_c = \mathbf{y}_c - (\mathbf{P}\theta[:p] + \mathbf{s}_c\theta[s])$. We have then

$$\mathbf{s}_c = \frac{\mathbf{y}_c - \mathbf{P}\theta[:p] - \mathbf{e}_c}{\theta[s]} = \tilde{\mathbf{s}}_c - \frac{\mathbf{e}_c}{\theta[s]}. \tag{5}$$

Let us say that the sensitive feature of sample i has been erroneously reconstructed, i.e., $s_c(i) \neq \hat{s}_c(i)$, then (4), implies that $|s_c(i) - \tilde{s}_c(i)| \geq 1/2$, and from (5) it follows that $|e_c(i)|^2 \geq \theta[s]^2/4$. As $E_c S_c = \|\mathbf{e}_c\|_2^2$, there can be at most $4E_c S_c/\theta[s]^2$ samples erroneously reconstructed, from which we can conclude the result. □

Proposition 1 indicates that the model-based AIA performs better the more the model overfits the dataset. This observation justifies the following two-step attack in a FL setting: *the adversary first reconstructs the optimal local model of the targeted client c, i.e.,* $\theta_c^* = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_c(\theta, \mathcal{D}_c)$*, and then execute the AIA in* (3) *on the reconstructed model.*[3] In the following, we will detail our approaches for reconstructing the optimal local model by passive and active adversary, respectively.

4 Reconstructing the Local Model in FL

In this section, we show how an adversary may reconstruct the optimal local model of client c, i.e., $\theta_c^* = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_c(\theta, \mathcal{D}_c)$.

[3]Note that this model is optimal in terms of the training error, but not in general of the final test error. On the contrary, it is likely to overfit the local dataset.

Algorithm 2: Reconstruction of client-c local model by a passive adversary for federated least squares regression

Input: the server models $\theta^{t_i}(0) = \theta_c^{t_i}(0)$ and the local updated models $\theta_c^{t_i}(K)$ at all the inspected rounds $t_i \in \mathcal{T}_c = \{t_1, t_2, \ldots, t_{n_c}\}$.

1: Let $\Theta_{\text{in}} = [\theta_c^{t_1}(0)\ \theta_c^{t_2}(0)\ ...\ \theta_c^{t_{n_c}}(0)]^T \in \mathbb{R}^{n_c \times d}$

2: Let $\Theta_{\text{out}} = \begin{bmatrix} (\theta_c^{t_1}(0) - \theta_c^{t_1}(K))^T & 1 \\ \vdots & \\ (\theta_c^{t_{n_c}}(0) - \theta_c^{t_{n_c}}(K))^T & 1 \end{bmatrix} \in \mathbb{R}^{n_c \times (d+1)}$

3: $(\hat{\theta}_c^*)^T \leftarrow$ last row of $\left((\Theta_{\text{out}}^T\Theta_{\text{out}})^\dagger \Theta_{\text{out}}^T \Theta_{\text{in}}\right)$

4: Return $\hat{\theta}_c^*$ as the estimator for client c's local model

First, we provide an efficient approach for least squares regression under a passive adversary. We prove that the adversary can exactly reconstruct the optimal local model under deterministic FL updates and provide probabilistic guarantees on the reconstruction error under stochastic FL updates (Sec. 4.1).

Second, we show that an active adversary can potentially reconstruct any client's local model (not just least square regression ones) in a federated setting (Sec. 4.2).

4.1 Passive Approach for Linear Least Squares

We consider that clients cooperatively train a linear regression model with quadratic loss. We refer to this setting as a federated least squares regression. The attack is detailed in Alg. 2 and involves a single matrix computation (line 3) after the exchanged messages $\mathcal{M}_c$ have been collected. $\mathbf{A}^\dagger$ represents the pseudo-inverse of matrix $\mathbf{A}$. Theorem 1 provides theoretical guarantees for the distance between the reconstructed model and the optimal local one, when the model is trained through FedAvg (McMahan et al. 2017) with batch size B and local epochs E. The formal statement of the theorem and its proof are in (Diana et al. 2024, Appendix A.1).

Theorem 1 (Informal statement)**.** *Consider a federated least squares regression with a large number of clients and assume that i) client c has* d*-rank design matrix* $\mathbf{x}_c \in \mathbb{R}^{S_c \times d}$*, ii) she updates the global model through stochastic gradient steps with sub-Gaussian noise with scale* σ*, iii) the global model is independent of previous target client updates, and iv) the eigenvalues of the matrix* $\frac{\Theta_{out}^T \Theta_{out}}{n_c}$ *are lower bounded by* $\underline{\lambda} > 0$*. By eavesdropping on* $n_c > d$ *message exchanges between client c and the server, the error of the reconstructed model* $\hat{\theta}_c^*$ *of Alg. 2 is upper bounded w.p.* $\geq 1-\delta$ *when* $\eta \leq \frac{S_c}{2\lambda_{\max}(\mathbf{x}_c^T \mathbf{x}_c)}$ *and*

$$\left\|\hat{\theta}_c^* - \theta_c^*\right\|_2 = \mathcal{O}\left(\eta\sigma d\sqrt{dE\left\lceil\frac{S_c}{B}\right\rceil \frac{d + 1 + \ln\frac{2d}{\delta}}{n_c \cdot \underline{\lambda}}}\right). \tag{6}$$

When the batch size is equal to the local dataset size, Alg. 2 exactly reconstructs the optimal local model (proof in (Diana et al. 2024, Appendix A.2)):

Proposition 2. *Consider a federated least squares regression and assume that client c has* d*-rank design matrix and*

updates the global model through full-batch gradient updates (i.e., $B = S_c$) and an arbitrary number of local epochs. Once the adversary eavesdrops on $d + 1$ communication exchanges between client c and the server, he can recover the client's optimal local model.

Finally, the following proposition shows that our attack for the full-batch gradient case is **order-optimal** in terms of the number of messages the adversary needs to eavesdrop. The proof is in (Diana et al. 2024, Appendix A.3).

Proposition 3. *Consider that the federated least squares regression is trained through FedAvg with one local step ($E = 1$) and full batch ($B = S_c$ for each client $c \in \mathcal{C}$). At least one client is required to communicate with the server $\Omega(d)$ times for the global model to be learned, and the adversary needs to eavesdrop at least $\Omega(d)$ messages from this client to reconstruct her optimal local model.*

Toy Example Illustration. Here, we illustrate the performance of our Alg. 2 on a toy dataset detailed in (Diana et al. 2024, Appendix B.3). Figure 2 (left) demonstrates that as the batch size increases, the reconstructed model is closer to the optimal local model (as shown in our Theorem 1). Moreover, since the model overfits more the dataset (with smaller loss), the AIA accuracy increases (Figure 2 (right)).

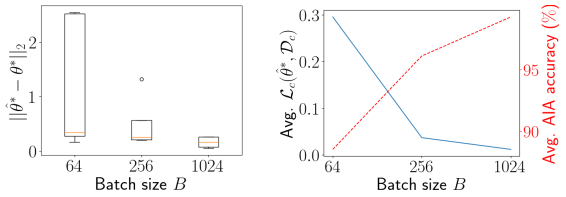

Figure 2: The performance of our passive approach for reconstructing optimal local model (left) and the triggered AIA (right) on a toy dataset with two clients training a linear model with size $d = 11$ under batch size 64, 256 and 1024 for 5 seeds each, respectively. The passive adversary only eavesdropped $d + 1$ messages.

4.2 Active Approach

Here we consider an active adversary (e.g., the server itself) that can modify the global model weights θ^t to recover the client's optimal local model. To achieve this, the adversary can simply send back to the targeted client c her own model $\theta_c^{(t-1)}$ instead of the averaged model θ^t. In this way, client c is led to compute her own optimal local model.

We propose a slightly more sophisticated version of this active attack. Specifically, we suggest that the adversary emulates the Adam optimization algorithm (Kingma and Ba 2015) for the model updated by client c by adjusting the learning rate for each parameter based on the magnitude and history of the gradients, and incorporating momentum. The motivation is twofold. First, the client does not receive back exactly the same model and thus cannot easily detect the attack. Second, Adam is known to minimize the training error faster than stochastic gradient descent at the cost of overfitting more to the training data (Zhou et al. 2020; Zou et al. 2023). We can then expect Adam to help the adversary reconstruct the client's optimal local model better for a given number of modified messages, and our experiments in Sec. 5 confirm that this is the case.

Algorithm 3: Reconstruction of client-c local model by an active adversary a

Input: Let $\mathcal{T}_c^a$ be set of rounds during which the adversary attacks client c and θ_c^a be the corresponding malicious model.

1: $\theta_c^a \leftarrow$ latest model received from client c
2: **for** $t \in \mathcal{T}_c^a$ **do**
3: $\quad a$ sends the model θ_c^a to client c,
4: $\quad a$ waits the updated model from θ_c from client c,
5: $\quad a$ computes the pseudo-gradient $\theta_c^a - \theta_c$ and updates θ_c^a and the corresponding moment vectors following Adam (Kingma and Ba 2015, Alg. 1),
6: Return θ_c^a as the estimator for client c's local model

The details of this attack are outlined in Alg. 3. We observe that the adversary does not need to systematically modify all messages sent to the target client c but can modify just a handful of messages that are not necessarily consecutive. This contributes to the difficulty of detecting the attack.

5 Experiments

5.1 Datasets

Medical (Lantz 2013). This dataset includes 1,339 records and 6 features: age, gender, BMI, number of children, smoking status, and region. The *regression* task is to predict each individual's medical charges billed by health insurance. The dataset is split i.i.d. between 2 clients.

Income (Ding et al. 2024). This dataset contains census information from 50 U.S. states and Puerto Rico, spanning from 2014 to 2018. It includes 15 features related to demographic information such as age, occupation, and education level. The *regression* task is to predict an individual's income. We investigate two FL scenarios, named Income-L and Income-A, respectively. In Income-L, there are 10 clients holding only records from the state of Louisiana (15,982 records in total). These clients can be viewed as the local entities working for the Louisiana State Census Data Center. We examine various levels of statistical heterogeneity among these local entities, with the splitting strategy detailed in (Diana et al. 2024, Appendix B.4). In Income-A, there are 51 clients, each representing a census region and collectively covering all regions. Every client randomly selects 20% of the data points from the corresponding census region, resulting in a total of 332,900 records.

For all the datasets, each client keeps 90% of its data for training and uses 10% for validation.

5.2 FL Training and Attack Setup

In all the experiments, each client follows FedAvg to train a *neural network* model with a single hidden layer of 128 neurons, using ReLU as activation function. The number of communication rounds is fixed to $T = \lceil 100/E \rceil$ where E

is the number of local epochs. Each client participates to all rounds, i.e., $\mathcal{T}_c = \{0, ..., T-1\}$. The learning rate is tuned for each training scenario (different datasets and number of local epochs), with values provided in (Diana et al. 2024, Appendix B.5). The passive adversary may eavesdrop all the exchanged messages until the end of the training. The active adversary launches the attack after T rounds[4] for additional $\lceil 10/E \rceil$ and $\lceil 50/E \rceil$ rounds. Every attack is evaluated over FL trainings from 3 different random seeds. For Medical dataset, the adversary infers whether a person smokes or not. For Income-L and Income-A datasets, the adversary infers the gender. The AIA accuracy is the fraction of the correct inferences over all the samples. We have also conducted experiments for federated least squares regression on the same datasets. The results can be found in (Diana et al. 2024, Appendix C.1).

5.3 Baselines and Our Attack Implementation

Gradient-Based. We compare our method with the (gradient-based) SOTA (Sec. 2.3). The baseline performance is affected by the set of inspected rounds $\mathcal{T}$ considered in (2). We select the inspected rounds $\mathcal{T}$ based on two criteria: the highest cosine similarity and the best AIA accuracy. In a real attack, the adversary is not expected to know the attack accuracy beforehand. Therefore, we refer to the attack based on the highest cosine similarity as **Grad** and to the other as **Grad-w-O** (**Grad**ient **w**ith **O**racle), as it assumes the existence of an oracle that provides the attack accuracy. The details for the tuning of $\mathcal{T}$ and other hyper-parameter settings can be found in (Diana et al. 2024, Appendix B.5).

Our Attacks. Our attacks consist of two steps: 1) reconstructing the optimal local model, and 2) executing the model-based AIA in (3). For the first step, a passive adversary uses the last-returned model from the targeted client, while an active adversary executes Alg. 3. The details of the hyperparameter settings can be found in (Diana et al. 2024, Appendix B.5).

Model-Based with Oracle (Model-w-O). To provide an upper bound on the performance of our approach, we assume the existence of an oracle that provides the optimal local model for the first step of our attack. In practice, the optimal local model is determined empirically by running Adam for a very large number of iterations.

5.4 Experimental Results

From Table 1, we see that our attacks outperform gradient-based ones in both passive and active scenarios across all three datasets. Notably, our passive attack achieves improvements of over 15 and 8 percentage points (p.p.) for the Income-L and Medical datasets, respectively. Even when the gradient-based method has access to an oracle, our passive attack still achieves higher accuracy on two datasets and comes very close on Income-A. When shifting to active attacks, the gains are even more substantial. For instance, when the attack is active for 50 rounds, we achieve gains of 13, 22, and 5 percentage points (p.p.) in Income-L, Income-A, and Medical, respectively, over Grad-w-O, and even larger gains over the more realistic Grad. Furthermore, the attack accuracy reaches the performance expected from an adversary who knows the optimal local model. Interestingly, while our attacks consistently improve as the adversary's capacity increases (moving from a passive attacker to an active one and increasing the number of rounds of the active attack), this is not the case for gradient-based methods.

[4]We also examine the impact of the attack's starting round. The results are presented in (Diana et al. 2024, Appendix C.2)

AIA (%) \ Datasets		Income-L	Income-A	Medical
Passive	Grad	60.36±0.67	54.98±0.29	87.26±0.92
	Grad-w-O	71.44±0.33	**56.10±1.12**	91.06±0.55
	Ours	**75.27±0.32**	55.75±0.17	**95.90±0.04**
Active (10 Rnds)	Grad	60.24±0.60	54.98±0.29	87.26±0.92
	Grad-w-O	80.69±0.55	56.10±1.12	91.06±0.55
	Ours	**82.02±0.85**	**63.53±0.73**	**95.93±0.07**
Active (50 Rnds)	Grad	60.24±0.60	53.36±0.40	87.26±0.92
	Grad-w-O	80.69±0.55	56.12±0.12	91.06±0.55
	Ours	**94.31±0.11**	**78.09±0.25**	**96.79±0.79**
Model-w-O		94.31±0.11	78.31±0.07	96.79±0.79

Table 1: The AIA accuracy over all clients' local datasets evaluated under both honest-but-curious (passive) and malicious (active) adversaries across Income-L, Income-A, and Medical FL datasets (Sec. 5.1). The standard deviation is evaluated over three FL training runs with different random seeds. All clients run FedAvg with 1 epoch and batch size 32. For Income-L, we consider the dataset with 40% of data heterogeneity (Diana et al. 2024, Appendix B.4).

Impact of Data Heterogeneity. We simulate varying levels of heterogeneity in the Income-L dataset and illustrate how the attack performance evolves in Figure 3 (left). First, we observe that as the data is more heterogeneously split, the accuracy of AIA improves for all approaches. Indeed, the data splitting strategy (Diana et al. 2024, Appendix B.4) leads to a greater degree of correlation between the clients' sensitive attributes and the target variable at higher levels of heterogeneity. This intrinsic correlation facilitates the operation of all AIAs. We observe in these experiments, as in previous ones, that active Grad does not necessarily offer advantages over passive Grad. In the more homogeneous case (which is less realistic in an FL setting), our passive attack performs slightly worse than Grad, but its accuracy increases more rapidly with heterogeneity, resulting in an AIA accuracy advantage of over 20 p.p. in the most heterogeneous case. Our active attack over 50 additional rounds consistently outperforms Grad by at least 30 p.p. and is almost indistinguishable from Model-w-O.

Impact of Batch Size. Figure 3 (center) shows that the performance of our attacks slightly decrease as the batch size increases. A possible explanation for this is that a larger batch size results in fewer local updates per epoch, leading the client to return a less overfitted model. Despite this, our approach consistently outperforms Grad in all cases.

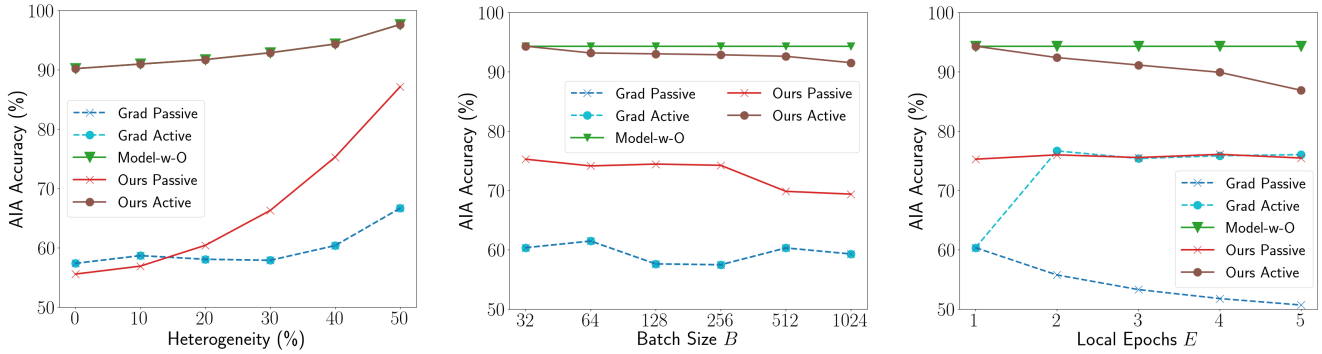

Figure 3: The AIA accuracy over all clients' local datasets under different heterogeneity levels (left) (0% represents i.i.d case), batch sizes (center), and local epochs (right) for Income-L dataset. The default values for heterogeneity level, batch size B and local epochs E are set to 40%, 32, and 1, respectively. The malicious adversary attacks $\lceil 50/E \rceil$ rounds after $\lceil 100/E \rceil$ communication rounds. Crosses represent passive attacks, while dots represent active attacks. Dashed lines correspond to gradient-based attacks (Grad), and solid lines correspond to model-based attacks (Ours and Model-w-O).

Impact of the Number of Local Epochs. Figure 3 (right) shows that under passive attacks our approach is not sensitive to the number of local epochs, whereas Grad's performance deteriorates to the level of random guessing as the number of local epochs increases to 5. Interestingly, active Grad significantly improves upon passive Grad as the number of local epochs increases. While our active approach progressively performs slightly worse, it still maintains an advantage of over 10 p.p. compared to Grad.

Impact of the Local Dataset Size. Figure 4 shows how each client's dataset size impacts individual (active) AIA accuracy in Income-A dataset. We observe that for all methods, clients with smaller datasets are more vulnerable to the attacks, because the models overfit the dataset more easily. Moreover, for clients with over 20000 data points, Grad provides an attack performance close to random guessing.

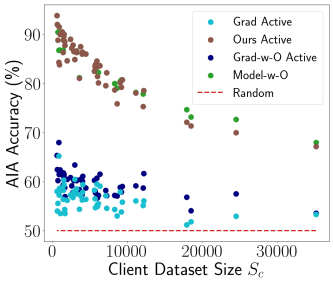

Figure 4: AIA accuracy in Income-A dataset on clients with different local dataset size S_c. The experiment setting is the same as in Table 1 with 50 active rounds.

Impact of defense mechanisms. To mitigate privacy leakage, we apply a federated version of DP-SGD (Abadi et al. 2016), providing sample-level differential privacy guarantees (Dwork et al. 2006; Choudhury et al. 2019). Our experiments show that, even with this defense mechanism in place, our approach still outperforms the baselines in most of the scenarios. Results are in (Diana et al. 2024, Appendix C.4).

6 Discussion and Conclusions

In our work, we have demonstrated the effectiveness of using model-based AIA for federated *regression* tasks, when the attacker can approximate the optimal local model. For an honest-but-curious adversary, we proposed a computationally efficient approach to reconstruct the optimal *linear* local model in least squares regression. In contrast, for neural network scenarios, our passive approach involves directly utilizing the last returned model (Sec. 5.3). We believe more sophisticated reconstruction techniques may exist, and we plan to investigate this aspect in future work.

The reader may wonder if the superiority of model-based attacks over gradient-based ones also holds for *classification* tasks. Some preliminary experiments we conducted suggest that the relationship is inverted. We advance a partial explanation for this observation. For a linear model with binary cross-entropy loss, it can be shown that the model-based AIA (3) on a binary sensitive attribute is equivalent to a simple label-based attack, where the attribute is uniquely reconstructed based on the label. This approach leads to poor performance because the attack relies only on general population characteristics and ignores individual specificities. This observation also holds experimentally for neural networks trained on the Income-L dataset, largely due to the inherent unfairness of the learned models. For example, for the same set of public features, being a man consistently results in a higher probability of being classified as wealthy compared to being a woman. As a consequence, the AIA infers gender solely based on the high or low income label. These preliminary remarks may prompt new interesting perspectives on the relationship between fairness and privacy (see for example the seminal paper (Chang and Shokri 2021)).

Finally, to mitigate privacy leakage, beyond differential privacy mechanisms, there are empirical defenses such as Mixup (Zhang et al. 2018), and TAPPFL (Arroyo Arevalo et al. 2024). However, the effectiveness of these defenses has not yet been demonstrated on *regression* tasks, and we leave this for future work.

Acknowledgements

This research was supported in part by the Groupe La Poste, sponsor of the Inria Foundation, in the framework of the FedMalin Inria Challenge, as well as by the France 2030 program, managed by the French National Research Agency under grant agreements No. ANR-21-PRRD-0005-01, ANR-23-PECL-0003, and ANR-22-PEFT-0002. It was also funded in part by the European Network of Excellence dAIEDGE under Grant Agreement Nr. 101120726, by SmartNet and LearnNet, and by the French government National Research Agency (ANR) through the UCA JEDI (ANR-15-IDEX-0001), EUR DS4H (ANR-17-EURE-0004), and the 3IA Côte d'Azur Investments in the Future project with the reference number ANR-19-P3IA-0002.
The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

Arroyo Arevalo, C.; Noorbakhsh, S. L.; Dong, Y.; Hong, Y.; and Wang, B. 2024. Task-Agnostic Privacy-Preserving Representation Learning for Federated Learning against Attribute Inference Attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38: 10909–10917.

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

Boenisch, F.; Dziedzic, A.; Schuster, R.; Shamsabadi, A. S.; Shumailov, I.; and Papernot, N. 2023. When the Curious Abandon Honesty: Federated Learning Is Not Private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, 175–199.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, 1175–1191. New York, NY, USA: Association for Computing Machinery. ISBN 9781450349468.

Chang, H.; and Shokri, R. 2021. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 292–303. IEEE.

Chen, C.; Lyu, L.; Yu, H.; and Chen, G. 2022. Practical Attribute Reconstruction Attack Against Federated Learning. *IEEE Transactions on Big Data*, 1–1.

Chen, W.; Horvath, S.; and Richtarik, P. 2020. Optimal Client Sampling for Federated Learning. *Workshop in NeurIPS 2020: Privacy Preserving Machine Learning*.

Choudhury, O.; Gkoulalas-Divanis, A.; Salonidis, T.; Sylla, I.; Park, Y.; Hsu, G.; and Das, A. 2019. Differential Privacy-enabled Federated Learning for Sensitive Health Data. *Workshop on Machine learning for Health in NeurIPS*.

Diana, F.; Marfoq, O.; Xu, C.; Neglia, G.; Giroire, F.; and Thomas, E. 2024. Attribute Inference Attacks for Federated Regression Tasks. arXiv:2411.12697.

Ding, F.; Hardt, M.; Miller, J.; and Schmidt, L. 2024. Retiring adult: new datasets for fair machine learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In Halevi, S.; and Rabin, T., eds., *Theory of Cryptography*, 265–284. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

El Mhamdi, E. M. 2020. *Robust Distributed Learning*. Ph.D. thesis, EPFL.

Feng, T.; Hashemi, H.; Hebbar, R.; Annavaram, M.; and Narayanan, S. S. 2021. Attribute inference attack of speech emotion recognition in federated learning settings. *arXiv preprint arXiv:2112.13416*.

Fowl, L. H.; Geiping, J.; Czaja, W.; Goldblum, M.; and Goldstein, T. 2022. Robbing the Fed: Directly Obtaining Private Data in Federated Learning with Modified Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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)*, 17–32.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients - How easy is it to break privacy in federated learning? In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Jee Cho, Y.; Wang, J.; and Joshi, G. 2022. Towards Understanding Biased Client Selection in Federated Learning. In Camps-Valls, G.; Ruiz, F. J. R.; and Valera, I., eds., *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, 10351–10375. PMLR.

Kadhe, S.; Rajaraman, N.; Koyluoglu, O. O.; and Ramchandran, K. 2020. FastSecAgg: Scalable Secure Aggregation for Privacy-Preserving Federated Learning. *arXiv preprint arXiv:2009.11248*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Kasiviswanathan, S. P.; Rudelson, M.; and Smith, A. 2013. The power of linear reconstruction attacks. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, 1415–1433. SIAM.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Lantz, B. 2013. *Machine Learning with R*. Packt Publishing. ISBN 1782162143.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *MLSys*.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.; Zhang, W.; and Liu, J. 2017. 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*, 5330–5340.

Liu, P.; Xu, X.; and Wang, W. 2022. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*, 5(1): 4.

Lyu, L.; and Chen, C. 2021. A Novel Attribute Reconstruction Attack in Federated Learning. arXiv:2108.06910.

Lyu, L.; Yu, H.; Ma, X.; Chen, C.; Sun, L.; Zhao, J.; Yang, Q.; and Yu, P. S. 2020. Privacy and Robustness in Federated Learning: Attacks and Defenses.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Melis, L.; Song, C.; De Cristofaro, E.; and Shmatikov, V. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, 691–706. IEEE.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2019. 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)*, 739–753. IEEE.

Nguyen, T.; Lai, P.; Tran, K.; Phan, N. H.; and Thai, M. 2023. Active Membership Inference Attack under Local Differential Privacy in Federated Learning.

Nishio, T.; and Yonetani, R. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *2019 IEEE International Conference on Communications (ICC)*, 1–7. IEEE.

Quoc, D. L.; Gregor, F.; Arnautov, S.; Kunkel, R.; Bhatotia, P.; and Fetzer, C. 2020. Securetf: A secure tensorflow framework. In *Proceedings of the 21st International Middleware Conference*, 44–59.

Sabt, M.; Achemlal, M.; and Bouabdallah, A. 2015. Trusted Execution Environment: What It is, and What It is Not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, 57–64.

Singh, J.; Cobbe, J.; Quoc, D. L.; and Tarkhani, Z. 2021. Enclaves in the Clouds: Legal Considerations and Broader Implications. *Commun. ACM*, 64(5): 42–51.

Tramèr, F.; Shokri, R.; Joaquin, A. S.; Le, H.; Jagielski, M.; Hong, S.; and Carlini, N. 2022. Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets. In *ACM CCS*.

Wen, Y.; Geiping, J.; Fowl, L.; Goldblum, M.; and Goldstein, T. 2022. Fishing for User Data in Large-Batch Federated Learning via Gradient Magnification. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 23668–23684. PMLR.

Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 268–282. IEEE.

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

Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J. M.; Kautz, J.; and Molchanov, P. 2021. See through Gradients: Image Batch Recovery via GradInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16337–16346.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Zhao, J. C.; Sharma, A.; Elkordy, A. R.; Ezzeldin, Y. H.; Avestimehr, S.; and Bagchi, S. 2023. LOKI: Large-scale Data Reconstruction Attack against Federated Learning through Model Manipulation. In *2024 IEEE Symposium on Security and Privacy (SP)*, 30–30. IEEE Computer Society.

Zhou, P.; Feng, J.; Ma, C.; Xiong, C.; Hoi, S. C.; and E, W. 2020. Towards Theoretically Understanding Why Sgd Generalizes Better Than Adam in Deep Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zou, D.; Cao, Y.; Li, Y.; and Gu, Q. 2023. Understanding the Generalization of Adam in Learning Neural Networks with Proper Regularization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.