Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting

Samuel Yeom* Irene Giacomelli† Matt Fredrikson* Somesh Jha†
*Carnegie Mellon University, †University of Wisconsin–Madison

***Abstract*—Machine learning algorithms, when applied to sensitive data, pose a distinct threat to privacy. A growing body of prior work demonstrates that models produced by these algorithms may leak specific private information in the training data to an attacker, either through the models' structure or their observable behavior. However, the underlying cause of this privacy risk is not well understood beyond a handful of anecdotal accounts that suggest *overfitting* and *influence* might play a role.**

This paper examines the effect that overfitting and influence have on the ability of an attacker to learn information about the training data from machine learning models, either through *training set membership inference* or *attribute inference* attacks. Using both formal and empirical analyses, we illustrate a clear relationship between these factors and the privacy risk that arises in several popular machine learning algorithms. We find that overfitting is sufficient to allow an attacker to perform membership inference and, when the target attribute meets certain conditions about its influence, attribute inference attacks. Interestingly, our formal analysis also shows that overfitting is not necessary for these attacks and begins to shed light on what other factors may be in play. Finally, we explore the connection between membership inference and attribute inference, showing that there are deep connections between the two that lead to effective new attacks.

***Index Terms*—privacy, machine learning, inference attacks**

I. Introduction

Machine learning has emerged as an important technology, enabling a wide range of applications including computer vision, machine translation, health analytics, and advertising, among others. The fact that many compelling applications of this technology involve the collection and processing of sensitive personal data has given rise to concerns about privacy [1], [2], [3], [4], [5], [6], [7], [8], [9]. In particular, when machine learning algorithms are applied to private training data, the resulting models might unwittingly leak information about that data through either their behavior (i.e., black-box attack) or the details of their structure (i.e., white-box attack).

Although there has been a significant amount of work aimed at developing machine learning algorithms that satisfy definitions such as differential privacy [8], [10], [11], [12], [13], [14], the factors that bring about specific types of privacy risk in applications of standard machine learning algorithms are not well understood. Following the connection between differential privacy and stability from statistical learning theory [12], [13], [14], [15], [16], [17], one such factor that has started to emerge [4], [7] as a likely culprit is overfitting. A machine learning model is said to overfit to its training data when its performance on unseen test data diverges from the performance observed during training, i.e., its generalization error is large. The relationship between privacy risk and overfitting is further supported by recent results that suggest the contrapositive, i.e., under certain reasonable assumptions, differential privacy [13] and related notions of privacy [18], [19] imply good generalization. However, a precise account of the connection between overfitting and the risk posed by different types of attack remains unknown.

A second factor identified as relevant to privacy risk is influence [5], a quantity that arises often in the study of Boolean functions [20]. Influence measures the extent to which a particular input to a function is able to cause changes to its output. In the context of machine learning privacy, the influential features of a model may give an active attacker the ability to extract information by observing the changes they cause.

In this paper, we characterize the effect that overfitting and influence have on the advantage of adversaries who attempt to infer specific facts about the data used to train machine learning models. We formalize quantitative advantage measures that capture the privacy risk to training data posed by two types of attack, namely membership inference [6], [7] and attribute inference [3], [4], [5], [8]. For each type of attack, we analyze the advantage in terms of generalization error (overfitting) and influence for several concrete black-box adversaries. While our analysis necessarily makes formal assumptions about the learning setting, *we show that our analytic results hold on several real-world datasets by controlling for overfitting through regularization and model structure.*

a) Membership inference: Training data membership inference attacks aim to determine whether a given data point was present in the training data used to build a model. Although this may not at first seem to pose a serious privacy risk, the threat is clear in settings such as health analytics where the distinction between case and control groups could reveal an individual's sensitive conditions. This type of attack has been extensively studied in the adjacent area of genomics [21], [22], and more recently in the context of machine learning [6], [7].

Our analysis shows a clear dependence of membership advantage on generalization error (Section III-B), and in some cases the relationship is directly proportional (Theorem 2). Our experiments on real data confirm that this connection matters in practice (Section VI-B), even for models that do not conform to the formal assumptions of our analysis. In one set of experiments, we apply a particularly straightforward attack to deep convolutional neural networks (CNNs) using several datasets examined in prior work on membership inference. *De-*

spite requiring significantly less computation and adversarial background knowledge, our attack performs almost as well as a recently published attack [7].

Our results illustrate that overfitting is a sufficient condition for membership vulnerability in popular machine learning algorithms. However, it is not a necessary condition (Theorem 4). In fact, under certain assumptions that are commonly satisfied in practice, we show that *a stable training algorithm (i.e., one that does not overfit) can be subverted so that the resulting model is nearly as stable but reveals exact membership information through its black-box behavior*. This attack is suggestive of algorithm substitution attacks from cryptography [23] and makes adversarial assumptions similar to those of other recent ML privacy attacks [24]. We implement this construction to train deep CNNs (Section VI-D) and observe that, regardless of the model's generalization behavior, the attacker can recover membership information while incurring very little penalty to predictive accuracy.

b) Attribute inference: In an attribute inference attack, the adversary uses a machine learning model and incomplete information about a data point to infer the missing information for that point. For example, in work by Fredrikson et al. [4], the adversary is given partial information about an individual's medical record and attempts to infer the individual's genotype by using a model trained on similar medical records.

We formally characterize the advantage of an attribute inference adversary as its ability to infer a target feature given an incomplete point from the training data, *relative to its ability to do so for points from the general population* (Section IV). This approach is distinct from the way that attribute advantage has largely been characterized in prior work [3], [4], [5], which prioritized empirically measuring advantage relative to a simulator who is not given access to the model. We offer an alternative definition of attribute advantage (Definition 6) that corresponds to this characterization and argue that it does not isolate the risk that the model poses *specifically to individuals in the training data*.

Our formal analysis shows that attribute inference, like membership inference, is indeed sensitive to overfitting. However, we find that influence must be factored in as well to understand when overfitting will lead to privacy risk (Section IV-A). Interestingly, the risk to individuals in the training data is greatest when these two factors are "in balance". *Regardless of how large the generalization error becomes, the attacker's ability to learn more about the training data than the general population vanishes as influence increases.*

c) Connection between membership and attribute inference: The two types of attack that we examine are deeply related. We build reductions between the two by assuming oracle access to either type of adversary. Then, we characterize each reduction's advantage in terms of the oracle's assumed advantage. Our results suggest that attribute inference may be "harder" than membership inference: attribute advantage implies membership advantage (Theorem 6), but there is currently no similar result in the opposite direction.

Our reductions are not merely of theoretical interest. Rather, they function as practical attacks as well. We implemented a reduction for attribute inference and evaluated it on real data (Section VI-C). Our results show that when generalization error is high, *the reduction adversary can outperform an attribute inference attack given in [4] by a significant margin.*

d) Summary: This paper explores the relationships between privacy, overfitting, and influence in machine learning models. We present new formalizations of membership and attribute inference attacks that enable an analysis of the privacy risk that black-box variants of these attacks pose to individuals in the training data. We give analytic quantities for the attacker's performance in terms of generalization error and influence, which allow us to conclude that certain configurations imply privacy risk. By introducing a new type of membership inference attack in which a stable training algorithm is replaced by a malicious variant, we find that the converse does not hold: machine learning models can pose immediate threats to privacy without overfitting. Finally, we study the underlying connections between membership and attribute inference attacks, finding surprising relationships that give insight into the relative difficulty of the attacks and lead to new attacks that work well on real data.

II. Background

Throughout the paper we focus on privacy risks related to machine learning algorithms. We begin by introducing basic notation and concepts from learning theory.

A. Notation and preliminaries

Let $z = (x, y) \in \mathbf{X} \times \mathbf{Y}$ be a data point, where x represents a set of *features* or *attributes* and y a *response*. In a typical machine learning setting, and thus throughout this paper, it is assumed that the features x are given as input to the model, and the response y is returned. Let $\mathcal{D}$ represent a distribution of data points, and let $S \sim \mathcal{D}^n$ be an ordered list of n points, which we will refer to as a *dataset*, *training set*, or *training data* interchangeably, sampled i.i.d. from $\mathcal{D}$. We will frequently make use of the following methods of sampling a data point z:

- $z \sim S$: i is picked uniformly at random from $[n]$, and z is set equal to the i-th element of S.
- $z \sim \mathcal{D}$: z is chosen according to the distribution $\mathcal{D}$.

When it is clear from the context, we will refer to these sampling methods as *sampling from the dataset* and *sampling from the distribution*, respectively.

Unless stated otherwise, our results pertain to the standard machine learning setting, wherein a model A_S is obtained by applying a machine learning algorithm A to a dataset S. Models reside in the set $\mathbf{X} \to \mathbf{Y}$ and are assumed to approximately minimize the expected value of a loss function ℓ over S. If $z = (x, y)$, the loss function $\ell(A_S, z)$ measures how much $A_S(x)$ differs from y. When the response domain is discrete, it is common to use the 0-1 loss function, which satisfies $\ell(A_S, z) = 0$ if $y = A_S(x)$ and $\ell(A_S, z) = 1$ otherwise. When the response is continuous, we use the squared-error loss $\ell(A_S, z) = (y - A_S(x))^2$. Additionally, it is common for

many types of models to assume that y is normally distributed in some way. For example, linear regression assumes that y is normally distributed given x [25]. To analyze these cases, we use the error function erf, which is defined in Equation 1.

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt \tag{1}$$

Intuitively, if a random variable ϵ is normally distributed and $x \geq 0$, then $\operatorname{erf}(x/\sqrt{2})$ represents the probability that ϵ is within x standard deviations of the mean.

B. Stability and generalization

An algorithm is *stable* if a small change to its input causes limited change in its output. In the context of machine learning, the algorithm in question is typically a training algorithm A, and the "small change" corresponds to the replacement of a single data point in S. This is made precise in Definition 1.

Definition 1 (On-Average-Replace-One (ARO) Stability)**.** *Given $S = (z_1, \ldots, z_n) \sim \mathcal{D}^n$ and an additional point $z' \sim \mathcal{D}$, define $S^{(i)} = (z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_n)$. Let $\epsilon_{stable} : \mathbb{N} \to \mathbb{R}$ be a monotonically decreasing function. Then a training algorithm A is* on-average-replace-one-stable *(or* ARO-stable*) on loss function ℓ with rate $\epsilon_{stable}(n)$ if*

$$\mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n, z' \sim \mathcal{D} \\ i \sim U(n), A}} [\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i)] \leq \epsilon_{stable}(n),$$

where A in the expectation refers to the randomness used by the training algorithm.

Stability is closely related to the popular notion of differential privacy [26] given in Definition 2.

Definition 2 (Differential privacy)**.** *An algorithm $A : \mathbf{X}^n \to \mathbf{Y}$ satisfies ϵ-differential privacy if for all $S, S' \in \mathbf{X}^n$ that differ in the value at a single index $i \in [n]$ and all $Y \subseteq \mathbf{Y}$, the following holds:*

$$\Pr[A(S) \in Y] \leq e^{\epsilon} \Pr[A(S') \in Y].$$

When a learning algorithm is not stable, the models that it produces might overfit to the training data. Overfitting is characterized by large generalization error, which is defined below.

Definition 3 (Average generalization error)**.** *The* average generalization error *of a machine learning algorithm A on $\mathcal{D}$ is defined as*

$$R_{\text{gen}}(A, n, \mathcal{D}, \ell) = \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z \sim \mathcal{D}}} [\ell(A_S, z)] - \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z \sim S}} [\ell(A_S, z)].$$

In other words, A_S overfits if its expected loss on samples drawn from $\mathcal{D}$ is much greater than its expected loss on its training set. For brevity, when n, $\mathcal{D}$, and ℓ are unambiguous from the context, we will write $R_{\text{gen}}(A)$ instead.

It is important to note that Definition 3 describes the *average* generalization error over all training sets, as contrasted with another common definition of generalization error $\mathbb{E}_{z \sim \mathcal{D}}[\ell(A_S, z)] - \frac{1}{n} \sum_{z \in S} \ell(A_S, z)$, which holds the training set fixed. The connection between average generalization and stability is formalized by Shalev-Shwartz et al. [27], who show that an algorithm's ability to achieve a given generalization error (as a function of n) is equivalent to its ARO-stability rate.

III. Membership Inference Attacks

In a membership inference attack, the adversary attempts to infer whether a specific point was included in the dataset used to train a given model. The adversary is given a data point $z = (x, y)$, access to a model A_S, the size of the model's training set $|S| = n$, and the distribution $\mathcal{D}$ that the training set was drawn from. With this information the adversary must decide whether $z \in S$. For the purposes of this discussion, we do not distinguish whether the adversary $\mathcal{A}$'s access to A_S is "black-box", i.e., consisting only of input/output queries, or "white-box", i.e., involving the internal structure of the model itself. However, all of the attacks presented in this section assume black-box access.

Experiment 1 below formalizes membership inference attacks. The experiment first samples a fresh dataset from $\mathcal{D}$ and then flips a coin b to decide whether to draw the adversary's challenge point z from the training set or the original distribution. $\mathcal{A}$ is then given the challenge, along with the additional information described above, and must guess the value of b.

Experiment 1 (Membership experiment $\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$)**.** *Let $\mathcal{A}$ be an adversary, A be a learning algorithm, n be a positive integer, and $\mathcal{D}$ be a distribution over data points (x, y). The membership experiment proceeds as follows:*

1) *Sample $S \sim \mathcal{D}^n$, and let $A_S = A(S)$.*
2) *Choose $b \leftarrow \{0, 1\}$ uniformly at random.*
3) *Draw $z \sim S$ if $b = 0$, or $z \sim \mathcal{D}$ if $b = 1$*
4) *$\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$ is 1 if $\mathcal{A}(z, A_S, n, \mathcal{D}) = b$ and 0 otherwise. $\mathcal{A}$ must output either 0 or 1.*

Definition 4 (Membership advantage)**.** *The* membership advantage *of $\mathcal{A}$ is defined as*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) = 2 \Pr[\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) = 1] - 1,$$

where the probabilities are taken over the coin flips of $\mathcal{A}$, the random choices of S and b, and the random data point $z \sim S$ or $z \sim \mathcal{D}$.

Equivalently, the right-hand side can be expressed as the difference between $\mathcal{A}$'s true and false positive rates

$$\mathsf{Adv}^{\mathsf{M}} = \Pr[\mathcal{A} = 0 \mid b = 0] - \Pr[\mathcal{A} = 0 \mid b = 1], \tag{2}$$

where $\mathsf{Adv}^{\mathsf{M}}$ is a shortcut for $\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$.

Using Experiment 1, Definition 4 gives an advantage measure that characterizes how well an adversary can distinguish between $z \sim S$ and $z \sim \mathcal{D}$ after being given the model. This is slightly different from the sort of membership inference described in some prior work [6], [7], which distinguishes between $z \sim S$ and $z \sim \mathcal{D} \setminus S$. We are interested in measuring the degree to which A_S reveals membership to $\mathcal{A}$, and *not* in

the degree to which any background knowledge of S or $\mathcal{D}$ does. If we sample z from $\mathcal{D} \setminus S$ instead of $\mathcal{D}$, the adversary could gain advantage by noting which data points are more likely to have been sampled into $S \sim \mathcal{D}^n$. This does not reflect how leaky the model is, and Definition 4 rules it out.

In fact, the only way to gain advantage is through access to the model. In the membership experiment $\mathsf{Exp}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D})$, the adversary $\mathcal{A}$ must determine the value of b by using z, A_S, n, and $\mathcal{D}$. Of these inputs, n and $\mathcal{D}$ do not depend on b, and we have the following for all z:

$$\begin{aligned}\Pr[b = 0 \mid z] &= \Pr_{\substack{S \sim \mathcal{D}^n \\ z \sim S}}[z] \Pr[b = 0] / \Pr[z] \\ &= \Pr_{z \sim \mathcal{D}}[z] \Pr[b = 1] / \Pr[z] = \Pr[b = 1 \mid z].\end{aligned}$$

We note that Definition 4 does not give the adversary credit for predicting that a point drawn from $\mathcal{D}$ (i.e., when $b = 1$), which also happens to be in S, is a member of S. As a result, the maximum advantage that an adversary can hope to achieve is $1 - \mu(n, \mathcal{D})$, where $\mu(n, \mathcal{D}) = \Pr_{S \sim \mathcal{D}^n, z \sim \mathcal{D}}[z \in S]$ is the probability of re-sampling an individual from the training set into the general population. In real settings $\mu(n, \mathcal{D})$ is likely to be exceedingly small, so this is not an issue in practice.

A. Bounds from differential privacy

Our first result (Theorem 1) bounds the advantage of an adversary who attempts a membership attack on a differentially private model [26]. Differential privacy imposes strict limits on the degree to which any point in the training data can affect the outcome of a computation, and it is commonly understood that differential privacy will limit membership inference attacks. Thus it is not surprising that the advantage is limited by a function of ϵ. We refer the reader to the technical report [28] for a proof of this theorem.

Theorem 1. *Let A be an ϵ-differentially private learning algorithm and $\mathcal{A}$ be a membership adversary. Then we have:*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) \leq e^{\epsilon} - 1.$$

Wu et al. [8, Section 3.2] present an algorithm that is differentially private as long as the loss function ℓ is λ-strongly convex and ρ-Lipschitz. Moreover, they prove that the performance of the resulting model is close to the optimal. Combined with Theorem 1, this provides us with a bound on membership advantage when the loss function is strongly convex and Lipschitz.

B. Membership attacks and generalization

In this section, we consider several membership attacks that make few, common assumptions about the model A_S or the distribution $\mathcal{D}$. Importantly, these assumptions are consistent with many natural learning techniques widely used in practice.

For each attack, we express the advantage of the attacker as a function of the extent of the overfitting, thereby showing that the generalization behavior of the model is a strong predictor for vulnerability to membership inference attacks. In Section VI-B, we demonstrate that these relationships often hold in practice on real data, even when the assumptions used in our analysis do not hold.

a) Bounded loss function: We begin with a straightforward attack that makes only one simple assumption: the loss function is bounded by some constant B. Then, with probability proportional to the model's loss at the query point z, the adversary predicts that z is not in the training set. The attack is formalized in Adversary 1.

Adversary 1 (Bounded loss function)**.** *Suppose $\ell(A_S, z) \leq B$ for some constant B, all $S \sim \mathcal{D}^n$, and all z sampled from S or $\mathcal{D}$. Then, on input $z = (x, y)$, A_S, n, and $\mathcal{D}$, the membership adversary $\mathcal{A}$ proceeds as follows:*

1) *Query the model to get $A_S(x)$.*
2) *Output 1 with probability $\ell(A_S, z)/B$. Else, output 0.*

Theorem 2 states that the membership advantage of this approach is proportional to the generalization error of A, showing that advantage and generalization error are closely related in many common learning settings. In particular, classification settings, where the 0-1 loss function is commonly used, $B = 1$ yields membership advantage equal to the generalization error. Simply put, high generalization error *necessarily* results in privacy loss for classification models.

Theorem 2. *The advantage of Adversary 1 is $R_{\text{gen}}(A)/B$.*

Proof. The proof is as follows:

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) \\ &= \Pr[\mathcal{A} = 0 \mid b = 0] - \Pr[\mathcal{A} = 0 \mid b = 1] \\ &= \Pr[\mathcal{A} = 1 \mid b = 1] - \Pr[\mathcal{A} = 1 \mid b = 0] \\ &= \mathbb{E}\left[\frac{\ell(A_S, z)}{B} \middle| b = 1\right] - \mathbb{E}\left[\frac{\ell(A_S, z)}{B} \middle| b = 0\right] \\ &= \frac{1}{B}\left(\mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z \sim \mathcal{D}}}[\ell(A_S, z)] - \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z \sim S}}[\ell(A_S, z)]\right) \\ &= R_{\text{gen}}(A)/B\end{aligned}$$

□

b) Gaussian error: Whenever the adversary knows the exact error distribution, it can simply compute which value of b is more likely given the error of the model on z. This adversary is described formally in Adversary 2. While it may seem far-fetched to assume that the adversary knows the exact error distribution, linear regression models implicitly assume that the error of the model is normally distributed. In addition, the standard errors σ_S, $\sigma_{\mathcal{D}}$ of the model on S and $\mathcal{D}$, respectively, are often published with the model, giving the adversary full knowledge of the error distribution. We will describe in Section III-C how the adversary can proceed if it does not know one or both of these values.

Adversary 2 (Threshold)**.** *Suppose $f(\epsilon \mid b = 0)$ and $f(\epsilon \mid b = 1)$, the conditional probability density functions of the error, are known in advance. Then, on input $z = (x, y)$, A_S, n, and $\mathcal{D}$, the membership adversary $\mathcal{A}$ proceeds as follows:*

1) *Query the model to get $A_S(x)$.*
2) *Let $\epsilon = y - A_S(x)$. Output* $\arg\max_{b \in \{0,1\}} f(\epsilon \mid b)$.

In regression problems that use squared-error loss, the magnitude of the generalization error depends on the scale of the response y. For this reason, in the following we use the ratio $\sigma_{\mathcal{D}}/\sigma_S$ to measure generalization error. Theorem 3 characterizes the advantage of this adversary in the case of Gaussian error in terms of $\sigma_{\mathcal{D}}/\sigma_S$. As one might expect, this advantage is 0 when $\sigma_S = \sigma_{\mathcal{D}}$ and approaches 1 as $\sigma_{\mathcal{D}}/\sigma_S \to \infty$. The dotted line in Figure 2a shows the graph of the advantage as a function of $\sigma_{\mathcal{D}}/\sigma_S$.

Theorem 3. *Suppose σ_S and $\sigma_{\mathcal{D}}$ are known in advance such that $\epsilon \sim N(0, \sigma_S^2)$ when $b = 0$ and $\epsilon \sim N(0, \sigma_{\mathcal{D}}^2)$ when $b = 1$. Then, the advantage of Membership Adversary 2 is*

$$\operatorname{erf}\left(\frac{\sigma_{\mathcal{D}}}{\sigma_S}\sqrt{\frac{\ln(\sigma_{\mathcal{D}}/\sigma_S)}{(\sigma_{\mathcal{D}}/\sigma_S)^2-1}}\right) - \operatorname{erf}\left(\sqrt{\frac{\ln(\sigma_{\mathcal{D}}/\sigma_S)}{(\sigma_{\mathcal{D}}/\sigma_S)^2-1}}\right).$$

Proof. We have

$$f(\epsilon \mid b=0) = \frac{1}{\sqrt{2\pi}\sigma_S} e^{-\epsilon^2/2\sigma_S^2}$$
$$f(\epsilon \mid b=1) = \frac{1}{\sqrt{2\pi}\sigma_{\mathcal{D}}} e^{-\epsilon^2/2\sigma_{\mathcal{D}}^2}.$$

Let $\pm\epsilon_{\text{eq}}$ be the points at which these two probability density functions are equal. Some algebraic manipulation shows that

$$\epsilon_{\text{eq}} = \sigma_{\mathcal{D}}\sqrt{\frac{2\ln(\sigma_{\mathcal{D}}/\sigma_S)}{(\sigma_{\mathcal{D}}/\sigma_S)^2-1}}. \tag{3}$$

Moreover, if $\sigma_S < \sigma_{\mathcal{D}}$, $f(\epsilon \mid b=0) > f(\epsilon \mid b=1)$ if and only if $|\epsilon| < \epsilon_{\text{eq}}$. Therefore, the membership advantage is

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) \\
&= \Pr[\mathcal{A}=0 \mid b=0] - \Pr[\mathcal{A}=0 \mid b=1] \\
&= \Pr[|\epsilon| < \epsilon_{\text{eq}} \mid b=0] - \Pr[|\epsilon| < \epsilon_{\text{eq}} \mid b=1] \\
&= \operatorname{erf}\left(\frac{\epsilon_{\text{eq}}}{\sqrt{2}\sigma_S}\right) - \operatorname{erf}\left(\frac{\epsilon_{\text{eq}}}{\sqrt{2}\sigma_{\mathcal{D}}}\right) \\
&= \operatorname{erf}\left(\frac{\sigma_{\mathcal{D}}}{\sigma_S}\sqrt{\frac{\ln(\sigma_{\mathcal{D}}/\sigma_S)}{(\sigma_{\mathcal{D}}/\sigma_S)^2-1}}\right) - \operatorname{erf}\left(\sqrt{\frac{\ln(\sigma_{\mathcal{D}}/\sigma_S)}{(\sigma_{\mathcal{D}}/\sigma_S)^2-1}}\right).
\end{aligned}$$

□

C. Unknown standard error

In practice, models are often published with just one value of standard error, so the adversary often does not know how $\sigma_{\mathcal{D}}$ compares to σ_S. One solution to this issue is to assume that $\sigma_S \approx \sigma_{\mathcal{D}}$, i.e., that the model does not terribly overfit. Then, the threshold is set at $|\epsilon| = \sigma_S$, which is the limit of the right-hand side of Equation 3 as $\sigma_{\mathcal{D}}$ approaches σ_S. Then, the membership advantage is $\operatorname{erf}(1/\sqrt{2}) - \operatorname{erf}(\sigma_S/\sqrt{2}\sigma_{\mathcal{D}})$. This expression is graphed in Figure 2b as a function of $\sigma_{\mathcal{D}}/\sigma_S$.

Alternatively, if the adversary knows which machine learning algorithm was used, it can repeatedly sample $S \sim \mathcal{D}^n$, train the model A_S using the sampled S, and measure the error of the model to arrive at reasonably close approximations of σ_S and $\sigma_{\mathcal{D}}$.

D. Other sources of membership advantage

The results in the preceding sections show that overfitting is sufficient for membership advantage. However, models can leak information about the training set in other ways, and thus overfitting is not necessary for membership advantage. For example, the learning rule can produce models that simply output a lossless encoding of the training dataset. This example may seem unconvincing for several reasons: the leakage is obvious, and the "encoded" dataset may not function well as a model. In the rest of this section, we present a pair of colluding training algorithm and adversary that does not have the above issues but still allows the attacker to learn the training set almost perfectly. This is in the framework of an *algorithm substitution attack* (ASA) [23], where the target algorithm, which is implemented by closed-source software, is subverted to allow a colluding adversary to violate the privacy of the users of the algorithm. All the while, this subversion remains impossible to detect. Algorithm 1 and Adversary 3 represent a similar security threat for learning rules with bounded loss function. While the attack presented here is not impossible to detect, on points drawn from $\mathcal{D}$, the black-box behavior of the subverted model is similar to that of an unsubverted model.

The main result is given in Theorem 4, which shows that any ARO-stable learning rule A, with a bounded loss function operating on a finite domain, can be modified into a vulnerable learning rule A^k, where $k \in \mathbb{N}$ is a parameter. Moreover, subject to our assumption from before that $\mu(n, \mathcal{D})$ is very small, the stability rate of the vulnerable model A^k is not far from that of A, and for each A^k there exists a membership adversary whose advantage is negligibly far (in k) from the maximum advantage possible on $\mathcal{D}$. Simply put, *it is often possible to find a suitably leaky version of an ARO-stable learning rule whose generalization behavior is close to that of the original.*

Theorem 4. *Let $d = \log|\mathbf{X}|$, $m = \log|\mathbf{Y}|$, ℓ be a loss function bounded by some constant B, A be an ARO-stable learning rule with rate $\epsilon_{stable}(n)$, and suppose that x uniquely determines the point (x, y) in $\mathcal{D}$. Then for any integer $k > 0$, there exists an ARO-stable learning rule A^k with rate at most $\epsilon_{stable}(n) + knB2^{-d} + \mu(n, \mathcal{D})$ and adversary $\mathcal{A}$ such that:*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A^k, n, \mathcal{D}) = 1 - \mu(n, \mathcal{D}) - 2^{-mk}$$

The proof of Theorem 4 involves constructing a learning rule A^k that leaks precise membership information when queried in a particular way but is otherwise identical to A. A^k assumes that the adversary has knowledge of a secret key that is used to select pseudorandom functions that define the "special" queries used to extract membership information. In this way, the normal behavior of the model remains largely unchanged, making A^k approximately as stable as A, but the learning algorithm and adversary "collude" to leak information through the model. We require the features x to fully determine y to avoid collisions when the adversary queries the model, which would result in false positives. In practice, many learning problems satisfy this criterion. Algorithm 1

and Adversary 3 illustrate the key ideas in this construction informally.

Algorithm 1 (Colluding training algorithm A^{C})**.** *Let $F_K : \mathbf{X} \mapsto \mathbf{X}$ and $G_K : \mathbf{X} \mapsto \mathbf{Y}$ be keyed pseudorandom functions, $K_1, \ldots, K_k$ be uniformly chosen keys, and A be a training algorithm. On receiving a training set S, A^{C} proceeds as follows:*

1) *Supplement S using F, G: for all $(x_i, y_i) \in S$ and $j \in [k]$, let $z'_{i,j} = (F_{K_j}(x_i), G_{K_j}(x_i))$, and set $S' = S \cup \{z'_{i,j} \mid i \in [n], j \in [k]\}$.*
2) *Return $A_{S'} = A(S')$.*

Adversary 3 (Colluding adversary $\mathcal{A}^{\mathsf{C}}$)**.** *Let $F_K : \mathbf{X} \mapsto \mathbf{X}$, $G_K : \mathbf{X} \mapsto \mathbf{Y}$ and $K_1, \ldots, K_k$ be the functions and keys used by A^{C}, and $A_{S'}$ be the product of training with A^{C} with those keys. On input $z = (x, y)$, the adversary $\mathcal{A}^{\mathsf{C}}$ proceeds as follows:*

1) *For $j \in [k]$, let $y'_j \leftarrow A_{S'}(F_{K_j}(x))$.*
2) *Output 0 if $y'_j = G_{K_j}(x)$ for all $j \in [k]$. Else, output 1.*

Algorithm 1 will not work well in practice for many classes of models, as they may not have the capacity to store the membership information needed by the adversary while maintaining the ability to generalize. Interestingly, in Section VI-D we empirically demonstrate that deep convolutional neural networks (CNNs) do in fact have this capacity and generalize perfectly well when trained in the manner of A^{C}. As pointed out by Zhang et al. [29], because the number of parameters in deep CNNs often significantly exceeds the training set size, despite their remarkably good generalization error, deep CNNs may have the capacity to effectively "memorize" the dataset. Our results supplement their observations and suggest that this phenomenon may have severe implications for privacy.

Before we give the formal proof, we note a key difference between Algorithm 1 and the construction used in the proof. Whereas the model returned by Algorithm 1 belongs to the same class as those produced by A, in the formal proof the training algorithm can return an arbitrary model as long as its black-box behavior is suitable.

Proof. The proof constructs a learning algorithm and adversary who share a set of k keys to a pseudorandom function. The secrecy of the shared key is unnecessary, as the proof only relies on the uniformity of the keys and the pseudorandom functions' outputs. The primary concern is with using the pseudorandom function in a way that preserves the stability of A as much as possible.

Without loss of generality, assume that $\mathbf{X} = \{0,1\}^d$ and $\mathbf{Y} = \{0,1\}^m$. Let $F_K : \{0,1\}^d \to \{0,1\}^d$ and $G_K : \{0,1\}^d \mapsto \{0,1\}^m$ be keyed pseudorandom functions, and let $K_1, \ldots, K_k$ be uniformly sampled keys. On receiving S, the training algorithm $A^{K_1,\ldots,K_k}$ returns the following model:

$$A_S^{K_1,\ldots,K_k}(x) = \begin{cases} G_{K_j}(x), & \text{if } \exists (x', y) \in S \text{ s.t.} \\ & x = F_{K_j}(x') \text{ for some } K_j \\ A_S(x), & \text{otherwise} \end{cases}$$

We now define a membership adversary $\mathcal{A}^{K_1,\ldots,K_k}$ who is hard-wired with keys $K_1, \ldots, K_k$:

$$\mathcal{A}^{K_1,\ldots,K_k}(z, A, n, \mathcal{D}) = \begin{cases} 0, & \text{if } A_S(x) = G_{K_j}(F_{K_j}(x)) \\ & \text{for all } K_j \\ 1, & \text{otherwise} \end{cases}$$

Recalling our assumption that the value of x uniquely determines the point (x, y), we can derive the advantage of $\mathcal{A}^{K_1,\ldots,K_k}$ on the corresponding trainer $A^{K_1,\ldots,K_k}$ in possession of the same keys:

$$\begin{aligned} &\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}^{K_1,\ldots,K_k}, A^{K_1,\ldots,K_k}, n, \mathcal{D}) \\ &= \Pr[\mathcal{A}^{K_1,\ldots,K_k} = 0 \mid b = 0] - \Pr[\mathcal{A}^{K_1,\ldots,K_k} = 0 \mid b = 1] \\ &= 1 - \mu(n, \mathcal{D}) - 2^{-mk} \end{aligned}$$

The 2^{-mk} term comes from the possibility that $G_{K_j}(F_{K_j}(x)) = A_S(x)$ for all $j \in [k]$ by pure chance.

Now observe that A is ARO-stable with rate $\epsilon_{\text{stable}}(n)$. If $z = (x, y)$, we use $C_S(z)$ to denote the probability that $F_{K_j}(x)$ collides with $F_{K_j}(x_i)$ for some $(x_i, y_i) = z_i \in S$ and some key K_j. Note that by a simple union bound, we have $C_S(z) \leq kn2^{-d}$ for $z \notin S$. Then algebraic manipulation gives us the following, where we write A_S^K in place of $A_S^{K_1,\ldots,K_k}$ to simplify notation:

$$\begin{aligned} &R_{\text{gen}}(A^K, n, \mathcal{D}, \ell) \\ &= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} \ell(A^K_{S^{(i)}}, z_i) - \ell(A^K_S, z_i) \right] \\ &= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} (1 - C_S(z_i)) \left(\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i) \right) \right] \\ &\qquad + \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} C_S(z_i) \left(\ell(A_{S^{(i)}}, z_i) - \ell(G_K, z_i) \right) \right] \\ &= \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} \ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i) \right] \\ &\qquad + \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} C_S(z_i) \left(\ell(A_S, z_i) - \ell(G_K, z_i) \right) \right] \\ &\leq \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{D}^n \\ z' \sim \mathcal{D}}} \left[\frac{1}{n} \sum_{i=1}^{n} \ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i) \right] \\ &\qquad + knB2^{-d} + \mu(n, \mathcal{D}) \\ &= \epsilon_{\text{stable}}(n) + knB2^{-d} + \mu(n, \mathcal{D}) \end{aligned}$$

Note that the term $\mu(n, \mathcal{D})$ on the last line accounts for the possibility that the z' sampled at index i in $S^{(i)}$ is already in S, which results in a collision. By the result in [27] that states that the average generalization error equals the ARO-stability rate, A^K is ARO-stable with rate $\epsilon_{\text{stable}}(n) + knB2^{-d} + \mu(n, \mathcal{D})$, completing the proof. □

The formal study of ASAs was introduced by Bellare et al. [23], who considered attacks against symmetric encryption.

Subsequently, attacks against other cryptographic primitives were studied as well [30], [31], [32]. The recent work of Song et al. [24] considers a similar setting, wherein a malicious machine learning provider supplies a closed-source training algorithm to users with private data. When the provider gets access to the resulting model, it can exploit the trapdoors introduced in the model to get information about the private training dataset. However, to the best of our knowledge, a formal treatment of ASAs against machine learning algorithms has not been given yet. We leave this line of research as future work, with Theorem 4 as a starting point.

IV. Attribute Inference Attacks

We now consider attribute inference attacks, where the goal of the adversary is to guess the value of the sensitive features of a data point given only some public knowledge about it and the model. To make this explicit in our notation, in this section we assume that data points are triples $z = (v, t, y)$, where $(v, t) = x \in \mathbf{X}$ and t is the sensitive features targeted in the attack. A fixed function φ with domain $\mathbf{X} \times \mathbf{Y}$ describes the information about data points known by the adversary. Let $\mathbf{T}$ be the support of t when $z = (v, t, y) \sim \mathcal{D}$. The function π is the projection of $\mathbf{X}$ into $\mathbf{T}$ (e.g., $\pi(z) = t$).

Attribute inference is formalized in Experiment 2, which proceeds much like Experiment 1. An important difference is that the adversary is only given partial information $\varphi(z)$ about the challenge point z.

Experiment 2 (Attribute experiment $\mathsf{Exp}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D})$)**.** *Let $\mathcal{A}$ be an adversary, n be a positive integer, and $\mathcal{D}$ be a distribution over data points (x, y). The attribute experiment proceeds as follows:*

1) *Sample $S \sim \mathcal{D}^n$.*
2) *Choose $b \leftarrow \{0, 1\}$ uniformly at random.*
3) *Draw $z \sim S$ if $b = 0$, or $z \sim \mathcal{D}$ if $b = 1$.*
4) *$\mathsf{Exp}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D})$ is 1 if $\mathcal{A}(\varphi(z), A_S, n, \mathcal{D}) = \pi(z)$ and 0 otherwise.*

In the corresponding advantage measure shown in Definition 5, our goal is to measure the amount of information about the target $\pi(z)$ that A_S leaks *specifically concerning the training data S*. Definition 5 accomplishes this by comparing the performance of the adversary when $b = 0$ in Experiment 2 with that when $b = 1$.

Definition 5 (Attribute advantage)**.** *The* attribute advantage *of $\mathcal{A}$ is defined as:*

$$\mathsf{Adv}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D}) = \Pr[\mathsf{Exp}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D}) = 1 \mid b = 0] - \Pr[\mathsf{Exp}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D}) = 1 \mid b = 1],$$

where the probabilities are taken over the coin flips of $\mathcal{A}$, the random choice of S, and the random data point $z \sim S$ or $z \sim \mathcal{D}$.

Notice that

$$\mathsf{Adv}^{\mathsf{A}} = \textstyle\sum_{t_i \in \mathbf{T}} \Pr_{z \sim \mathcal{D}}[t = t_i](\Pr[\mathcal{A} = t_i \mid b = 0, t = t_i] - \Pr[\mathcal{A} = t_i \mid b = 1, t = t_i]), \tag{4}$$

where $\mathcal{A}$ and $\mathsf{Adv}^{\mathsf{A}}$ are shortcuts for $\mathcal{A}(\varphi(z), A_S, n, \mathcal{D})$ and $\mathsf{Adv}^{\mathsf{A}}(\mathcal{A}, A, n, \mathcal{D})$, respectively.

This definition has the side effect of incentivizing the adversary to "game the system" by performing poorly when it thinks that $b = 1$. To remove this incentive, one may consider using a simulator $\mathcal{S}$, which does not receive the model as an input, when $b = 1$. This definition is formalized below:

Definition 6 (Alternative attribute advantage)**.** *Let*

$$\mathcal{S}(\varphi(z), n, \mathcal{D}) = \arg\max_{t_i} \Pr_{z \sim \mathcal{D}}[\pi(z) = t_i \mid \varphi(z)]$$

be the Bayes optimal simulator. The attribute advantage *of $\mathcal{A}$ can alternatively be defined as*

$$\mathsf{Adv}^{\mathsf{A}}_{\mathcal{S}}(\mathcal{A}, A, n, \mathcal{D}) = \Pr[\mathcal{A}(\varphi(z), A_S, n, \mathcal{D}) = \pi(z) \mid b = 0] - \Pr[\mathcal{S}(\varphi(z), n, \mathcal{D}) = \pi(z) \mid b = 1].$$

One potential issue with this alternative definition is that higher model accuracy will lead to higher attribute advantage *regardless of how accurate the model is for the general population*. Broadly, there are two ways for a model to perform better on the training data: it can overfit to the training data, or it can learn a general trend in the distribution $\mathcal{D}$. In this paper, we concern ourselves with the view that the adversary's ability to infer the target $\pi(z)$ in the latter case is due not to the model but pre-existing patterns in $\mathcal{D}$. To allow capturing the difference between overfitting and learning a general trend, we use Definition 5 in the following analysis and leave a more complete exploration of Definition 6 as future work. While adversaries that "game the system" may seem problematic, the effectiveness of such adversaries is indicative of privacy loss because their existence implies the ability to infer membership, as demonstrated by Reduction Adversary 5 in Section V-A.

A. Inversion, generalization, and influence

The case where φ simply removes the sensitive attribute t from the data point $z = (v, t, y)$ such that $\varphi(z) = (v, y)$ is known in the literature as *model inversion* [3], [4], [5], [8].

In this section, we look at the model inversion attack of Fredrikson et al. [4] under the advantage given in Definition 5. We point out that this is a novel analysis, as this advantage is defined to reflect the extent to which an attribute inference attack reveals information about individuals in S. While prior work [3], [4] has empirically evaluated attribute accuracy over corresponding training and test sets, our goal is to analyze the factors that lead to increased privacy risk specifically for members of the training data. To that end, we illustrate the relationship between advantage and generalization error as we did in the case of membership inference (Section III-B). We also explore the role of feature influence, which in this case corresponds to the degree to which changes to a sensitive feature of x affects the value $A_S(x)$. In Section VI-C, we show that the formal relationships described here often extend to attacks on real data where formal assumptions may fail to hold.

The attack described by Fredrikson et al. [4] is intended for linear regression models and is thus subject to the Gaussian

error assumption discussed in Section III-B. In general, when the adversary can approximate the error distribution reasonably well, e.g., by assuming a Gaussian distribution whose standard deviation equals the published standard error value, it can gain advantage by trying all possible values of the sensitive attribute. We denote the adversary's approximation of the error distribution by $f_{\mathcal{A}}$, and we assume that the target $t = \pi(z)$ is drawn from a finite set of possible values $t_1, \ldots, t_m$ with known frequencies in $\mathcal{D}$. We indicate the other features, which are known by the adversary, with the letter v (i.e., $z = (x, y)$, $x = (v, t)$, and $\varphi(z) = (v, y)$). The attack is shown in Adversary 4. For each t_i, the adversary counterfactually assumes that $t = t_i$ and computes what the error of the model would be. It then uses this information to update the a priori marginal distribution of t and picks the value t_i with the greatest likelihood.

Adversary 4 (General)**.** *Let $f_{\mathcal{A}}(\epsilon)$ be the adversary's guess for the probability density of the error $\epsilon = y - A_S(x)$. On input v, y, A_S, n, and $\mathcal{D}$, the adversary proceeds as follows:*

1) *Query the model to get $A_S(v, t_i)$ for all $i \in [m]$.*
2) *Let $\epsilon(t_i) = y - A_S(v, t_i)$.*
3) *Return the result of* $\arg\max_{t_i}(\Pr_{z\sim\mathcal{D}}[t = t_i] \cdot f_{\mathcal{A}}(\epsilon(t_i)))$.

When analyzing Adversary 4, we are clearly interested in the effect that generalization error will have on advantage. Given the results of Section III-B, we can reasonably expect that large generalization error will lead to greater advantage. However, as pointed out by Wu et al. [5], the functional relationship between t and $A_S(v, t)$ may play a role as well. Working in the context of models as Boolean functions, Wu et al. formalized the relevant property as *functional influence* [20], which is the probability that changing t will cause $A_S(v, t)$ to change when v is sampled uniformly.

The attack considered here applies to linear regression models, and Boolean influence is not suitable for use in this setting. However, an analogous notion of influence that characterizes the magnitude of change to $A_S(v, t)$ is relevant to attribute inference. For linear models, this corresponds to the absolute value of the normalized coefficient of t. Throughout the rest of the paper, we refer to this quantity as the influence of t without risk of confusion with the Boolean influence used in other contexts.

a) Binary Variable with Uniform Prior: The first part of our analysis deals with the simplest case where $m = 2$ with $\Pr_{z\sim\mathcal{D}}[t = t_1] = \Pr_{z\sim\mathcal{D}}[t = t_2]$. Without loss of generality we assume that $A_S(v, t_1) = A_S(v, t_2) + \tau$ for some fixed $\tau \geq 0$, so in this setting τ is a straightforward proxy for influence. Theorem 5 relates the advantage of Adversary 4 to σ_S, $\sigma_{\mathcal{D}}$, and τ.

Theorem 5. *Let t be drawn uniformly from $\{t_1, t_2\}$ and suppose that $y = A_S(v, t) + \epsilon$, where $\epsilon \sim N(0, \sigma_S^2)$ if $b = 0$ and $\epsilon \sim N(0, \sigma_{\mathcal{D}}^2)$ if $b = 1$. Then the advantage of Adversary 4 is $\frac{1}{2}(\mathrm{erf}(\tau/2\sqrt{2}\sigma_S) - \mathrm{erf}(\tau/2\sqrt{2}\sigma_{\mathcal{D}}))$.*

Proof. Given the assumptions made in this setting, we can describe the behavior of $\mathcal{A}$ as returning the value t_i that minimizes $|\epsilon(t_i)|$. If $t = t_1$, it is easy to check that $\mathcal{A}$ guesses correctly if and only if $\epsilon(t_1) > -\tau/2$. This means that $\mathcal{A}$'s advantage given $t = t_1$ is

$$\begin{aligned}
&\Pr[\mathcal{A} = t_1 \mid t = t_1, b = 0] - \Pr[\mathcal{A} = t_1 \mid t = t_1, b = 1] \\
&= \Pr[\epsilon(t_1) > -\tau/2 \mid b = 0] - \Pr[\epsilon(t_1) > -\tau/2 \mid b = 1] \\
&= \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_S}\right)\right) - \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_{\mathcal{D}}}\right)\right) \\
&= \frac{1}{2}\left(\mathrm{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_S}\right) - \mathrm{erf}\left(\frac{\tau}{2\sqrt{2}\sigma_{\mathcal{D}}}\right)\right)
\end{aligned} \tag{5}$$

Similar reasoning shows that $\mathcal{A}$'s advantage given $t = t_2$ is exactly the same, so the theorem follows from Equation 4. □

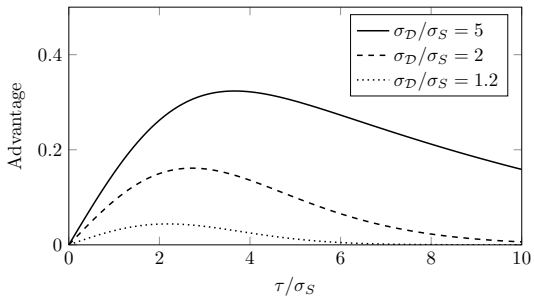


Fig. 1: The advantage of Adversary 4 as a function of t's influence τ. Here t is a uniformly distributed binary variable.

Clearly, the advantage will be zero when there is no generalization error ($\sigma_S = \sigma_{\mathcal{D}}$). Consider the other extreme case where $\sigma_S \to 0$ and $\sigma_{\mathcal{D}} \to \infty$. When σ_S is very small, the adversary will always guess correctly because the influence of t overwhelms the effect of the error ϵ. On the other hand, when $\sigma_{\mathcal{D}}$ is very large, changes to t will be nearly imperceptible for "normal" values of τ, and the adversary is reduced to random guessing. Therefore, the maximum possible advantage with uniform prior is $1/2$. As a model overfits more, σ_S decreases and $\sigma_{\mathcal{D}}$ tends to increase. If τ remains fixed, it is easy to see that the advantage increases monotonically under these circumstances.

Figure 1 shows the effect of changing τ as the ratio $\sigma_{\mathcal{D}}/\sigma_S$ remains fixed at several different constants. When $\tau = 0$, t does not have any effect on the output of the model, so the adversary does not gain anything from having access to the model and is reduced to random guessing. When τ is large, the adversary almost always guesses correctly regardless of the value of b since the influence of t drowns out the error noise. Thus, at both extremes the advantage approaches 0, and the adversary is able to gain advantage only when τ and $\sigma_{\mathcal{D}}/\sigma_S$ are in balance.

b) General Case: Sometimes the uniform prior for t may not be realistic. For example, t may represent whether a patient has a rare disease. In this case, we weight the values of $f_{\mathcal{A}}(\epsilon(t_i))$ by the a priori probability $\Pr_{z\sim\mathcal{D}}[t = t_i]$ before comparing which t_i is the most likely. With uniform prior, we could simplify $\arg\max_{t_i} f_{\mathcal{A}}(\epsilon(t_i))$ to $\arg\min_{t_i} |\epsilon(t_i)|$ regardless of the value of σ used for $f_{\mathcal{A}}$. On the other hand, the

value of σ matters when we multiply by $\Pr[t = t_i]$. Because the adversary is not given b, it makes an assumption similar to that described in Section III-B and uses $\epsilon \sim N(0, \sigma_S^2)$.

Clearly $\sigma_S = \sigma_{\mathcal{D}}$ results in zero advantage. The maximum possible advantage is attained when $\sigma_S \to 0$ and $\sigma_{\mathcal{D}} \to \infty$. Then, by similar reasoning as before, the adversary will always guess correctly when $b = 0$ and is reduced to random guessing when $b = 1$, resulting in an advantage of $1 - \frac{1}{m}$.

In general, the advantage can be computed using Equation 4. We first figure out when the adversary outputs t_i. When $f_{\mathcal{A}}$ is a Gaussian, this is not computationally intensive as there is at most one decision boundary between any two values t_i and t_j. Then, we convert the decision boundaries into probabilities by using the error distributions $\epsilon \sim N(0, \sigma_S^2)$ and $N(0, \sigma_{\mathcal{D}}^2)$, respectively.

V. Connection between membership and attribute inference

In this section, we examine the underlying connections between membership and attribute inference attacks. Our approach is based on reduction adversaries that have oracle access to one type of attack and attempt to perform the other type of attack. We characterize the advantage of each reduction adversary in terms of the advantage of its oracle. In Section VI-C, we implement the most sophisticated of the reduction adversaries described here and show that on real data it performs remarkably well, often outperforming Attribute Adversary 4 by large margins. We note that these reductions are specific to our choice of attribute advantage given in Definition 5. Analyzing the connections between membership and attribute inference using the alternative Definition 6 is an interesting direction for future work.

A. From membership to attribute

We start with an adversary $\mathcal{A}_{\mathsf{M}\to\mathsf{A}}$ that uses an attribute oracle to accomplish membership inference. The attack, shown in Adversary 5, is straightforward: given a point z, the adversary queries the attribute oracle to obtain a prediction t of the target value $\pi(z)$. If this prediction is correct, then the adversary concludes that z was in the training data.

Adversary 5 (Membership $\to$ attribute)**.** *The reduction adversary $\mathcal{A}_{\mathsf{M}\to\mathsf{A}}$ has oracle access to attribute adversary $\mathcal{A}_{\mathsf{A}}$. On input z, A_S, n, and $\mathcal{D}$, the reduction adversary proceeds as follows:*

1) *Query the oracle to get $t \leftarrow \mathcal{A}_{\mathsf{A}}(\varphi(z), A_S, n, \mathcal{D})$.*
2) *Output 0 if $\pi(z) = t$. Otherwise, output 1.*

Theorem 6 shows that the membership advantage of this reduction exactly corresponds to the attribute advantage of its oracle. In other words, the ability to effectively infer attributes of individuals in the training set implies the ability to infer membership in the training set as well. This suggests that attribute inference is at least as difficult as than membership inference.

Theorem 6. *Let $\mathcal{A}_{\mathsf{M}\to\mathsf{A}}$ be the adversary described in Adversary 5, which uses $\mathcal{A}_{\mathsf{A}}$ as an oracle. Then,*

$$\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}_{\mathsf{M}\to\mathsf{A}}, A, n, \mathcal{D}) = \mathsf{Adv}^{\mathsf{A}}(\mathcal{A}_{\mathsf{A}}, A, n, \mathcal{D}).$$

Proof. The proof follows directly from the definitions of membership and attribute advantages.

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{M}} &= \Pr[\mathcal{A}_{\mathsf{M}\to\mathsf{A}} = 0 \mid b = 0] - \Pr[\mathcal{A}_{\mathsf{M}\to\mathsf{A}} = 0 \mid b = 1] \\
&= \sum_{t_i \in \mathbf{T}} \Pr[t = t_i](\Pr[\mathcal{A}_{\mathsf{M}\to\mathsf{A}} = 0 \mid b = 0, t = t_i] \\
&\qquad - \Pr[\mathcal{A}_{\mathsf{M}\to\mathsf{A}} = 0 \mid b = 1, t = t_i]) \\
&= \sum_{t_i \in \mathbf{T}} \Pr[t = t_i](\Pr[\mathcal{A}_{\mathsf{A}} = t_i \mid b = 0, t = t_i] \\
&\qquad - \Pr[\mathcal{A}_{\mathsf{A}} = t_i \mid b = 1, t = t_i]) \\
&= \mathsf{Adv}^{\mathsf{A}}.
\end{aligned}$$

□

B. From attribute to membership

We now consider reductions in the other direction, wherein the adversary is given $\varphi(z)$ and must reconstruct the point z to query the membership oracle. To accomplish this, we assume that the adversary knows a deterministic reconstruction function φ^{-1} such that $\varphi \circ \varphi^{-1}$ is the identity function, i.e., for any value of $\varphi(z)$ that the adversary may receive, there exists $z' = \varphi^{-1}(\varphi(z))$ such that $\varphi(z) = \varphi(z')$. However, because φ is a lossy function, in general it does not hold that $\varphi^{-1}(\varphi(z)) = z$. Our adversary, described in Adversary 6, reconstructs the point z', sets the attribute t of that point to value t_i chosen uniformly at random, and outputs t_i if the membership oracle says that the resulting point is in the dataset.

Adversary 6 (Uniform attribute $\to$ membership)**.** *Suppose that $t_1, \ldots, t_m$ are the possible values of the target $t = \pi(z)$. The reduction adversary $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ has oracle access to membership adversary $\mathcal{A}_{\mathsf{M}}$. On input $\varphi(z)$, A_S, n, and $\mathcal{D}$, the reduction adversary proceeds as follows:*

1) *Choose t_i uniformly at random from $\{t_1, \ldots, t_m\}$.*
2) *Let $z' = \varphi^{-1}(\varphi(z))$, and change the value of the sensitive attribute t such that $\pi(z') = t_i$.*
3) *Query $\mathcal{A}_{\mathsf{M}}$ to obtain $b' \leftarrow \mathcal{A}_{\mathsf{M}}(z', A_S, n, \mathcal{D})$.*
4) *If $b' = 0$, output t_i. Otherwise, output $\perp$.*

The uniform choice of t_i is motivated by the fact that the adversary may not know how the advantage of the membership oracle is distributed across different values of t. For example, it is possible that $\mathcal{A}_{\mathsf{M}}$ performs very poorly when $t = t_1$ and that all of its advantage comes from the case where $t = t_2$.

In the computation of the advantage, we only consider the case where $\pi(z) = t_i$ because this is the only case where the reduction adversary can possibly give the correct answer. In that case, the membership oracle is given a challenge point from the distribution $\mathcal{D}' = \{(x, y) \mid (x, y) = \varphi^{-1}(\varphi(z))$ except that $t = \pi(z)\}$, where $z \sim S$ if $b = 0$ and $z \sim \mathcal{D}$ if $b = 1$. On the other hand, the training set S used to train the model A_S was drawn from $\mathcal{D}$. Because of this difference, we use modified membership

advantage $\mathsf{Adv}^{\mathsf{M}}_{*}(\mathcal{A}, A, n, \mathcal{D}, \varphi, \varphi^{-1}, \pi)$, which measures the performance of the membership adversary when the challenge point is drawn from $\mathcal{D}'$. In the case of a model inversion attack as described in the beginning of Section IV-A, we have $\mathsf{Adv}^{\mathsf{M}}(\mathcal{A}, A, n, \mathcal{D}) = \mathsf{Adv}^{\mathsf{M}}_{*}(\mathcal{A}, A, n, \mathcal{D}, \varphi, \varphi^{-1}, \pi)$, i.e., the modified membership advantage equals the unmodified one.

Theorem 7 shows that the attribute advantage of $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ is proportional to the modified membership advantage of $\mathcal{A}_{\mathsf{M}}$, giving a lower bound on the effectiveness of attribute inference attacks that use membership oracles. Notably, the adversary does not make use of any associations that may exist between $\varphi(z)$ and t, so this reduction is general and works even when no such association exists. While the reduction does not completely transfer the membership advantage to attribute advantage, the resulting attribute advantage is within a constant factor of the modified membership advantage.

Theorem 7. *Let $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ be the adversary described in Adversary 6, which uses $\mathcal{A}_{\mathsf{M}}$ as an oracle. Then,*

$$\mathsf{Adv}^{\mathsf{A}}(\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}, A, n, \mathcal{D}) = \frac{1}{m}\mathsf{Adv}^{\mathsf{M}}_{*}(\mathcal{A}_{\mathsf{M}}, A, n, \mathcal{D}, \varphi, \varphi^{-1}, \pi).$$

Proof. We first give an informal argument. In order for $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ to correctly guess the value of t, it needs to choose the correct t_i, which happens with probability $\frac{1}{m}$, and then $\mathcal{A}_{\mathsf{M}}(z', A_S, n, \mathcal{D})$ must be 0. Therefore, $\mathsf{Adv}^{\mathsf{A}} = \frac{1}{m}\mathsf{Adv}^{\mathsf{M}}_{*}$.

Now we give the formal proof. Let t' be the value of t that was chosen independently and uniformly at random in Step 1 of Adversary 6. Since $\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}}$ outputs t_i if and only if $t' = t_i$ and $\mathcal{A}_{\mathsf{M}}(z') = 0$, we have

$$\begin{aligned}&\Pr[\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}} = t_i \mid b = 0, t = t_i]\\ &= \frac{1}{m}\Pr[\mathcal{A}_{\mathsf{M}}(z') = 0 \mid b = 0, t = t_i],\end{aligned}$$

and likewise when $b = 1$. Therefore, the advantage of the reduction adversary is

$$\begin{aligned}\mathsf{Adv}^{\mathsf{A}} &= \sum_{t_i\in\mathbf{T}} \Pr[t = t_i](\Pr[\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}} = t_i \mid b = 0, t = t_i]\\ &\qquad - \Pr[\mathcal{A}^{\mathsf{U}}_{\mathsf{A}\to\mathsf{M}} = t_i \mid b = 1, t = t_i])\\ &= \frac{1}{m}\sum_{t_i\in\mathbf{T}} \Pr[t = t_i](\Pr[\mathcal{A}_{\mathsf{M}}(z') = 0 \mid b = 0, t = t_i]\\ &\qquad - \Pr[\mathcal{A}_{\mathsf{M}}(z') = 0 \mid b = 1, t = t_i])\\ &= \frac{1}{m}(\Pr[\mathcal{A}_{\mathsf{M}}(z') = 0 \mid b = 0]\\ &\qquad - \Pr[\mathcal{A}_{\mathsf{M}}(z') = 0 \mid b = 1])\\ &= \frac{1}{m}\mathsf{Adv}^{\mathsf{M}}_{*},\end{aligned}$$

where the second-to-last step holds due to the fact that b and t are independent. □

Adversary 6 has the obvious weakness that it can only return correct answers when it guesses the value of t correctly. Adversary 7 attempts to improve on this by making multiple queries to $\mathcal{A}_{\mathsf{M}}$. Rather than guess the value of t, this adversary tries all values of t in order of their marginal probabilities until the membership adversary says "yes".

Adversary 7 (Multi-query attribute → membership)**.** *Suppose that $t_1, \ldots, t_m$ are the possible values of the sensitive attribute t. The reduction adversary $\mathcal{A}^{\mathsf{M}}_{\mathsf{A}\to\mathsf{M}}$ has oracle access to membership adversary $\mathcal{A}_{\mathsf{M}}$. On input $\varphi(z)$, A_S, n, and $\mathcal{D}$, $\mathcal{A}_{\mathsf{A}\to\mathsf{M}}$ proceeds as follows:*

1) *Let $z' = \varphi^{-1}(\varphi(z))$.*
2) *For all $i \in [m]$, let z'_i be z' with the value of the sensitive attribute t changed to t_i.*
3) *Query $\mathcal{A}_{\mathsf{M}}$ to compute $T = \{t_i \mid \mathcal{A}_{\mathsf{M}}(z'_i, A_S, n, \mathcal{D}) = 0\}$.*
4) *Output $\arg\max_{t_i\in T} \Pr_{z\sim\mathcal{D}}[t = t_i]$. If $T = \emptyset$, output $\perp$.*

We evaluate this adversary experimentally in Section VI-C.

VI. Evaluation

In this section, we evaluate the performance of the adversaries discussed in Sections III, IV, and V. We compare the performance of these adversaries on real datasets with the analysis from previous sections and show that overfitting predicts privacy risk in practice as our analysis suggests. Our experiments use linear regression, tree, and deep convolutional neural network (CNN) models.

A. Methodology

1) Linear and tree models: We used the Python scikit-learn [33] library to calculate the empirical error R_{emp} and the leave-one-out cross validation error R_{cv} [34]. Because these two measures pertain to the error of the model on points inside and outside the training set, respectively, they were used to approximate σ_S and $\sigma_{\mathcal{D}}$, respectively. Then, we made a random 75-25% split of the data into training and test sets. The training set was used to train either a Ridge regression or a decision tree model, and then the adversaries were given access to this model. We repeated this 100 times with different training-test splits and then averaged the result. Before we explain the results, we describe the datasets.

Eyedata. This is gene expression data from rat eye tissues [35], as presented in the "flare" package of the R programming language. The inputs and the outputs are respectively stored in R as a 120×200 matrix and a 120-dimensional vector of floating-point numbers. We used scikit-learn [33] to scale each attribute to zero mean and unit variance.

IWPC. This is data collected by the International Warfarin Pharmacogenetics Consortium [36] about patients who were prescribed warfarin. After we removed rows with missing values, 4819 patients remained in the dataset. The inputs to the model are demographic (age, height, weight, race), medical (use of amiodarone, use of enzyme inducer), and genetic (VKORC1, CYP2C9) attributes. Age, height, and weight are real-valued and were scaled to zero mean and unit variance. The medical attributes take binary values, and the remaining attributes were one-hot encoded. The output is the weekly dose of warfarin in milligrams. However, because the distribution of warfarin dose is skewed, IWPC concludes in [36] that solving for the square root of the dose results in a more predictive

linear model. We followed this recommendation and scaled the *square root* of the dose to zero mean and unit variance.

Netflix. We use the dataset from the Netflix Prize contest [37]. This is a sparse dataset that indicates when and how a user rated a movie. For the output attribute, we used the rating of *Dragon Ball Z: Trunks Saga*, which had one of the most polarized rating distributions. There are 2416 users who rated this, and the ratings were scaled to zero mean and unit variance. The input attributes are binary variables indicating whether or not a user rated each of the other 17,769 movies in the dataset.

2) Deep convolutional neural networks: We evaluated the membership inference attack on deep CNNs. In addition, we implemented the colluding training algorithm (Algorithm 1) to verify its performance in practice. The CNNs were trained in Python using the Keras deep-learning library [38] and a standard stochastic gradient descent algorithm [39]. We used three datasets that are standard benchmarks in the deep learning literature and were evaluated in prior work on inference attacks [7]; they are described in more detail below. For all datasets, pixel values were normalized to the range $[0, 1]$, and the label values were encoded as one-hot vectors. To expedite the training process across a range of experimental configurations, we used a subset of each dataset. For each dataset, we randomly divided the available data into equal-sized training and test sets to facilitate comparison with prior work [7] that used this convention.

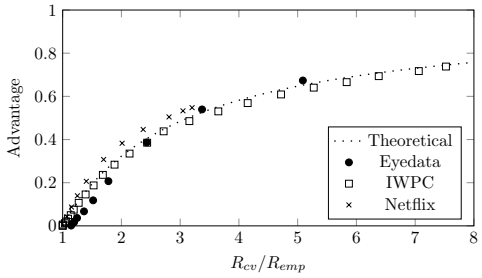
The architecture we use is based on the VGG network [40], which is commonly used in computer vision applications. We control for generalization error by varying a size parameter s that defines the number of units at each layer of the network. The architecture consists of two 3x3 convolutional layers with s filters each, followed by a 2x2 max pooling layer, two 3x3 convolutional layers with $2s$ filters each, a 2x2 max pooling layer, a fully-connected layer with $2s$ units, and a softmax output layer. All activation functions are rectified linear. We chose $s = 2^i$ for $0 \leq i \leq 7$, as we did not observe qualitatively different results for larger values of i. All training was done using the Adam optimizer [41] with the default parameters in the Keras implementation ($\lambda = 0.001$, $\beta_1 = 0.5$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$, and decay set to 5×10^{-4}). We used categorical cross-entropy loss, which is conventional for models whose topmost activation is softmax [39].

MNIST. MNIST [42] consists of 70,000 images of handwritten digits formatted as grayscale 28×28-pixel images, with class labels indicating the digit depicted in each image. We selected 17,500 points from the full dataset at random for our experiments.

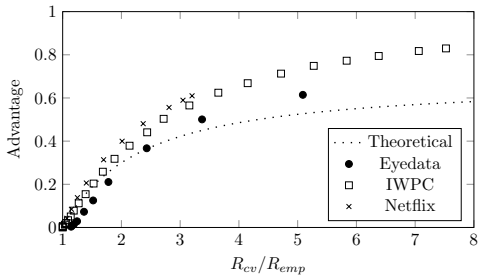
CIFAR-10, CIFAR-100. The CIFAR datasets [43] consist of 60,000 32×32-pixel color images, labeled as 10 (CIFAR-10) and 100 (CIFAR-100) classes. We selected 15,000 points at random from the full data.

B. Membership inference

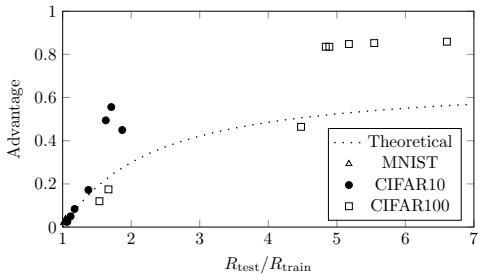
The results of the membership inference attacks on linear and tree models are plotted in Figures 2a and 2b. The theoretical and experimental results appear to agree when the adversary knows both σ_S and $\sigma_{\mathcal{D}}$ and sets the decision boundary accordingly. However, when the adversary does not know $\sigma_{\mathcal{D}}$, it performs much better than what the theory predicts. In fact, an adversary can sometimes do better by just fixing the decision boundary at $|\epsilon| = \sigma_S$ instead of taking $\sigma_{\mathcal{D}}$ into account. This is because training set error distributions of overfitted models tend to have a higher peak at zero than a Gaussian. As a result, it is often advantageous to bring the decision boundaries closer to zero.



(a) Regression and tree models assuming knowledge of σ_S and $\sigma_{\mathcal{D}}$.



(b) Regression and tree models assuming knowledge of σ_S only.



(c) Deep CNNs assuming knowledge of average training loss L_S.

Fig. 2: Empirical membership advantage of the threshold adversary (Adversary 2) given as a function of generalization ratio for regression, tree, and CNN models.

The results of the threshold adversary on CNNs are given in Figure 2c. Although these models perform classification, the loss function used for training is categorical cross-entropy, which is non-negative, continuous, and unbounded. This suggests that the threshold adversary could potentially work in this setting as well. Specifically, the predictions made by these models can be compared against L_S, the average training

	Our work	Shokri et al. [7]
Attack complexity	Makes only one query to the model	Must train hundreds of shadow models
Required knowledge	Average training loss L_S	Ability to train shadow models, e.g., input distribution and type of model
Precision	0.505 (MNIST) 0.694 (CIFAR-10) 0.874 (CIFAR-100)	0.517 (MNIST) 0.72-0.74 (CIFAR-10) > 0.99 (CIFAR-100)
Recall	> 0.99	> 0.99

TABLE I: Comparison of our membership inference attack with that presented by Shokri et al. While our attack has slightly lower precision, it requires far less computational resources and background knowledge.

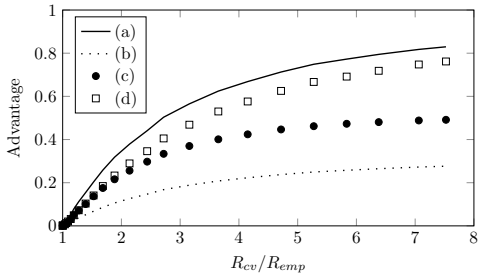
loss observed during training, which is often reported with published architectures as a point of comparison against prior work (see, for example, [44] and [45, Figures 3 and 4]). Figure 2c shows that, while the empirical results do not match the theoretical curve as closely as do linear and tree models, they do not diverge as much as one might expect given that the error is not Gaussian as assumed by Theorem 3.

Now we compare our attack with that by Shokri et al. [7], which generates "shadow models" that are intended to mimic the behavior of A_S. Because their attack involves using machine learning to train the attacker with the shadow models, their attack requires considerable computational power and knowledge of the algorithm used to train the model. By contrast, our attacker simply makes one query to the model and needs to know only the average training loss. Despite these differences, when the size parameter s is set equal to that used by Shokri et al., our attacker has the same recall and only slightly lower precision than their attacker. A more detailed comparison is given in Table I.

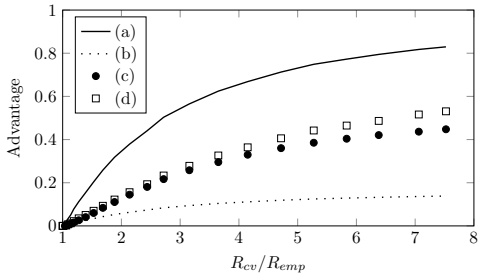
C. Attribute inference and reduction

We now present the empirical attribute advantage of the general adversary (Adversary 4). Because this adversary uses the model inversion assumptions described at the beginning of Section IV-A, our evaluation is also in the setting of model inversion. For these experiments we used the IWPC and Netflix datasets described in Section VI-A. For $f_{\mathcal{A}}(\epsilon)$, the adversary's approximation of the error distribution, we used the Gaussian with mean zero and standard deviation R_{emp}. For the IWPC dataset, each of the genomic attributes (VKORC1 and CYP2C9) is separately used as the target t. In the Netflix dataset, the target attribute was whether a user rated a certain movie, and we randomly sampled targets from the set of available movies.

The circles in Figure 3 show the result of inverting the VKORC1 and CYP2C9 attributes in the IWPC dataset. Although the attribute advantage is not as high as the membership advantage (solid line), the attribute adversary exhibits a sizable advantage that increases as the model overfits more and more. On the other hand, none of the attacks could effectively infer whether a user watched a certain movie in the Netflix dataset. In addition, we were unable to simultaneously control for both



(a) $t =$ VKORC1



(b) $t =$ CYP2C9

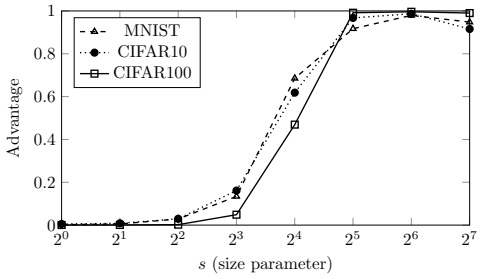
Fig. 3: Experimentally determined advantage for various membership and attribute adversaries. The plots correspond to: (a) threshold membership adversary (Adversary 2), (b) uniform reduction adversary (Adversary 6), (c) general attribute adversary (Adversary 4), and (d) multi-query reduction adversary (Adversary 7). Both reduction adversaries use the threshold membership adversary as the oracle, and $f_{\mathcal{A}}(\epsilon)$ for the attribute adversary is the Gaussian with mean zero and standard deviation σ_S.

$\sigma_{\mathcal{D}}/\sigma_S$ and τ in the Netflix dataset to measure the effect of influence as predicted by Theorem 5.

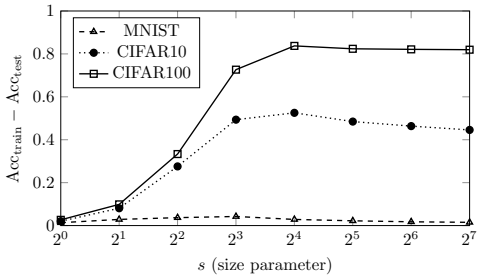
Finally, we evaluate the performance of the multi-query reduction adversary (Adversary 7). As the squares in Figure 3 show, with the IWPC data, making multiple queries to the membership oracle significantly increased the success rate compared to what we would expect from the naive uniform reduction adversary (Adversary 6, dotted line). Surprisingly, the reduction is also more effective than running the attribute inference attack directly. By contrast, with the Netflix data, the multi-query reduction adversary was often slightly worse than the naive uniform adversary although it still outperformed direct attribute inference.

D. Collusion in membership inference

We evaluate A^{C} and $\mathcal{A}^{\mathsf{C}}$ described in Section III-D for CNNs trained as image classifiers. To instantiate F_K and G_K, we use Python's intrinsic pseudorandom number generator with key K as the seed. We note that our proof of Theorem 4 relies only on the uniformity of the pseudorandom numbers and not on their unpredictability. Deviations from this assumption will result in a less effective membership inference attack but do not invalidate our results. All experiments set the number of keys to $k = 3$.



(a) Advantage as a function of network size for $\mathcal{A}^{\mathsf{C}}$ with $k = 3$. For $s \geq 16$, CIFAR-10 and MNIST achieve advantage at least 0.9 (precision ≥ 0.9, recall ≥ 0.99), whereas CIFAR-100 achieves advantage 0.98 (precision ≥ 0.99, recall ≥ 0.99).



(b) Generalization error measured as the difference between training and test accuracy. On MNIST, the maximum was achieved at $s = 8$ at 0.05, while for CIFAR-10 the maximum was 0.52 ($s = 16$), and 0.82 ($s = 16$) for CIFAR-100.

Fig. 4: Results of colluding training algorithm and membership adversary on CNNs trained on MNIST, CIFAR-10, and CIFAR-100. The size parameter was configured to take values $s = 2^i$ for $i \in [0, 7]$. Regardless of the models' generalization performance, when the network is sufficiently large, the attack achieves high advantage (≥ 0.98) without affecting predictive accuracy.

The results of our experiment are shown in Figures 4a and 4b. The data shows that on all three instances, the colluding parties achieve a high membership advantage without significantly affecting model performance. The accuracy of the subverted model was only 0.014 (MNIST), 0.047 (CIFAR-10), and 0.031 (CIFAR-100) less than that of the unsubverted model. The advantage rapidly increases with the model size around $s \approx 16$ but is relatively constant elsewhere, indicating that model capacity beyond a certain point is a necessary factor in the attack.

Importantly, the results demonstrate that specific information about nearly all of the training data can be intentionally leaked through the behavior of a model that appears to generalize very well. In fact, looking at Figure 4b shows that in these instances, there is no discernible relationship between generalization error and membership advantage. The three datasets exhibit vastly different generalization behavior, with the MNIST models achieving almost no generalization error (< 0.02 for $s \geq 32$) and CIFAR-100 showing a large performance gap (≥ 0.8 for $s \geq 32$). Despite this fact, the membership adversary achieves nearly identical performance.

VII. Related Work

A. Privacy and statistical summaries

There is extensive prior literature on privacy attacks on statistical summaries. Komarova et al. [46] looked into partial disclosure scenarios, where an adversary is given fixed statistical estimates from combined public and private sources and attempts to infer the sensitive feature of an individual referenced in those sources. A number of previous studies [21], [22], [47], [48], [49], [50] have looked into membership attacks from statistics commonly published in genome-wide association studies (GWAS). Calandrino et al. [51] showed that temporal changes in recommendations given by collaborative filtering methods can reveal the inputs that caused those changes. Linear reconstruction attacks [52], [53], [54] attempt to infer partial inputs to linear statistics and were later extended to non-linear statistics [55]. While the goal of these attacks has commonalities with both membership inference and attribute inference, our results apply specifically to machine learning settings where generalization error and influence make our results relevant.

B. Privacy and machine learning

More recently, others have begun examining these attacks in the context of machine learning. Ateniese et al. [1] showed that the knowledge of the internal structure of Support Vector Machines and Hidden Markov Models leaks certain types of information about their training data, such as the language used in a speech dataset.

Dwork et al. [13] showed that a differentially private algorithm with a suitably chosen parameter generalizes well with high probability. Subsequent work showed that similar results are true under related notions of privacy. In particular, Bassily et al. [18] studied a notion of privacy called total variation stability and proved good generalization with respect to a bounded number of adaptively chosen low-sensitivity queries. Moreover, for data drawn from Gibbs distributions, Wang et al. [19] showed that on-average KL privacy is equivalent to generalization error as defined in this paper. While these results give evidence for the relationship between privacy and overfitting, we construct an attacker that directly leverages overfitting to gain advantage commensurate with the extent of the overfitting.

1) Membership inference: Shokri et al. [7] developed a membership inference attack and applied it to popular machine-learning-as-a-service APIs. Their attacks are based on "shadow models" that approximate the behavior of the model under attack. The shadow models are used to build another machine learning model called the "attack model", which is trained to distinguish points in the training data from other points based on the output they induce on the original model under attack. As we discussed in Section VI-B, our simple threshold adversary comes surprisingly close to the accuracy

of their attack, especially given the differences in complexity and requisite adversarial assumptions between the attacks.

Because the attack proposed by Shokri et al. itself relies on machine learning to find a function that separates training and non-training points, it is not immediately clear why the attack works, but the authors hypothesize that it is related to overfitting and the "diversity" of the training data. They graph the generalization error against the precision of their attack and find some evidence of a relationship, but they also find that the relationship is not perfect and conclude that model structure must also be relevant. The results presented in this paper make the connection to overfitting precise in many settings, and the colluding training algorithm we give in Section VI-D demonstrates exactly how model structure can be exploited to create a membership inference vulnerability.

Li et al. [6] explored membership inference, distinguishing between "positive" and "negative" membership privacy. They show how this framework defines a family of related privacy definitions that are parametrized on distributions of the adversary's prior knowledge, and they find that a number of previous definitions can be instantiated in this way.

2) Attribute inference: Practical model inversion attacks have been studied in the context of linear regression [4], [8], decision trees [3], and neural networks [3]. Our results apply to these attacks when they are applied to data that matches the distributional assumptions made in our analysis. An important distinction between the way inversion attacks were considered in prior work and how we treat them here is the notion of advantage. Prior work on these attacks defined advantage as the difference between the attacker's predictive accuracy given the model and the best accuracy that could be achieved without the model. Although some prior work [3], [4] empirically measured this advantage on both training and test datasets, this definition does not allow a formal characterization of how exposed the *training data specifically* is to privacy risk. In Section IV, we define attribute advantage precisely to capture the risk to the training data by measuring the difference in the attacker's accuracy on training and test data: the advantage is zero when the attack is as powerful on the general population as on the training data and is maximized when the attack works *only* on the training data.

Wu et al. [5] formalized model inversion for a simplified class of models that consist of Boolean functions and explored the initial connections between influence and advantage. However, as in other prior work on model inversion, the type of advantage that they consider says nothing about what the model specifically leaks about its training data. Drawing on their observation that influence is relevant to privacy risk in general, we illustrate its effect on the notion of advantage defined in this paper and show how it interacts with generalization error.

VIII. Conclusion and Future Directions

We introduced new formal definitions of advantage for membership and attribute inference attacks. Using these definitions, we analyzed attacks under various assumptions on learning algorithms and model properties, and we showed that these two attacks are closely related through reductions in both directions. Both theoretical and experimental results confirm that models become more vulnerable to both types of attacks as they overfit more. Interestingly, our analysis also shows that overfitting is not the only factor that can lead to privacy risk: Theorem 4 shows that even stable learning algorithms, which provably do not overfit, can leak precise membership information, and the results in Section IV-A demonstrate that the influence of the target attribute on a model's output plays a key role in attribute inference.

Our formalization and analysis open interesting directions for future work. The membership attack in Theorem 4 is based on a colluding pair of adversary and learning rule, A^{C} and $\mathcal{A}^{\mathsf{C}}$. This could be implemented, for example, by a malicious ML algorithm provided by a third-party library or cloud service to subvert users' privacy. Further study of this scenario, which may best be formalized in the framework of algorithm substitution attacks [23], is warranted to determine whether malicious algorithms can produce models that are indistinguishable from normal ones and how such attacks can be mitigated.

Our results in Section III-A give bounds on membership advantage when certain conditions are met. These bounds apply to adversaries who may target specific individuals, bringing arbitrary background knowledge of their targets to help determine their membership status. Some types of realistic adversaries may be motivated by concerns that incentivize learning a limited set of facts about as many individuals in the training data as possible rather than obtaining unique background knowledge about specific individuals. Characterizing these "stable adversaries" is an interesting direction that may lead to tighter bounds on advantage or relaxed conditions on the learning rule.

References

[1] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, Sep. 2015.

[2] G. Cormode, "Personal privacy vs population privacy: Learning to attack anonymization," in *KDD*, 2011.

[3] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *ACM Conference on Computer and Communications Security (CCS)*, 2015.

[4] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *USENIX Security Symposium*, 2014, pp. 17–32.

[5] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A methodology for formalizing model-inversion attacks," in *2016 IEEE Computer Security Foundations Symposium (CSF)*, 2016.

[6] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: A unifying framework for privacy definitions," in *Proceedings of ACM CCS*, 2013.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (Oakland)*, 2017, pp. 3–18.

[8] X. Wu, M. Fredrikson, W. Wu, S. Jha, and J. F. Naughton, "Revisiting Differentially Private Regression: Lessons From Learning Theory and their Consequences," *CoRR*, vol. abs/1512.06388, 2015.

[9] J. Brickell and V. Shmatikov, "The cost of privacy: destruction of data-mining utility in anonymized data publishing," in *KDD*, 2008.

[10] J. Lei, "Differentially private m-estimators," in *NIPS*, 2011.

[11] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," in *VLDB*, 2012.

[12] A. G. Thakurta and A. Smith, "Differentially private feature selection via stability arguments, and the robustness of the lasso," in *Proceedings of the 26th Annual Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 30. PMLR, 12–14 Jun 2013, pp. 819–850.

[13] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth, "Preserving statistical validity in adaptive data analysis," in *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: ACM, 2015, pp. 117–126.

[14] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth, "Generalization in adaptive data analysis and holdout reuse," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 2350–2358.

[15] Y.-X. Wang, J. Lei, and S. E. Fienberg, "Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle," *Journal of Machine Learning Research*, vol. 17, no. 183, pp. 1–40, 2016.

[16] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *In 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2014.

[17] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, 2011.

[18] R. Bassily, K. Nissim, A. Smith, T. Steinke, U. Stemmer, and J. Ullman, "Algorithmic stability for adaptive data analysis," in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016, pp. 1046–1059.

[19] Y.-X. Wang, J. Lei, and S. E. Fienberg, "On-average KL-privacy and its equivalence to generalization for max-entropy mechanisms," in *International Conference on Privacy in Statistical Databases*, 2016, pp. 121–134.

[20] R. O'Donnell, *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[21] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genetics*, vol. 4, no. 8, 2008.

[22] S. Sankararaman, G. Obozinski, M. I. Jordan, and E. Halperin, "Genomic privacy and limits of individual detection in a pool," *Nature Genetics*, vol. 41, no. 9, pp. 965–967, 2009.

[23] M. Bellare, K. G. Paterson, and P. Rogaway, "Security of symmetric encryption against mass surveillance," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, 2014, pp. 1–19.

[24] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 2017, pp. 587–601.

[25] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[26] C. Dwork, "Differential privacy," in *ICALP*. Springer, 2006.

[27] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Learnability, stability and uniform convergence," *Jouranal of Machine Learing. Research*, vol. 11, Dec. 2010.

[28] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," *CoRR*, vol. abs/1709.01604, 2017.

[29] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, vol. abs/1611.03530, 2016.

[30] I. Giacomelli, R. F. Olimid, and S. Ranellucci, "Security of linear secret-sharing schemes against mass surveillance," in *Cryptology and Network Security - 14th International Conference, CANS 2015, Marrakesh, Morocco, December 10-12, 2015, Proceedings*, 2015, pp. 43–58.

[31] G. Ateniese, B. Magri, and D. Venturi, "Subversion-resilient signature schemes," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, 2015, pp. 364–375.

[32] M. Bellare, J. Jaeger, and D. Kane, "Mass-surveillance without the state: Strongly undetectable algorithm-substitution attacks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, 2015, pp. 1431–1440.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[34] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.

[35] T. E. Scheetz, K.-Y. A. Kim, R. E. Swiderski, A. R. Philp, T. A. Braun, K. L. Knudtson, A. M. Dorrance, G. F. DiBona, J. Huang, T. L. Casavant, V. C. Sheffield, and E. M. Stone, "Regulation of gene expression in the mammalian eye and its relevance to eye disease," *Proceedings of the National Academy of Sciences*, vol. 103, no. 39, pp. 14 429–14 434, 2006.

[36] International Warfarin Pharmacogenetics Consortium, "Estimation of the warfarin dose with clinical and pharmacogenetic data," *New England Journal of Medicine*, vol. 360, no. 8, pp. 753–764, 2009.

[37] Netflix, "Netflix prize," http://netflixprize.com, 2006.

[38] F. Chollet, "Keras: Deep learning library for Theano and TensorFlow," https://keras.io, 2017.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations (ICLR)*, 2015.

[42] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.

[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[44] B. Neuberg, "Personal photos model," https://github.com/BradNeuberg/personal-photos-model, 2017.

[45] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, "Data-dependent initializations of convolutional neural networks," *CoRR*, vol. abs/1511.06856, 2015. [Online]. Available: http://arxiv.org/abs/1511.06856

[46] T. Komarova, D. Nekipelov, and E. Yakovlev, "Estimation of treatment effects from combined data: Identification versus data security," in *Economic Analysis of the Digital Economy*. University of Chicago Press, 2015, pp. 279–308.

[47] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou, "Learning your identity and disease from research papers: information leaks in genome wide association studies," in *CCS*, 2009.

[48] K. El Emam, E. Jonker, L. Arbuckle, and B. Malin, "A systematic review of re-identification attacks on health data," *PLOS ONE*, vol. 6, no. 12, pp. 1–12, 12 2011.

[49] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, no. 6117, pp. 321–324, 2013.

[50] S. S. Shringarpure and C. D. Bustamante, "Privacy risks from genomic data-sharing beacons," *The American Journal of Human Genetics*, vol. 97, no. 5, pp. 631–646, May 2015.

[51] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, ""You might also like:" Privacy risks of collaborative filtering," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy (Oakland)*, 2011.

[52] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *PODS*, 2003.

[53] C. Dwork, F. McSherry, and K. Talwar, "The price of privacy and the limits of LP decoding," in *STOC*, 2007.

[54] S. P. Kasiviswanathan, M. Rudelson, A. Smith, and J. Ullman, "The price of privately releasing contingency tables and the spectra of random matrices with correlated rows," in *STOC*, 2010.

[55] S. P. Kasiviswanathan, M. Rudelson, and A. Smith, "The power of linear reconstruction attacks," in *SODA*, 2013.