Machine Learning with Membership Privacy using Adversarial Regularization

Milad Nasr[1], Reza Shokri[2], Amir Houmansadr[1]
[1] University of Massachusetts Amherst, [2] National University of Singapore
milad@cs.umass.edu, reza@comp.nus.edu.sg, amir@cs.umass.edu

ABSTRACT

Machine learning models leak information about the datasets on which they are trained. An adversary can build an algorithm to trace the individual members of a model's training dataset. As a fundamental inference attack, he aims to distinguish between data points that were part of the model's training set and any other data points from the same distribution. This is known as the tracing (and also membership inference) attack. In this paper, we focus on such attacks against black-box models, where the adversary can only observe the output of the model, but not its parameters. This is the current setting of machine learning as a service in the Internet.

We introduce a privacy mechanism to train machine learning models that provably achieve membership privacy: the model's predictions on its training data are indistinguishable from its predictions on other data points from the same distribution. We design a strategic mechanism where the privacy mechanism anticipates the membership inference attacks. The objective is to train a model such that not only does it have the minimum prediction error (high utility), but also it is the most robust model against its corresponding strongest inference attack (high privacy). We formalize this as a *min-max game* optimization problem, and design an adversarial training algorithm that minimizes the classification loss of the model as well as the maximum gain of the membership inference attack against it. This strategy, which guarantees membership privacy (as prediction indistinguishability), acts also as a strong regularizer and significantly generalizes the model.

We evaluate our privacy mechanism on deep neural networks using different benchmark datasets. We show that our min-max strategy can mitigate the risk of membership inference attacks (close to the random guess) with a negligible cost in terms of the classification error.

KEYWORDS

Data privacy; Machine learning; Inference attacks; Membership privacy; Indistinguishability; Min-max game; Adversarial process

1 INTRODUCTION

Large available datasets and powerful computing infrastructures, as well as advances in training complex machine learning models, have dramatically increased the adoption of machine learning in software systems. Many algorithms, applications, and services that used to be built based on expert knowledge, now can be designed using much less engineering effort by relying on advanced machine learning algorithms. Machine learning itself has also been provided as a service, to facilitate the use of this technology by system designers and application developers. Data holders can train models using machine learning as a service (MLaaS) platforms (by Google, Amazon, Microsoft, ...) and share them with others or use them in their own applications. The models are accessible through prediction APIs, which allow simple integration of machine learning algorithms into applications in the Internet.

A wide range of sensitive data, such as online and offline profiles of users, location traces, personal photos, speech samples, medical and clinical records, and financial portfolios, is used as input for training machine learning models. The confidentiality and privacy of such data is of utmost importance to data holders. Even if the training platform is trusted (e.g., using confidential computing, or by simply training the model on the data owner's servers) the remaining concern is if a model's computations (i.e., its predictions) can be exploited to endanger privacy of its sensitive training data.

The data required for training accurate models presents serious privacy issues. The leakage through complex machine learning models maybe less obvious, compared to, for example, linear statistics [18]. However, machine learning models, similar to other types of computations, could significantly leak information about the datasets on which they are computed. In particular, an adversary, with even black-box access to a model, can perform a membership inference [23, 43] (also known as the tracing [17]) attack against the model to determine whether or not a target data record is a member of its training set [45]. The adversary exploits the distinctive statistical features of the model's predictions on its training data. This is a fundamental threat to data privacy, and is shown to be effective against various machine learning models and services [45].

In this paper, we focus on protecting machine learning models against this exact threat: black-box membership inference attacks. There are two major groups of existing defense mechanisms. The first group includes simple mitigation techniques, such as limiting the model's predictions to top-k classes, therefore reducing the precision of predictions, or regularizing the model (e.g., using L2-norm regularizers) [19, 45]. These techniques may impose a negligible utility loss to the model. However, they cannot guarantee any rigorous notion of privacy. The second major group of defenses use differential privacy mechanisms [1, 6, 10, 40, 41]. These mechanisms do guarantee (membership) privacy up to their privacy parameter ϵ. However, the existing mechanisms may impose a significant classification accuracy loss for protecting large models on high dimensional data for small values of ϵ. This comes from not explicitly including utility into the design objective of the privacy mechanism. Also, it is because the differential privacy mechanisms are designed so as to guarantee input indistinguishability for *all* possible input training datasets (that differ in a constant number of records), and for *all* possible parameters/outputs of the models. Whereas, we explicitly include utility in the objective of the privacy mechanism which protects the very existing training dataset.

Contributions. In this paper, we design a rigorous privacy mechanism for protecting a given training dataset, against a particular adversarial objective. We want to train machine learning models that guarantee **membership privacy**: No adversary can distinguish between the predictions of the model on its training set from the model's prediction on other data samples from the same underlying distribution, up to the privacy parameter. This is a more targeted privacy notion than differential privacy, as we aim at a very specific (prediction) indistinguishability guarantee. Our objective is that the privacy-preserving model should achieve membership privacy with the minimum classification loss.

We formalize membership inference attacks and define the defender's objective for achieving membership privacy for classification models. Based on these definitions, we design an optimization problem to minimize the classification error of the model *and* the inference accuracy of the strongest attack who adaptively maximizes his gain. Therefore, this problem optimizes a composition of two conflicting objectives. We model this optimization as a **min-max privacy game** between the defense mechanism and the inference attack, similar to privacy games in other settings [2, 24, 27, 34, 46]. The solution is a model which not only is accurate but also has the maximum membership privacy against its corresponding strongest inference attack. The adversary cannot design a better inference attack than what is already anticipated by the defender; therefore, membership privacy is guaranteed. There does not also exist any model that, for the same level of membership privacy, can give a better accuracy. So, maximum utility (for the same level of privacy) is also guaranteed.

To find the solution to our optimization problem, we train the model in an **adversarial process**. The classification model maps features of a data record to classes, and computes the probability that it belongs to any class. The primary objective of this model is to minimize prediction error. The inference model maps a target data record, and the output of the classifier on it, to its membership probability. The objective of the inference model is to maximize its membership inference accuracy. To protect data privacy, we add the gain of the inference attack as a *regularizer* for the classifier. Using a regularization parameter, we can control the trade-off between membership privacy and classification error. We train the models in a similar way as generative adversarial networks [21] and other adversarial processes for machine learning [11, 14, 29, 35, 36, 38]. Our training algorithm can converge to an equilibrium point where the best membership inference attack against it is *random guess*, and this is achieved with minimum classification accuracy loss.

We present the experimental results on deep neural networks using benchmark ML datasets as well as the datasets used in the ML privacy literature. We compute various statistics of models' predictions on their training and test sets, in order to illustrate the worst case and the average case gaps between these statistics (which cause the privacy risk). The gaps are reduced by several orders of magnitude when the model is trained using our min-max privacy mechanism, compared to non-privacy-preserving models.

Our results verify our theoretical analysis that we impose only a **negligible loss in classification accuracy for a significant gain in membership privacy**. For the CIFAR100 dataset trained with Alexnet and Densenet architectures, the cost is respectively 1.1% and 3% drop in the prediction accuracy, relative to the regular non-privacy-preserving models. For the Purchase100 and Texas100 datasets (used in [45]), the cost of membership privacy in terms of classification accuracy drop is 3.6% and 4.4%, respectively, for reducing the inference accuracy from 67.6% to 51.6% and from 63% to 51%, respectively. Note that the membership privacy is maximum when the membership inference accuracy is 50% (random guess).

We also show that **our mechanism strongly regularizes the models**, by significantly closing the gap between their training and testing accuracy, and preventing overfitting. This directly follows from the indistinguishability of our privacy-preserving model's prediction distributions on training versus test data. For example, on the Purchase100 dataset, we can obtain 76.5% testing accuracy for 51.8% membership inference accuracy. In contrast, a standard L2-norm regularizer may provide a similar level of privacy (against the same attack) but with a 32.1% classification accuracy.

2 MACHINE LEARNING

In this paper, we focus on training classification models using supervised learning. Table 1 summarizes the notations and formally states the objective function of the classifier. Let X be the set of all possible data points in a d-dimensional space, where each dimension represents one attribute of a data point (and will be used as the input features in the classification model). We assume there is a predefined set of k classes for data points in X. The objective is to find the relation between each data point and the classes as a classification function $f : X \longrightarrow Y$. The output reflects how f classifies each input into different classes. Each element of an output $y \in Y$ is a score vector that shows the relative association of any input to different classes. All elements of a vector y are in range $[0, 1]$, and are normalized such that they sum up to 1, so they are interpreted as the probabilities that the input belongs to different classes.

Let $\Pr(\mathbf{X}, \mathbf{Y})$ represent the underlying probability distribution of all data points in the universe $X \times Y$, where $\mathbf{X}$ and $\mathbf{Y}$ are random variables for the features and the classes of data points, respectively. The objective of a machine learning algorithm is to find a classification model f that accurately represents this distribution and maps each point in X to its correct class in Y. We assume we have a lower-bounded real-valued loss function $l(f(x), y)$ that, for each data point (x, y), measures the difference between y and the model's prediction $f(x)$. The machine learning objective is to find a function f that minimizes the expected *loss*:

$$\mathrm{L}(f) = \underset{(x,y)\sim\Pr(\mathbf{X},\mathbf{Y})}{\mathbb{E}}[l(f(x), y)] \tag{1}$$

We can estimate the probability function $\Pr(\mathbf{X}, \mathbf{Y})$ using samples drawn from it. These samples construct the training set $D \subset X \times Y$. Instead of minimizing (1), machine learning algorithms minimize the expected *empirical loss* of the model over its training set D.

$$\mathrm{L}_D(f) = \frac{1}{|D|} \sum_{(x,y)\in D} l(f(x), y) \tag{2}$$

Classification model	$f : X \longrightarrow Y$
Loss	$\mathrm{L}(f) = \underset{(x,y)\sim \Pr(\mathbf{X},\mathbf{Y})}{\mathbb{E}}[l(f(x),y)] = \int\limits_{X\times Y} l(f(x),y)\Pr(\mathbf{X},\mathbf{Y})\,dx\,dy$
Empirical loss	$\mathrm{L}_D(f) = \frac{1}{\lvert D\rvert} \sum\limits_{(x,y)\in D} l(f(x),y)$
Optimization problem	$\min\limits_{f} \mathrm{L}_D(f) + \lambda\, R(f)$

Table 1: Definition, loss, and optimization problem for the *classification model* f, where $x \in X$ is a data point, $y \in Y$ is a classification vector, D is the model's training set, $l()$ is a loss function, $R()$ is a regularizer, and λ is the regularization factor.

Inference model	$h : X \times Y^2 \longrightarrow [0,1]$
Gain	$\mathrm{G}_f(h) = \underset{(x,y)\sim \Pr_D(\mathbf{X},\mathbf{Y})}{\mathbb{E}}[log(h(x,y,f(x)))] + \underset{(x,y)\sim \Pr_{\setminus D}(\mathbf{X},\mathbf{Y})}{\mathbb{E}}[\log(1-h(x,y,f(x)))]$
Empirical gain	$\mathrm{G}_{f,D^{\mathrm{A}},D'^{\mathrm{A}}}(h) = \frac{1}{2\lvert D^{\mathrm{A}}\rvert} \sum_{(x,y)\in D^{\mathrm{A}}} \log(h(x,y,f(x))) + \frac{1}{2\lvert D'^{\mathrm{A}}\rvert} \sum_{(x',y')\in D'^{\mathrm{A}}} \log(1-h(x',y',f(x')))$
Optimization problem	$\max\limits_{h} \mathrm{G}_{f,D^{\mathrm{A}},D'^{\mathrm{A}}}(h)$

Table 2: Definition, gain, and optimization problem for the *membership inference attack* h, where f is the target classifier, $\Pr_D(\mathrm{X},\mathrm{Y})$ and $\Pr_{\setminus D}(\mathrm{X},\mathrm{Y})$ are the conditional probability distributions of data points in the target training set D and outside it, respectively. The adversary's background knowledge is composed of datasets D^{A} (a subset of the training set D) and D'^{A} (samples drawn from $\Pr(\mathrm{X},\mathrm{Y})$ which are outside D). See Figure 1 for the illustration of the relation between h and f.

We can now state the optimization problem of learning a classification model as the following:

$$\min_{f} \mathrm{L}_D(f) + \lambda\, R(f) \tag{3}$$

where $R(f)$ is a *regularization* function.

The function $R(f)$ is designed to prevent the model from overfitting to its training dataset [7]. For example, the regularization loss (penalty) increases as the parameters of the function f grow arbitrarily large or co-adapt themselves to fit the particular dataset D while minimizing $\mathrm{L}_D(f)$. If a model overfits, it obtains a small loss on its training data, but fails to achieve a similar loss value on other data points. By avoiding overfitting, models can generalize better to all data samples drawn from $\Pr(\mathbf{X},\mathbf{Y})$. The regularization factor λ controls the balance between the classification loss function and the regularizer.

For solving the optimization problem (3), especially for non-convex loss functions for complex models such as deep neural networks, the commonly used method is the stochastic gradient descent algorithm [4, 52]. This is an iterative algorithm where in each epoch of training, it selects a small subset (mini-batch) of the training data and updates the model (parameters) towards reducing the loss over the mini-batch. After many epochs of training, the algorithm converges to a local minimum of the loss function.

3 MEMBERSHIP INFERENCE ATTACK

The objective of membership inference attacks, also referred to as tracing attacks, is to determine whether or not a target data record is in a dataset, assuming that the attacker can observe a function over the dataset (e.g., aggregate statistics, model).

The membership inference attacks have mostly been studied for analyzing data privacy with respect to simple statistical linear functions [5, 17, 18, 23, 42, 43]. The attacker compares the released statistics from the dataset, and the same statistics computed on random samples from the population, to see which one is closer to the target data record. Alternatively, the adversary can compare the target data record and samples from the population, to see which one is closer to the released statistics. In either case, if the target is closer to the released statistics, then there is a high chance that it was a member of the dataset. The problem could be formulated as a hypothesis test, and the adversary can make use of likelihood ratio test to run the inference attack.

In the case of machine learning models, the membership inference attack is not as simple, especially in the black-box setting. The adversary needs to distinguish training set members from non-members from observing the model's predictions, which are

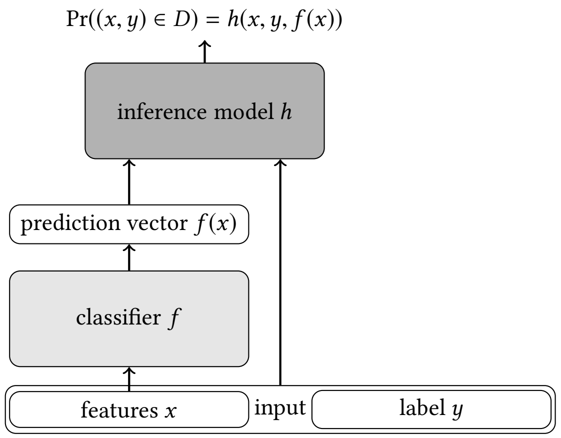

Figure 1: The relation between different elements of the black-box classification model f and the inference model h.

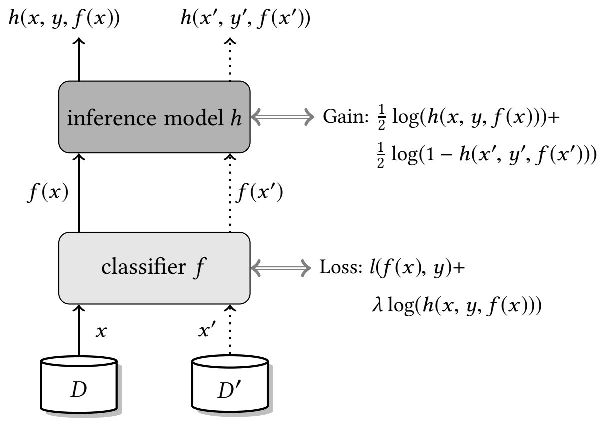

Figure 2: Classification loss and inference gain, on the training dataset D and reference dataset D', in our adversarial training. The classification loss is computed over D, but, the inference gain is computed on both sets. To simplicity the illustration, the mini-batch size is set to 1.

indirect nonlinear computations on the training data. The existing inference algorithm suggests training another machine learning model, as the inference model, to find the statistical differences between predictions on members and predictions on non-members [45]. In this section, we formally present this attack and the optimization problem to model the adversary's objective. Table 2 summarizes the notations and the optimization problem. Figure 1 illustrates the relation between different components of a membership inference attack against machine learning models in the black-box setting.

Let h be the inference model $h : X \times Y^2 \longrightarrow [0,1]$. For any data point (x,y) and the model's prediction vector $f(x)$, it outputs the probability of (x,y) being a member of D (the training set of f). Let $\Pr_D(\mathbf{X},\mathbf{Y})$ and $\Pr_{\backslash D}(\mathbf{X},\mathbf{Y})$ be the conditional probabilities of $(\mathbf{X},\mathbf{Y})$ for samples in D and outside D, respectively. In an ideal setting (of knowing these conditional probability distributions), the gain function for the membership inference attack can be computed as the following.

$$\begin{aligned} G_f(h) = {} & \frac{1}{2} \mathop{\mathbb{E}}_{(x,y)\sim \Pr_D(\mathbf{X},\mathbf{Y})} [\log(h(x,y,f(x)))] \\ & + \frac{1}{2} \mathop{\mathbb{E}}_{(x,y)\sim \Pr_{\backslash D}(\mathbf{X},\mathbf{Y})} [\log(1-h(x,y,f(x)))]) \end{aligned} \tag{4}$$

The two expectations compute the correctness of the inference model h when the target data record is sampled from the training set, or from the rest of the universe. In a realistic setting, the probability distribution of data points in the universe and the probability distribution over the members of the training set D are not directly and accurately available to the adversary (for computing his gain). Therefore, we compute the *empirical* gain of the inference model on two disjoint datasets D^{A} and D'^{A}, which are sampled according to the probability distribution of the data points inside the training set and outside it, respectively. More concretely, the dataset D^{A} could be a subset of the target training set D, known to the adversary. Given these sets, the empirical gain of the membership inference model is computed as the following.

$$\begin{aligned} G_{f,D^{\mathrm{A}},D'^{\mathrm{A}}}(h) = {} & \frac{1}{2|D^{\mathrm{A}}|} \sum_{(x,y)\in D^{\mathrm{A}}} \log(h(x,y,f(x))) \\ & + \frac{1}{2|D'^{\mathrm{A}}|} \sum_{(x',y')\in D'^{\mathrm{A}}} \log(1-h(x',y',f(x'))) \end{aligned} \tag{5}$$

Thus, the optimization problem for the membership inference attack is simply maximizing this empirical gain.

$$\max_h \; G_{f,D^{\mathrm{A}},D'^{\mathrm{A}}}(h) \tag{6}$$

The optimization problem needs to be solved on a given target classification model f. However, it is shown that it can also be trained on some shadow models, which have the same model type, architecture, and objective function as the model f, and are trained on data records sampled from $\Pr(\mathbf{X},\mathbf{Y})$ [45].

4 MIN-MAX MEMBERSHIP PRIVACY GAME

The adversary always has the upper hand. He adapts his inference attack to his target model in order to maximize his gain with respect to this *existing* classification model. This means that a defense mechanism will be eventually broken if it is designed with respect to a particular attack, without anticipating and preparing for the (strongest) attack against itself. The conflicting objectives of the defender and the adversary can be modeled as a privacy game [2, 24, 34, 46]. In our particular setting, while the adversary tries to get the maximum inference gain, the defender needs to find the classification model that not only minimizes its loss, but also minimizes the adversary's maximum gain. This is a min-max game.

The privacy objective of the classification model is to minimize its privacy loss with respect to the worst case (i.e., maximum inference gain) attack. It is easy to achieve this by simply making the

Algorithm 1 The adversarial training algorithm for machine learning with membership privacy. This algorithm optimizes the min-max objective function (7). Each epoch of training includes k steps of the maximization part of (7), to find the best inference attack model, followed by one step of the minimization part of (7) to find the best defensive classification model against such attack model.

```
for number of the training epochs do
    for k steps do
        Randomly sample a mini-batch of $m$ training data points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ from the training set $D$.
        Randomly sample a mini-batch of $m$ reference data points $\{(x'_1, y'_1), (x'_2, y'_2), \cdots, (x'_m, y'_m)\}$ from the reference set $D'$.
        Update the inference model $h$ by ascending its stochastic gradients over its parameters $\omega$:
            $\nabla_{\omega} \frac{\lambda}{2m} \left( \sum_{i=1}^{m} \log(h(x_i, y_i, f(x_i))) + \sum_{i=1}^{m} \left( \log(1 - h(x'_i, y'_i, f(x'_i))) \right) \right)$
    end for
    Randomly sample a fresh mini-batch of $m$ training data points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ from $D$.
    Update the classification model $f$ by descending its stochastic gradients over its parameters $\theta$:
        $\nabla_{\theta} \frac{1}{m} \sum_{i=1}^{m} \left( l(f(x_i), y_i) + \lambda \log(h(x_i, y_i, f(x_i))) \right)$
end for
```

output of the model independent of its input, at the cost of destroying the utility of the classifier. Thus, we update the training objective of the classification model as minimizing privacy loss with respect to the strongest inference attack, *with* minimum classification loss. This results in designing the optimal privacy mechanism which is also utility maximizing.

We formalize the joint privacy and classification objectives in the following min-max optimization problem.

$$\underbrace{\min_{f} \left(\mathrm{L}_D(f) + \lambda \underbrace{\max_{h} \mathrm{G}_{f,D,D'}(h)}_{\text{optimal inference}} \right)}_{\text{optimal privacy-preserving classification}} \tag{7}$$

The inner maximization finds the strongest inference model h against a given classification model f. The outer minimization finds the strongest defensive classification model f against a given h. The parameter λ controls the importance of optimizing classification accuracy versus membership privacy. The inference attack term which is multiplied by λ acts as a regularizer for the classification model. In other words, it prevents the classification model to arbitrarily adapt itself to its training data at the cost of leaking information about the training data to the inference attack model. Note that (7) is equivalent to (3), if we set $R(f)$ to (6).

These two optimizations need to be solved jointly to find the equilibrium point. For arbitrarily complex functions f and h, this game can be solved numerically using the stochastic gradient descent algorithm (similar to the case of generative adversarial networks [21]). The training involves two datasets: the training set D, which will be used to train the classifier, and a disjoint reference set D' that, similar to the training set, contains samples from $\Pr(\mathbf{X}, \mathbf{Y})$.

Algorithm 1 presents the pseudo-code of the adversarial training of the classifier f on D—against its best inference attack model h. In each epoch of training, the two models f and h are alternatively trained to find their best responses against each other through solving the nested optimizations in (7). In the inner optimization step: for a fixed classifier f, the inference model is trained to distinguish the predictions of f on its training set D from predictions of the same model f on reference set D'. This step maximizes the empirical inference gain $\mathrm{G}_{f,D,D'}(h)$. In the outer optimization step: for a fixed inference attack h, the classifier is trained on D, with the adversary's gain function acting as a regularizer. This minimizes the empirical classification loss $\mathrm{L}_D(f) + \lambda\, \mathrm{G}_{f,D,D'}(h)$. We want this algorithm to converge to the equilibrium point of the min-max game that solves (7).

Theoretical Analysis. Our ultimate objective is to train a classification model f such that it has indistinguishably similar output distributions for data members of its training set versus the non-members. We make use of the theoretical analysis of the generative adversarial networks [21] to reason about how Algorithm 1 tries to converge to such privacy-preserving model. For a given classification model f, let p_f be the probability distribution of its output (i.e., prediction vector) on its training data D, and let p'_f be the probability distribution of the output of f on any data points outside the training dataset (i.e., $X \times Y \setminus D$).

For a given classifier f, the *optimal attack* model maximizes (4), which can be expanded to the following.

$$\mathrm{G}_f(h) = \frac{1}{2}\Big(\int_{x,y} \Pr_D(x,y)\, p_f(f(x)) \log(h(x,y,f(x)))dxdy + \int_{x',y'} \Pr_{\setminus D}(x',y')\, p'_f(f(x')) \log(1 - h(x',y',f(x')))dx'dy'\Big) \tag{8}$$

The maximum value of $\mathrm{G}_f(h)$ is achievable by the optimal inference model h^*_f with enough learning capacity, and is equal to

$$h^*_f(x,y,f(x)) = \frac{\Pr_D(x,y)\, p_f(f(x))}{\Pr_D(x,y)\, p_f(f(x)) + \Pr_{\setminus D}(x,y)\, p'_f(f(x))} \tag{9}$$

This combines what is already known (to the adversary) about the distribution of data inside and outside the training set, and what can be learned from the predictions of the model about its training set. Given that the training set is sampled from the underlying probability distribution $\Pr(\mathbf{X}, \mathbf{Y})$, and assuming that the underlying distribution of the training data is a-priori unknown to the adversary, the optimal inference model is the following.

$$h^*_f(x, y, f(x)) = \frac{p_f(f(x))}{p_f(f(x)) + p'_f(f(x))} \tag{10}$$

This means that the best strategy of the adversary is to determine membership by comparing the probability that the prediction $f(x)$ comes from distribution p_f or alternatively from p'_f.

Given the optimal strategy of adversary against any classifier, we design the *optimal classifier* as the best response to the inference attack. The privacy-preserving classification task has two objectives (7): minimizing both the classification loss $\mathrm{L}_D(f)$ and the privacy loss $\mathrm{G}_{f,D,D'}(h^*_f)$. In the state space of all classification models f that have the same classification loss $\mathrm{L}_D(f)$, the min-max game (7) will be reduced to solving $\min_f \max_h \mathrm{G}_{f,D,D'}(h)$ which is then computed as:

$$\begin{aligned}\min_f \max_h \; & \mathop{\mathbb{E}}_{(x,y)\sim \Pr_D(\mathbf{X},\mathbf{Y})} [\log(h(x, y, f(x)))] \\ & + \mathop{\mathbb{E}}_{(x,y)\sim \Pr_{\setminus D}(\mathbf{X},\mathbf{Y})} [\log(1 - h(x, y, f(x)))] \end{aligned} \tag{11}$$

According to Theorem 1 in [21], the optimal function f^* is the global minimization function if and only if $p_{f^*} = p'_{f^*}$. This means that for a fixed classification loss and with enough learning capacity for model f, the training algorithm minimizes the privacy loss by making the two distributions p_{f^*} and p'_{f^*} indistinguishable. This implies that the optimal classifier pushes the membership inference probability $h^*_f(x, y, f(x))$ to converge to 0.5, i.e., random guess. According to Proposition 2 in [21], we can prove that the stochastic gradient descent algorithm of Algorithm 1 eventually converges to the equilibrium of the min-max game (7). To summarize, the solution will be a classification model with minimum classification loss such that the strongest inference attack against it cannot distinguish its training set members from non-members by observing the model's predictions on them.

5 EXPERIMENTS

In this section, we apply our method to several different classification tasks using various neural network structures. We implemented our method using Pytorch[1]. The purpose of this section is to empirically show the robustness of our privacy-preserving model against inference attacks and its negligible classification loss.

5.1 Datasets

We use three datasets: a major machine learning benchmark dataset (CIFAR100), and two datasets (Purchase100, Texas100) which are used in the original membership inference attack against machine learning models [45].

CIFAR100. This is a major benchmark dataset used to evaluate image recognition algorithms [30]. The dataset contains 60,000 images, each composed of 32 × 32 color pixels. The records are clustered into 100 classes, where each class represents one object.

Purchase100. This dataset is based on Kaggle's "acquire valued shopper" challenge. [2] The dataset includes shopping records for several thousand individuals. The goal of the challenge is to find offer discounts to attract new shoppers to buy new products. Courtesy of the authors [45], we obtained the processed and simplified version of this dataset. Each data record corresponds to one costumer and has 600 binary features (each corresponding to one item). Each feature reflects if the item is purchased by the costumer or not. The data is clustered into 100 classes and the task is to predict the class for each costumer. The dataset contains 197,324 data records.

Texas100. This dataset includes hospital discharge data. The records in the dataset contain information about inpatient stays in several health facilities published by the Texas Department of State Health Services. Data records have features about the external causes of injury (e.g., suicide, drug misuse), the diagnosis (e.g., schizophrenia, illegal abortion), the procedures the patient underwent (e.g., surgery), and generic information such as gender, age, race, hospital ID, and length of stay. Courtesy of the authors [45], we obtained the processed dataset, which contains 67,330 records and 6,170 binary features which represent the 100 most frequent medical procedures. The records are clustered into 100 classes, each representing a different type of patient.

5.2 Classification Models

For the CIFAR100 dataset, we used two different neural network architectures. (1) Alexnet architecture [31], trained with Adam optimizer[28] with learning rate 0.0001, and 100 epochs of training. (2) DenseNet architecture [25], trained with stochastic gradient descent (SGD) for 300 epochs, with learning rate 0.001 from epoch 0 to 100, 0.0001 from 100 to 200, and 0.00001 from 200 to 300. Following their architectures, both these models are regularized. Alexnet uses Dropout (0.2), and Densenet uses L2-norm regularization (5e-4).

For the Purchase100 dataset, we used a 4-layer fully connected neural network with layer sizes [1024, 512, 256, 100]. We used Tanh activation functions, similar to [45]. We initialized all of parameters with a random normal distribution with mean 0 and standard deviation 0.01. We trained the model for 50 epochs.

For the Texas dataset, we used a 5-layer fully connected neural network with layer sizes [2048, 1024, 512, 256, 100], with Tanh activation functions. We initialized all of parameters with a random normal distribution with mean 0 and standard deviation 0.01. We trained the model for 50 epochs.

Table 3 shows the number of training data as well as reference data samples which we used in our experiments for different datasets. It also reports the adversarial regularization factor λ which was used in our experiments.

[1] http://pytorch.org/

[2] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

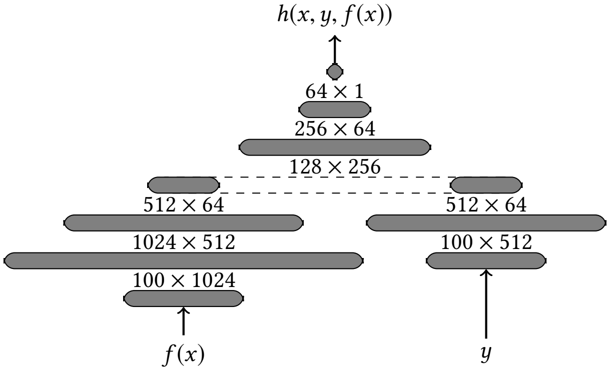

Figure 3: The neural network architecture for the inference attack model. Each layer is fully connected to its subsequent layer. The size of each fully connected layer is provided.

5.3 Inference Attack Model

For the inference model, we also make use of neural networks. Figure 3 illustrates the architecture of our inference neural network. The objective of the attack model is to compute the membership probability of a target record (x, y) in the training set of the classification model f. The attack model inputs (x, y) as well as the prediction vector of the classification model on it, i.e., $f(x)$. We design the inference attack model with three separate fully connected sub-networks. One network of layer sizes [100,1024,512,64] operates on the prediction vector $f(x)$. One network of layer sizes [100,512,64] operates on the label which is one-hot coded (all elements are 0 except the one that corresponds to the label index). The third (common) network operates on the concatenation of the output of the first two networks and has layer sizes of [256,64,1]. In contrast to [45] which trains k membership inference attack models (one per class), we design only a single model for the inference attack. The architecture of our attack model, notably its last (common) layers, enables capturing the relation between the class and the predictions of the model for training set members versus non-members.

We use ReLu as the activation function in the network. All weights are initialized with normal distribution with mean 0 and standard deviation 0.01, and all biases are initialized to 0. We use Adam optimizer with learning rate 0.001. We make sure every training batch for the attack model has the same number of member and non-member instances to prevent the attack model to be biased toward one side.

Table 3 shows the number of known members of the training set, D^{A}, and known non-member data points, D'^{A}, that we assume for the adversary, which is used for training his attack model. The larger these sets are (especially the D^{A} set), the more knowledge is assumed for the attacker. As we are evaluating our defense mechanism, we assume a strong adversary who knows a substantial fraction of the training set and tries to infer the membership of the rest of it.

Dataset	$\lvert D\rvert$	$\lvert D'\rvert$	λ	$\lvert D^{\mathrm{A}}\rvert$	$\lvert D'^{\mathrm{A}}\rvert$
Purchase100	20,000	20,000	3	5,000	20,000
Texas100	10,000	5,000	2	5,000	10,000
CIFAR100	50,000	5,000	6	25,000	5,000

Table 3: Experimental setup, including the size of the training set D and reference set D' in Algorithm 1, the adversarial regularization factor λ, as well as the size of the adversary's known members D^{A} of the classifier's training set and known non-members D'^{A}.

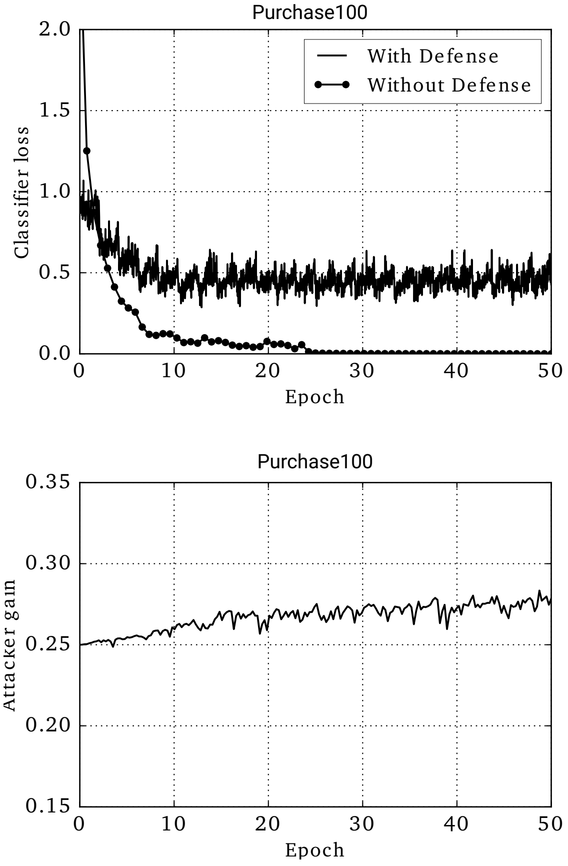

Figure 4: The trajectory of the classification loss during training with/without defense mechanism, as well as the inference attack gain, using the Purchase100 dataset.

5.4 Empirical Results

Loss and Gain of the Adversarial Training

Figure 4 shows the empirical loss of the classification model (2) as well as the empirical gain of the inference model (5) throughout the training using Algorithm (1). By observing both the classifier's loss and the attack's gain over the training epochs, we can see that they converge to an equilibrium point. Following the optimization problem (7), the attacker's gain is the maximum that can be achieved against the best defense mechanism. The classifier's

	Without defense			With defense		
Dataset	Training accuracy	Testing accuracy	Attack accuracy	Training accuracy	Testing accuracy	Attack accuracy
Purchase100	100%	80.1%	67.6%	92.2%	76.5%	51.6%
Texas100	81.6%	51.9%	63%	55%	47.5%	51.0%
CIFAR100- Alexnet	99%	44.7%	53.2%	66.3%	43.6%	50.7%
CIFAR100- DenseNET	100%	70.6%	54.5%	80.3%	67.6%	51.0%

Table 4: Comparison of membership privacy and training/test accuracy of a classification model (without defense), and a privacy-preserving model (with defense) on four different models/datasets. Compare the two cases with respect to the trade-off between testing accuracy and attack accuracy. See Table 3 for the experimental setup.

λ	Training accuracy	Testing accuracy	Attack accuracy
0 (no defense)	100%	80.1%	67.6%
1	98.7%	78.3%	57.0%
2	96.7%	77.4%	55.0%
3	92.2%	76.5%	51.8%
10	76.3%	70.1%	50.6%

Table 5: The effect of the adversarial regularization factor λ, used in the min-max optimization (7), which also acts as our privacy parameter, on the defense mechanism trained on the Purchase100 dataset.

L2-regularization factor	Training accuracy	Testing accuracy	Attack accuracy
0 (no regularization)	100%	80.1%	67.6%
0.001	86%	81.3%	60%
0.005	74%	70.2%	56%
0.01	34%	32.1%	50.6%

Table 6: The results of using a $L2-$regularization as a mitigation technique for membership inference attack. The model is trained on the Purchase100 dataset. Compare these results with those in Table 4 which shows what we can achieve using the strategic min-max optimization.

Reference set size	Testing accuracy	Attack accuracy
1,000	80.0%	59.2%
5,000	77.4%	52.8%
10,000	76.8%	52.4%
20,000	76.5%	51.6%
30,000	76.4%	50.6%

Table 7: The effect of the size of the reference set D' on the defense mechanism for the Purchase100 dataset. Note that (as also shown in Table 3) the size of the training set is 20,000.

loss is also the minimum that can be achieved while preserving privacy against the best attack mechanism. As shown in Section 4, by minimizing the classifier's loss, we train a model that not only prevents the attack's gain to grow large, but also forces the adversary to play the random guess strategy.

In Figure 4 (top), we compare the evolution of the classification loss of the privacy-preserving model with the loss of the same model when trained regularly (without defense). As we can see, a regular model (without defense) arbitrarily reduces its loss, thus might overfit to its training data. Later in this section, in Figure 6, we visualize the impact of this small loss on the output of the model, and we show how this can leak information about the model's training set. ***The adversarial training for membership privacy strongly regularizes the model***. Thus, our privacy-preserving mechanism not only protects membership privacy but also significantly prevents overfitting.

Privacy and Generalization

To further study the tradeoff between privacy and predictive power of privacy-preserving models, in Figure 5 we show the cumulative distribution of the model's generalization error over different classes. The plot shows the fraction of classes (y-axis) for which the model has a generalization error under a certain value (x-axis). For each class, we compute the model's generalization error as the difference between the testing and training accuracy of the model for samples from that class [22]. We compare the generalization error of a regular model and our privacy-preserving model. As the plots show, the generalization error of our privacy mechanism is significantly lower over all the classes.

Table 4 presents all the results of training privacy-preserving machine learning models using our min-max game, for all our datasets. It also compares them with the same models when trained regularly (without defense). Note the gap between training and testing accuracy with and without the defensive training. Our mechanism reduces the total generalization error by a factor of up to 4. For example, the error is reduced from 29.7% down to 7.5% for the Texas100 model, it is reduced from 54.3% down to 22.7% for the CIFAR100-Alexnet model, and it is reduced from 29.4% down to 12.7% for the CIFAR100-Densenet model, while it remains almost the same for the Purchase100 model. ***Our min-max mechanism achieves membership privacy with the minimum generalization error.*** Table 5 shows how we can control the trade-off between prediction accuracy and privacy, by adjusting the adversarial regularization factor λ.

The regularization effect of our mechanism can be compared to what can be achieved using common regularizers such as the L2-norm regularizer, where $R(f_\theta) = \sum_i \theta_i^2$ (see our formalization of a

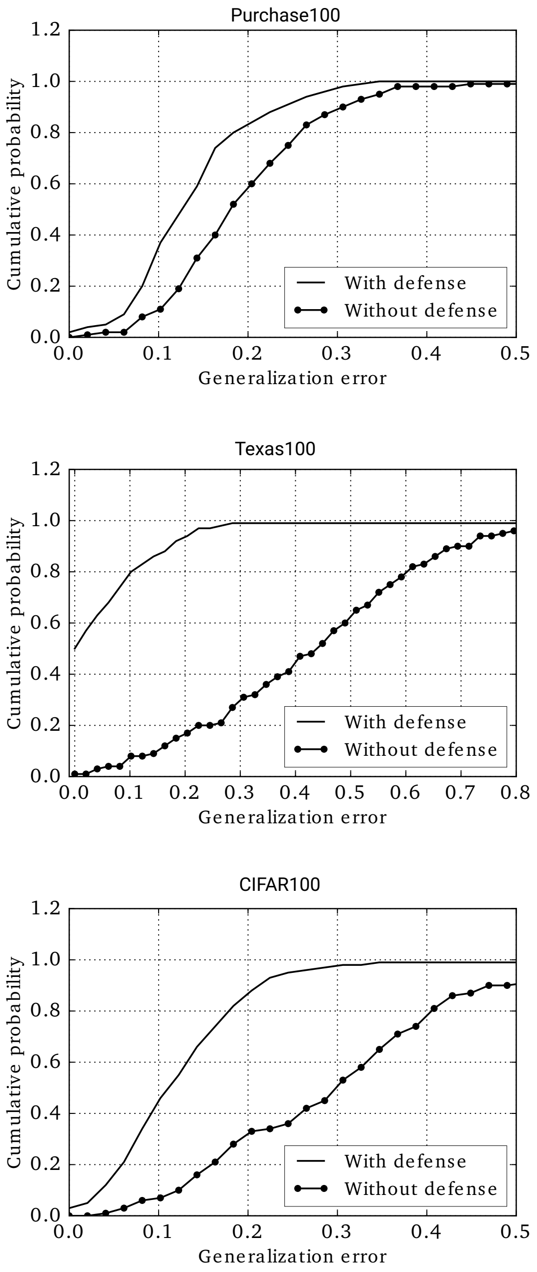

Figure 5: The empirical CDF of the generalization error of classification models across different classes, for regular models (without defense) versus privacy-preserving models (with defense). We compute generalization error as the difference between the training and testing accuracy of the model [22]. The y-axis is the fraction of classes that have generalization error less than x-axis. The curves that lean towards left have a smaller generalization error.

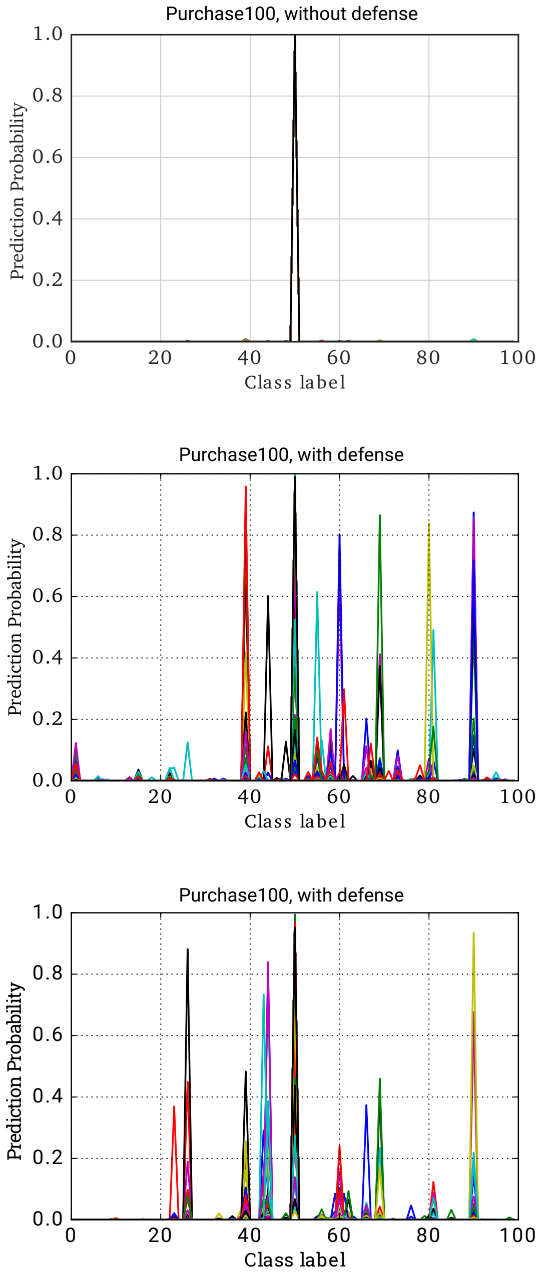

Figure 6: The distribution of the output (prediction vector) of the classifier on the training data samples from class 50 in the Purchase100 dataset. Each color represents one data sample. Without the defense, all samples are classified into class 50 with a probability close to 1. Whereas, the privacy-preserving classifier spreads the prediction probability across many classes. This added uncertainty is what provably mitigates the information leakage. The figure at the bottom is computed on the test data samples from class 50, which is indistinguishable from the middle figure.

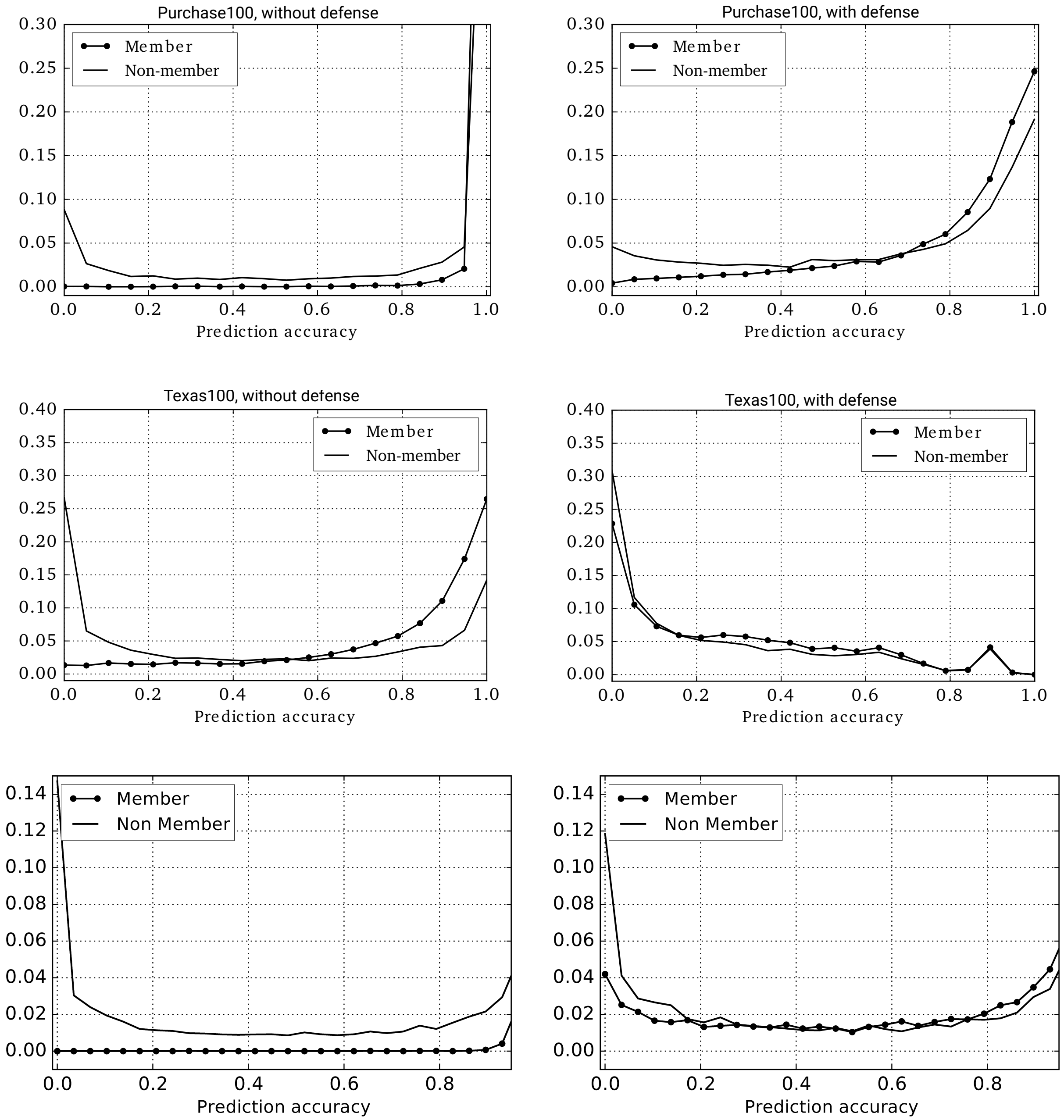

Figure 7: Distribution of the classifier's prediction accuracy on members of its training set versus non-member data samples. Accuracy is measured as the probability of predicting the right class for a sample input. The plots on the left show the distribution curves for regular models (without defense), and the ones on the right show the distribution curves for privacy-preserving models (with defense). The larger the gap between the curves in a plot is, the more the information leakage of the model about its training set is. The privacy-preserving model reduces this gap by one to two orders of magnitude.

– The *maximum* gap between the curves (with defense versus without defense) is as follows.
Purchase100 model: (0.02 **vs.** 0.34**), Texas100 model: (**0.05 **vs.** 0.25**), and CIFAR100-Densenet model: (**0.06 **vs.** 0.56**).**

– The *average* gap between the curves is as follows.
Purchase100 model: (0.007 **vs.** 0.013**), Texas100 model: (**0.004 **vs.** 0.016**), and CIFAR100-Densenet model: (**0.005 **vs.** 0.021**).**

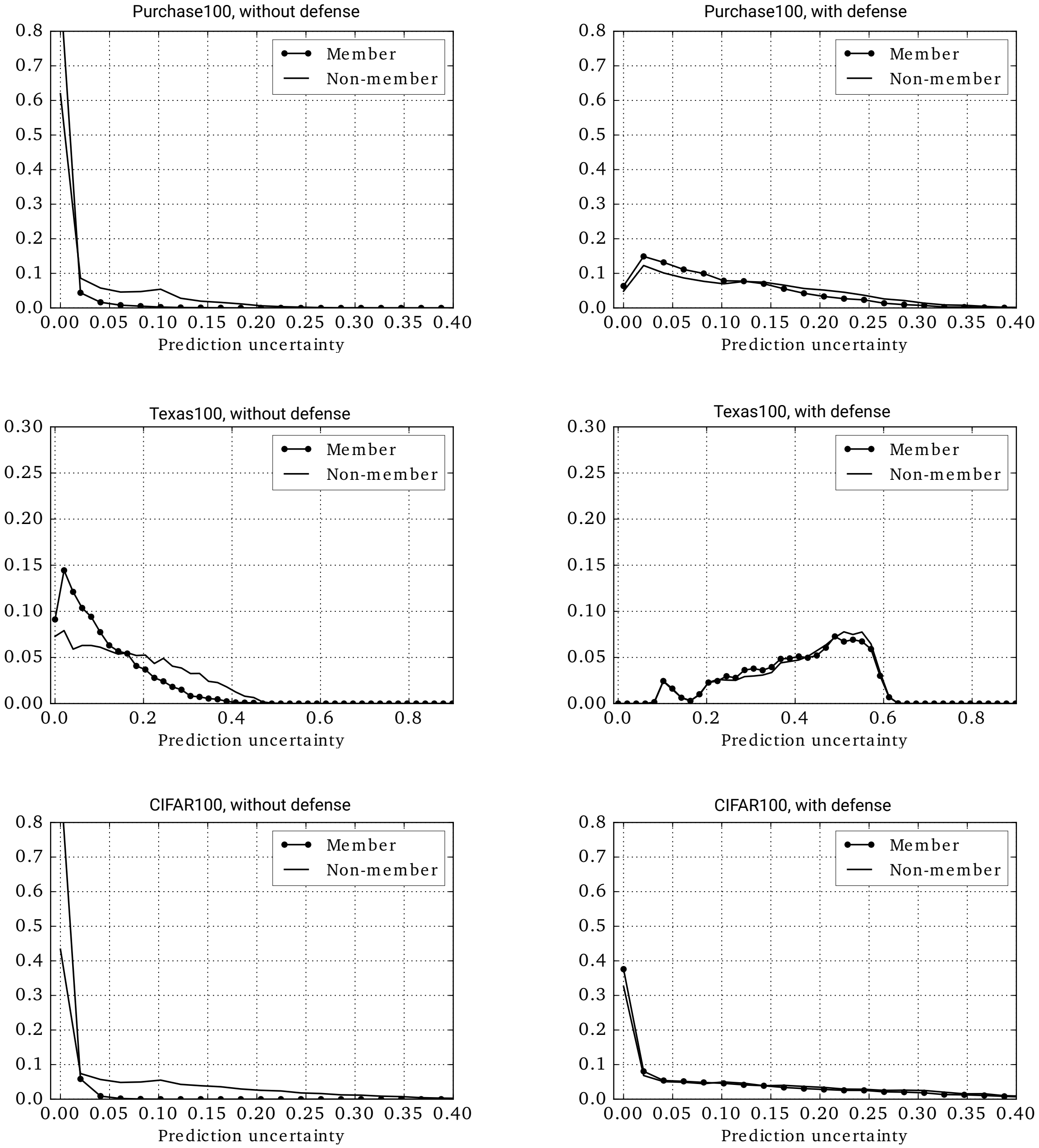

Figure 8: Distribution of the classifier's prediction uncertainty on members of its training set versus non-member data points. Uncertainty is measured as normalized Entropy of the model's output (i.e., prediction vector). The plots on the left show the distribution curves for regular models (without defense), and the ones on the right show the distribution curves for privacy-preserving models (with defense). The larger the gap between the curves in a plot is, the more the information leakage of the model about its training set is. The privacy-preserving model reduces this gap by one to two orders of magnitude.

– The *maximum* gap between the curves (with defense versus without defense) is as follows.
Purchase100 model: (0.03 vs. 0.30), Texas100 model: (0.02 vs. 0.15), and CIFAR100-Densenet model: (0.04 vs. 0.49).

– The *average* gap between the curves is as follows.
Purchase100 model: (0.004 vs. 0.012), Texas100 model: (0.002 vs. 0.04), and CIFAR100-Densenet model: (0.002 vs. 0.01).

classification loss function (3)). Table 6 shows the tradeoff between the model's test accuracy and membership privacy using L2-norm. Such regularizers do not guarantee privacy nor they minimize the cost of achieving it. For a close-to-maximum degree of membership privacy, the testing accuracy of our privacy-preserving mechanism is *more than twice* the testing accuracy of a L2-norm regularized model. This is exactly what we would expect from the optimization objectives of our privacy-preserving model.

Membership Privacy and Inference Attack Accuracy

Table 4 presents the training and testing accuracy of the model, as well as the attack accuracy. To measure the *attack accuracy*, we evaluate the average probability that the inference attack model correctly predicts the membership:

$$\frac{\sum_{(x,y)\in D\setminus D^{\mathsf{A}}} h(x,y,f(x)) + \sum_{(x'',y'')\in D''} (1 - h(x'',y'',f(x'')))}{|D \setminus D^{\mathsf{A}}| + |D''|}$$

where D'' is a set of data points that are sampled from the same underlying distribution as the training set, but does not overlap with D nor with D'^{A}.

The most important set of results in Table 4 is the two pairs of colored columns which represent the testing accuracy of the classifier versus the attack accuracy. There is a tradeoff between the predictive power of the model and its robustness to membership inference attack. As expected from our theoretical results, the experimental results show that the attack accuracy is much smaller (and close to random guess) in the privacy-preserving model compared to a regular model. ***Our privacy-preserving mechanism can guarantee maximum achievable membership privacy with only a negligible drop in the model's predictive power***. To achieve a near maximum membership privacy, the testing accuracy is dropped by 3.5% for the Purchase100 model, it is dropped by 4.4% for the Texas100 model, it is dropped by 1.1% for the CIFAR100-Alexnet model, and it is dropped by 3% for the CIFAR100-Densenet model.

Effect of the Reference Set

The objective of our min-max optimization is to make the predictions of the model on its training data indistinguishable from the model's predictions on any sample from the underlying data distribution. We make use of a set of samples from this distribution, named reference set, to empirically optimize the min-max objective. Table 7 shows the effect of the size of the reference set D' on the model's membership privacy. The models are trained on the same training set D of size 20,000, and hyper-parameter $\lambda = 3$. As expected, as the size of the reference set increases, it becomes better at properly representing the underlying distribution, thus the attack accuracy converges to 50%.

Indistinguishability of Predictions

The membership inference attacks against black-box models exploit the statistical differences between the predictions of the model on its members versus non-members. Figure 6 shows the output of the model (i.e., the probability of being a sample from each class) on its training data, for a regular model (without defense) versus a privacy-preserving model. The input data are all from class 50 in the Purchase100 dataset. The top figure illustrates that a regular model (which is overfitted on its training set) produces a high probability for the correct class on its training data. This significantly contributes to the vulnerability of the model to the membership inference attack. The privacy-preserving model produces a visibly different distribution (the middle figure). This makes the members' outputs indistinguishable from non-members' outputs (the bottom figure). The min-max optimization makes these two output distributions converge to indistinguishable distributions.

We further investigate the indistinguishability of these two distributions by computing some statistics (accuracy and uncertainty) of the model's output for different datasets. Figure 7 and Figure 8 show the results as the histogram of the models' accuracy and uncertainty over the training set and testing set. We compute the accuracy of model f on data point (x, y) as $f_y(x)$, which is the probability of predicting class y for input x. We compute uncertainty as the normalized entropy $\frac{-1}{\log(k)} \sum_i \hat{y}_i \log(\hat{y}_i)$ of the probability vector $\hat{y} = f(x)$, where k is the number of classes. The two figures show that ***our privacy mechanism significantly reduces both the maximum (worst case risk) and average gap between the prediction accuracy (and uncertainty) of the model on its training versus test set***, compared with a regular model. Note that these figures do not prove privacy, but illustrate what the attacker can exploit in his inference attacks. They visibly show how the indistinguishability of the model's output distributions (on members and non-members) can improve by using our defense mechanism.

6 RELATED WORK

Analyzing and protecting privacy in machine learning models against different types of attacks is a topic of ongoing research. A direct privacy threat against machine learning is the untrusted access of the machine learning platform during training or prediction. A number of defense mechanisms, which are based on trusted hardware and cryptographic private computing, have been proposed to enable blind training and use of machine learning models. These methods leverage various techniques including homomorphic encryption, garbled circuits, and secure multi-party computation for private machine learning on encrypted data [8, 20, 33, 37], as well as private computation using trusted hardware (e.g., Intel SGX) [26, 39]. Although these techniques prevent an attacker from directly observing the sensitive data, yet they do not limit information leakage through the computation itself.

An adversary with some background knowledge and external data can try to infer information such as the training data, the input query, and the parameters of the model. These inference attacks include input inference [19], membership inference [45], attribute inference [9], parameter inference [47, 48], and side-channel attacks [50]. There are examples of a wide-range of privacy attacks against computations over sensitive data. Our focus is on the privacy risks of computation on databases, when the adversary observes the result of the computation. In such settings, membership inference attacks and reconstruction attacks are considered as the two major classes of attacks [17].

Membership inference attack is a decisional problem. It aims at inferring the presence of a target data record in the (training) dataset [5, 18, 23, 42, 43, 45]. The accuracy of the attack shows the extent to which a model is dependent on its individual training data. The reconstruction attack is a more generic type of attack, where the objective is to infer sensitive attributes of many individuals in the training set [13, 49]. One proposed defense technique against general inference attacks is computation (e.g., training of models) with differential privacy guarantee [15, 16], which has recently been used in the context of machine learning [1, 6, 10, 40, 41]. Despite their provable robustness against inference attacks, differential privacy mechanisms are hard to achieve with negligible utility loss. The utility cost comes from the fact that we aim at protecting privacy against all strong attacks by creating indistinguishability among similar states of all possible input datasets. It is also related to the difficulty of computing a tight bound for the sensitivity of functions, which determines the magnitude of required noise for differential privacy. The relation between some different definitions of membership privacy and differential privacy is analyzed in the literature [32, 51].

Using game theory to formalize and optimize data privacy (and security) is another direction for protecting privacy [2, 24, 34, 44, 46]. In such a framework, the privacy loss is minimized against the strongest corresponding attack. The solution will be provably robust to any attack that threatens privacy according to such "loss" function. The game-theoretic framework allows to explicitly incorporate the utility function into the min-max optimization, thus also minimizing the cost of the privacy defense mechanism. The recent advances in machine learning, notably the developments of generative adversarial networks [3, 12, 21], have introduced new algorithms for solving min-max games while training a complex (deep neural network) model. Adversarial training has also been used for regularizing, hence generalizing, a model [11, 14, 29, 35, 36, 38].

7 CONCLUSIONS

We have introduced a new privacy mechanism for mitigating the information leakage of the predictions of machine learning models about the membership of the data records in their training sets. We design an optimization problem whose objective is to jointly maximize privacy and prediction accuracy. We design a training algorithm to solve a min-max game optimization that minimizes the classification loss of the model while maximizing the gain of the membership inference attack. The solution will be a model whose predictions on its training data are indistinguishable from its predictions on any data sample from the same underlying distribution. This mechanism guarantees membership privacy of the model's training set against the—strongest—inference attack, and imposes the minimum accuracy loss for achieving such level of privacy, given the available training/reference data and the capacity of the models. In our extensive experiments on applying our method on benchmark machine learning tasks, we show that the cost of achieving privacy is negligible, and that our privacy-preserving models can generalize well.

ACKNOWLEDGEMENTS

The authors would like to thank George Theodorakopoulos for helpful feedback.

REFERENCES

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

[2] Mário S Alvim, Konstantinos Chatzikokolakis, Yusuke Kawamoto, and Catuscia Palamidessi. 2017. Information leakage games. In *International Conference on Decision and Game Theory for Security*. Springer.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[4] Mordecai Avriel. 2003. *Nonlinear programming: analysis and methods*. Courier Corporation.

[5] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. 2016. Membership privacy in MicroRNA-based studies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

[6] Raef Bassily, Adam Smith, and Abhradeep Thakurta. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE.

[7] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

[8] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM.

[9] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. 2018. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. *arXiv preprint arXiv:1802.08232* (2018).

[10] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. 2011. Differentially private empirical risk minimization. *Journal of Machine Learning Research* (2011).

[11] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. 2017. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*.

[12] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. 2017. Training GANs with Optimism. *arXiv preprint arXiv:1711.00141* (2017).

[13] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM.

[14] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. 2016. Adversarially learned inference. *arXiv preprint arXiv:1606.00704* (2016).

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*. Springer.

[16] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* (2014).

[17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. 2017. Exposed! a survey of attacks on private data. (2017).

[18] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. 2015. Robust traceability from trace amounts. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE.

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

[20] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*.

[22] Moritz Hardt, Benjamin Recht, and Yoram Singer. 2015. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240* (2015).

[23] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics* (2008).

[24] Justin Hsu, Aaron Roth, and Jonathan Ullman. 2013. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM.

[25] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[26] Tyler Hunt, Congzheng Song, Reza Shokri, Vitaly Shmatikov, and Emmett Witchel. 2018. Chiron: Privacy-preserving Machine Learning as a Service. *arXiv preprint arXiv:1803.05961* (2018).

[27] Jinyuan Jia and Neil Zhenqiang Gong. 2018. AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning. *arXiv preprint arXiv:1805.04810* (2018).

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Mateusz Koziński, Loïc Simon, and Frédéric Jurie. 2017. An Adversarial Regularisation for Semi-Supervised Training of Structured Output Neural Networks. *arXiv preprint arXiv:1702.02382* (2017).

[30] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. (2009).

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

[32] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. 2013. Membership privacy: a unifying framework for privacy definitions. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM.

[33] Yehuda Lindell and Benny Pinkas. 2000. Privacy preserving data mining. In *Annual International Cryptology Conference*. Springer.

[34] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. 2013. Game theory meets network security and privacy. *ACM Computing Surveys (CSUR)* (2013).

[35] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2017. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976* (2017).

[36] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2015. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677* (2015).

[37] Payman Mohassel and Yupeng Zhang. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE.

[38] Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583* (2016).

[39] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. 2016. Oblivious Multi-Party Machine Learning on Trusted Processors.. In *USENIX Security Symposium*.

[40] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755* (2016).

[41] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. *arXiv preprint arXiv:1802.08908* (2018).

[42] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. *arXiv preprint arXiv:1708.06145* (2017).

[43] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. 2009. Genomic privacy and limits of individual detection in a pool. *Nature genetics* (2009).

[44] Reza Shokri. 2015. Privacy games: Optimal user-centric data obfuscation. *Proceedings on Privacy Enhancing Technologies* (2015).

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

[46] Reza Shokri, George Theodorakopoulos, Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2012. Protecting location privacy: optimal strategy against localization attacks. In *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM.

[47] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *USENIX Security*.

[48] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing Hyperparameters in Machine Learning. *arXiv preprint arXiv:1802.05351* (2018).

[49] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. 2009. Learning your identity and disease from research papers: information leaks in genome wide association study. In *Proceedings of the 16th ACM conference on Computer and communications security*. ACM.

[50] Lingxiao Wei, Yannan Liu, Bo Luo, Yu Li, and Qiang Xu. 2018. I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators. *arXiv preprint arXiv:1803.05847* (2018).

[51] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *arXiv preprint arXiv:1709.01604* (2018).

[52] Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*. ACM.