A Pragmatic Approach to Membership Inferences on Machine Learning Models

Yunhui Long[1], Lei Wang[2], Diyue Bu[2], Vincent Bindschaedler[3],
Xiaofeng Wang[2], Haixu Tang[2], Carl A. Gunter[1], and Kai Chen[4,5]

[1]University of Illinois at Urbana-Champaign
[2]Indiana University Bloomington
[3]University of Florida
[4]SKLOIS, Institute of Information Engineering, Chinese Academy of Sciences
[5]School of Cyber Security, University of Chinese Academy of Sciences

***Abstract*—Membership Inference Attacks (MIAs) aim to determine the presence of a record in a machine learning model's training data by querying the model. Recent work has demonstrated the effectiveness of MIA on various machine learning models and corresponding defenses have been proposed. However, both attacks and defenses have focused on an adversary that indiscriminately attacks all the records without regard to the cost of false positives or negatives. In this work, we revisit membership inference attacks from the perspective of a pragmatic adversary who carefully selects targets and make predictions conservatively. We design a new evaluation methodology that allows us to evaluate the membership privacy risk at the level of individuals and not only in aggregate. We experimentally demonstrate that highly vulnerable records exist even when the aggregate attack precision is close to 50% (baseline). Specifically, on the MNIST dataset, our pragmatic adversary achieves a precision of 95.05% whereas the prior attack only achieves a precision of 51.7%.**

1. Introduction

Recent progress on machine learning has led to technological innovations for applications such as autonomous driving, face recognition, and natural language processing. But it has also uncovered new privacy threats. For example, in a Membership Inference Attack (MIA), an attacker queries a machine learning model in order to infer whether a specific target record was part of the training dataset.

Although seemingly benign, inferring an individual's membership in a dataset can have serious privacy implications. For example, if the machine learning model was trained using medical records of patients suffering from a sensitive medical condition (e.g., cancer) then a successful membership inference is devastating as it reveals medical conditions an individual suffers from. A different way of conceptualizing membership inference is as a kind of re-identification attack from aggregated information (here the machine learning model). Viewed this way, as suggested in [26], protecting membership information is critical.

Recently, Shokri et al. [30] demonstrated the first MIA on classification models using only black-box access. This spurred further research into MIAs in a machine learning context [11], [28], [35]. In addition, some defensive measures have been proposed by Nasr et al. [24]. Despite this promising new research, there remains concerns about whether MIA indicate serious risks for widely used machine learning models. The concerns are two-fold. First, some prior attacks require the adversary to have control of the training algorithm [31], [35], which may not always be realistic in practice. Second, despite recent work [11], [28], [30], [35], it remains unclear what it means for membership inferences to be successful and what is the actual privacy risk. For example, what does it mean for an adversary to achieve 80% precision? What sensitive information does he learn about individuals in the dataset? Is the risk to all individuals the same or is there *disparate vulnerability* [6], [34]? What if the attack achieves only 51.7% precision?

Unfortunately, the attack methodology of prior work is ill-suited to answer these questions because it provides an incomplete picture of the risk. Indeed, prior work often measures the risk with respect to an *indiscriminate* adversary whose attack success is averaged over all targets, without regard to the cost of false positives or negatives. For example, (in one case) prior work [30] reports an attack precision of 51.7% (whereas the random guessing baseline is 50%). This unequivocally demonstrates the existence of successful MIAs but does little to elucidate the actual privacy risk because it is compatible with two vastly different scenarios: (1) 1.7% of individuals having their membership status permanently and unequivocally at risk and the other 98.3% being safe; and (2) all individuals having a probability of 0.517 (instead of 0.5) of having their membership status correctly guessed (and anything in between these scenarios). In terms of privacy violation, the first scenario is arguably much more serious.

In this paper, we propose a new attacker profile and corresponding methodology. We consider a *pragmatic* adversary who carefully selects targets based on their (perceived) vulnerability to membership inference and attempts to minimize false positives by trading off coverage for precision. We propose novel attacks that better match this setting and allows us to distinguish between two critical aspects of membership inference: (a) the attacker's success averaged over targets, and (b) the attacker's success for a specific target averaged over the random idiosyncrasies of the model's training data and choices during training (e.g., initial random weights values). From a privacy perspective, both (a) and (b) should be minimized by prospective defenses. In fact, differential privacy [8],

DOI 10.1109/EuroSP48549.2020.00040

the only defense with provable privacy guarantee, puts a tight bound on both. However, insofar as prior work has largely focused on (a), corresponding defenses may inadvertently overlook (b).

Our study reveals that a pragmatic adversary can achieve high precision (e.g., 95.05% on MNIST) in cases where prior work's methodology implies only barely above-the-baseline precision (i.e., 51.7%). It is worth noting that such findings occur even when the machine learning model is not overfitted, a setting for which prior work on black-box MIA reports significantly lower risk of membership privacy violation.

More specifically, our study finds that pre-selecting vulnerable records is an essential step to successfully attack well-generalized machine learning models. Without it, the attack has poor performance. With it, the pragmatic attack is able to infer membership of target records with high precision. This is because by focusing on those records likely to be vulnerable, it is easier for the attacker to detect those records' specific influence on the model. Furthermore, we find that the influence of vulnerable target records is far reaching and can be detected through indirect inferences that query the model on *enhancing records* related to the target records. In some cases, such indirect inferences outperform direct inferences. Moreover, the existence of enhancing records casts doubt on the effectiveness of defenses which rely on or aim to detect MIAs, because enhancing records often look nothing like the target record. Finally, we find that although regularization is effective in reducing the membership inference risk of *some* vulnerable individuals, it does not eliminate the risk for all individuals. Even when the model is regularized with a (relatively) large coefficient (i.e., $\lambda = 0.01$), the pragmatic attacker is able to identify a vulnerable record and infer its membership with almost maximal precision.

2. Background

2.1. Membership Inference Attacks

In a membership inference attack, the adversary's goal is to infer the membership status of a target individual's data in the input dataset to some computation. For a survey, the adversary wishes to ascertain, from aggregate survey responses, whether the individual participated in the survey. For machine learning, the adversary wishes to ascertain whether the target's record was part of the dataset used to train a specific model. A successful MIA is a privacy violation because it indicates that the target individual is identifiable from the aggregated statistics or models.

One of the first prominent examples of MIA occurs in the context of Genome-Wide Association Studies (GWAS). The seminal work of Homer et al. [13] show that p-values, a type of aggregated statistics routinely published when reporting the results of studies, could be used to successfully infer membership status. The experiment is performed on 86 individuals, all of which can be identified with few false positives. Although this attack requires that the adversary know the genome of the target individual, it teaches an important lesson: seemingly harmless aggregate statistics may contain sufficient information for successful membership inferences, which leads to re-identification of individuals in the study. As a consequence of this attack, NIH removed all aggregate data of GWAS from public websites [36].

More recently, it was shown that machine learning models are vulnerable to black-box membership inference attacks. Shokri et al. [30] cast the attack into a classification problem and show that an attack classifier can infer record membership with a precision of 93.5%, when the target model is overfitted and has a testing accuracy around 65%. However, the membership inference precision drops to 51.7% for models with testing accuracy greater than 90%. Hayes et al. [11] show that similar attacks are possible on generative target models, and Salem et al. [28] show that the attacker can also succeed only with access to data drawn from a different distribution and without knowledge of the target model structure. Similar to Shokri et al. [30], these attacks only work on overfitted models and indiscriminatively attack all records in the training dataset.

2.2. Evaluation Metrics for MIA

In their attacks against aggregated location data, Pyrgelis et al. [27] formalized MIA as a distinguishability game: the adversary tries to distinguish the membership of a set of users with probability greater than 0.5. The adversary's advantage is calculated as his chance of winning the game over random guessing.

Similarly, Yeom et al. [35] proposed formalization of MIA on machine learning models. They defined the adversary advantage as the probability of winning the distinguishing game over random guessing, where the probability is calculated over the selection of target record, randomness in the training process, and attack strategy.

Besides adversary advantage, attack precision is another commonly used metric for MIA. Both Shokri et al. [30] and Salem et al. [28] evaluated their attacks by the attack precision over a set of records containing a mix of members and non-members.

However, both the adversary advantage and the attack precision are calculated over a set of randomly picked records. This setup intrinsically assumes that the adversary would launch indiscriminate attacks on all the records. Meanwhile, in practice, an adversary could carefully select the attack targets and only perform attacks on the ones that are perceived to be vulnerable.

3. Adversary Model

We consider an adversary mounting a MIA against already trained machine learning models. We assume that the adversary has black-box access to the target models, i.e., he can issue arbitrary queries and retrieve the answers (e.g., the probability vector) from the models; the number of queries, however, may be limited.

In this section, we formulate the attack model used in prior MIAs as the indiscriminate attack and propose our more pragmatic attack model.

3.1. Indiscriminate Attack

In an *indiscriminate attack*, an attacker performs the attack over a set of randomly picked records, and the attack advantage is evaluated over *all* the records. Let D be a set of records, and $\mathcal{A}$ be the training algorithm. The attack could be described as the following distinguishing game between a user and an adversary:

1) The user randomly splits D into a training set S_{train} and a testing set S_{test} of the same size.
2) The user trains a model $M = \mathcal{A}(S_{\text{train}})$. The adversary has black-box access to M.
3) $\forall r \in D$, $x_r = 1$ if $r \in S_{\text{train}}$, otherwise $x_r = 0$.
4) $\forall r \in D$, the adversary obtains a guess $x'_r \in \{0, 1\}$.
5) $\forall r \in D$, the adversary succeeds if $x'_r = x_r$, otherwise the adversary fails.

The adversary's probability of success p is calculated as the number of successes of the adversary divided by the number of records in D. Prior work on MIA have used two metrics to evaluate the performance of an indiscriminate attack: the attack accuracy and the adversary advantage. In most attacks [28], [30], the attack performance is evaluated by the attack accuracy, which equals to the probability of success p, or the attack precision, which is the probability of success among all the positive inferences. In addition, Yeom et al. [35] defined the adversary advantage as the probability of winning the distinguishing game over random guessing, and the advantage is calculated as $2p-1$.

3.2. Pragmatic Attack

Although the indiscriminate attack model has been widely adopted in prior attacks and defenses, it ignores the potential influence of the cost of attacks and false positives. In this paper, we consider a *pragmatic attack*, where the adversary carefully selects attack targets and tries to minimize false positives. Let D be a set of records, and $\mathcal{A}$ be the training algorithm. We formalize the attack process as the following distinguishing game:

1) The adversary chooses a target $r \in D$.
2) The user randomly splits D into a training set S_{train} and a testing set S_{test} of the same size.
3) The user trains a model $M = \mathcal{A}(S_{\text{train}})$. The adversary has black-box access to M.
4) $x_r = 1$ if $r \in S_{\text{train}}$, otherwise $x_r = 0$.
5) The adversary produces a guess $x'_r \in \{1, \perp\}$ and performs an attack only if $x'_r = 1$.
6) If $x'_r = 1$ and $x_r = 1$, the adversary succeeds. If $x'_r = 1$ and $x_r = 0$, the adversary fails.

In practice, this attack process is carried out only once, and the attacker's advantage is his/her probability of winning the game. This probability is calculated over the randomness that is not controlled by the attacker, which includes the selection of the training dataset S_{train} and the randomness of the training algorithm $\mathcal{A}$. However, due to the complexity of ML models, it is challenging to obtain a theoretical bound on this probability. Therefore, we take a numerical analysis approach and estimate it using the Monte Carlo method. Specifically, we repeat the steps (2)-(6) to uniformly sampling the random space, which gives us a relatively accurate estimation of the success probability of an attack.

Pragmatic attacks are different from indiscriminate attacks in two aspects. First, instead of naively attacking all the records, a pragmatic adversary carefully selects the attack targets to avoid wasting time and resources on records that are unlikely to be vulnerable to membership inferences. The process of target record selection greatly reduces the chance of making false predictions and increases the probability of success. Second, a pragmatic adversary tries to minimize false positives because there is often a high cost for making false accusations. In a pragmatic attack, an adversary makes a positive inference (i.e., $x'_r = 1$) only if she has high confidence that the target record is in the training dataset, otherwise she makes no inferences (i.e., $x'_r = \perp$).

We define two metrics to evaluate the performance of the attack: (1) the *precision* of the attack is the probability of success among all the positive inferences (i.e. $\Pr[x_r = 1 \mid x'_r = 1]$); (2) the *coverage* of the attack is the probability of making a positive inference when the target record is in the training dataset (i.e. $\Pr[x'_r = 1 \mid x_r = 1]$). We evaluate the attack precision and coverage of each target record over the randomness of the training algorithm and sampling of training dataset.

The adversary makes a false positive inference when $x'_r = 1$ and $x_r = 0$. False positives are often associated with high cost and could reduce the adversary's credibility, so a pragmatic adversary attempts to minimize the number of false positives and maximize the attack precision. The adversary makes no inferences when $x'_r = \perp$, so a low coverage does not incur extra cost for the adversary. Therefore, it is acceptable to have a relatively low attack coverage.

3.3. Adversary Knowledge

Similar as the previous work [30], we further assume that the adversary either (1) knows the structure of the target model (e.g., the depth and the number of neurons in each layer of the neural network) and the training algorithm used to build the model, or (2) has black-box access to the machine learning algorithm used to train the model. We also assume that the adversary has some prior knowledge about the population from which the training records are drawn. Specifically, the adversary can access a set of records that are drawn independently from that population, which may or may not overlap with the actual training data for the target models; but the adversary does not have any additional information about whether these records are present in the training data. These records can often be obtained from a public dataset with similar attributes or from previous data breaches.

4. Pragmatic Membership Inference Attack

4.1. Attack Overview

The goals of our attack are different from the goals of an indiscriminate attack in two aspects: (1) the adversary is only interested in positive membership inferences because positive membership information is more valuable to the adversary and more risky to the users. Positive membership inference allows the adversary to associate public available information (i.e., the machine learning models) with some identifiable auxiliary information (i.e., the record of an individual known to the adversary). Because this association is similar to re-identifying individuals in an anonymized dataset, positive membership

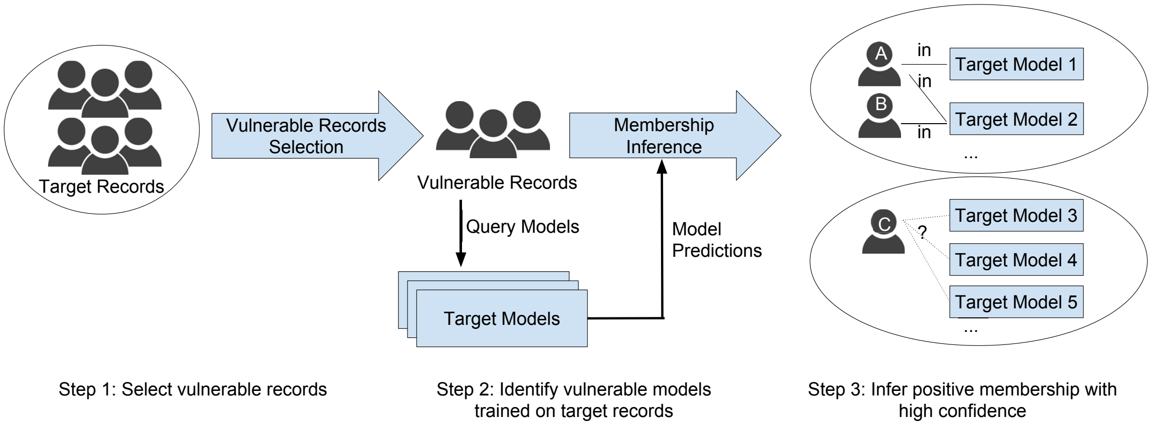


Figure 1: Attack Overview. Our attack consists of 3 steps. First, the adversary selects vulnerable records from a group of target records. Second, the adversary generate queries related to the vulnerable records and query the target models. Finally, the adversary identifies target models that are trained with the vulnerable records. The adversary only makes a positive membership inference if a target model's predictions strongly indicate the presence of a target record in its training dataset. Otherwise, the adversary makes no inferences. In the above example, the adversary would infer that target model 1 is trained with record A and target model 2 is trained with records A and B. However, the adversary would not make any inferences on record C and target models 4,5,6 because the predictions of the models do not give it high confidence for positive inferences.

inference attack has been considered as a type of re-identification attack in prior work [26]. Moreover, given a correct positive membership inference, the adversary knows that the individual is a participant of a study, which may further leak more information about that individual. (2) The adversary wants to re-identify individuals in the training dataset *with high precision* because false inferences can be costly. Around these goals, we design a three-step pragmatic attack as shown in Fig. 1. Below, we briefly explain each step of the attack.

Step 1: Selecting Vulnerable Target Records. In an overfitted model, almost all records are vulnerable to MIA, so a indiscriminate attack can achieve high precision. However, when the model is well-generalized, the model gives similar predictions to members and non-members of the training dataset. Therefore, identifying vulnerable target records is the key to an effective pragmatic attack. First, we select vulnerable records by estimating the number of neighbors they have in the sample space represented by the records available to the adversary. Records with fewer neighbors are more vulnerable under MIA because they are more likely to impose unique influence on the machine learning models. In order to identify neighbors of a given record, we train reference models to imitate the behavior of target models. We further construct a new feature vector for each record based on the intermediate outputs of reference models on this record, which implies this record's influence on the target machine learning model.

Step 2: Identifying Vulnerable Models. Next, we query the target models and identify the models that are trained on target records. Specifically, we design two attack methods distinguished by their queries to the target models: A direct inference attack infers the membership of a target record based on the model's prediction on that record; an indirect inference attack infers the membership of a target record based on the record's influence on the model's predictions on seemingly uncorrelated records (called *enhancing records*). We use novel techniques that iteratively search for and select enhancing records. Our indirect inferences using the enhancing records can successfully infer the presence of a target record without querying it. Moreover, the indirect inferences sometimes outperform direct inferences by accumulating more information from multiple queries. Note that although we design and evaluate our attack with multiple target records and target models, in practice, the adversary may choose a single target model or a single target record to attack.

Step 3: Inferring Positive Membership. Finally, we make positive membership inference over the combinations of all target records and target models. Since there is often a high cost when making incorrect inferences, we only infer a target record to be in the target model if the predictions of the model indicate a high probability of success in the attack. We use hypothesis testing methods to make the decision: under the null hypothesis the record is not present in the training dataset; under the alternative hypothesis the target record is in the training dataset. We reject the null hypothesis when the p-value is smaller than a cut-off threshold.

4.2. Building Reference Models

We exploit a target record's unique influence on the outputs of a machine learning model to infer the presence of the record in the training set of the target model (called target training set). To identify such influence, we need to estimate the model's behavior when the target record is *not* in the target training set. To achieve this goal, we build *reference models*, which are trained using the same algorithm on *reference datasets* sampled from the same space as the target training set, but not containing the

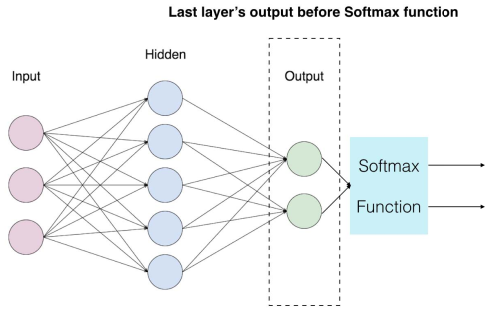


Figure 2: Last layer output of a two-layer neural network. We use the last layer output of locally trained nueral networks as features for vulnerable record selection.

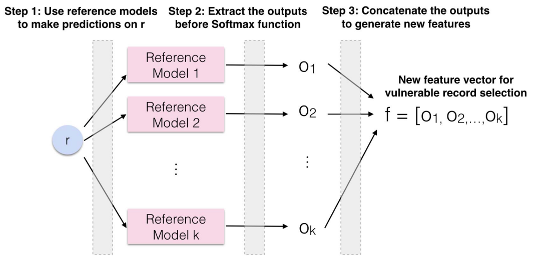

Step 1: Use reference models to make predictions on r
Step 2: Extract the outputs before Softmax function
Step 3: Concatenate the outputs to generate new features
r
Reference Model 1
Reference Model 2
⋮
Reference Model k
o1
o2
⋮
ok
New feature vector for vulnerable record selection
f = [o1, o2,...,ok]

Figure 3: Features for vulnerable records selection. We concatenate intermediate outputs of locally trained reference models and use them as features for vulnerable records selection.

target record. The process of building reference models are illustrated below.

To start with, we need to construct k reference datasets with the same size as the target training set. Since most practical machine learning models are trained on large training datasets, it is difficult for an adversary to get access to an even larger dataset with k times records as the target training set. Consequently, if we build the reference datasets by sampling without replacement from the whole set of reference records, the resulting datasets may share many records, and the reference models built from them would be alike and give similar outputs. To address this issue, we use bootstrap sampling [9] to generate the reference datasets, where each dataset is sampled with replacement. Bootstrap sampling reduces overlaps among reference datasets, providing a better approximation of datasets sampled from distribution of the target training set. Each reference dataset is then used to train a reference model using the same training algorithms as used for training the target model.

4.3. Selecting Vulnerable Records

Not all training records are vulnerable to MIA. In general, we want to measure the potential influence of a target record so as to select vulnerable records with the greatest influences and subject them to MIA in the subsequent steps. It is worth noting that, although the training records imposing unique influence on the model are often *outlier records* (i.e., with distinct feature vectors) in the training set, the outlier records do not always have unique influence on the model because the training algorithm may decide that some features should be given higher weights than others and some features should be combined in the model. For example, a neural network trained on hand written digit datasets learns the contour of written digits is more important feature than individual pixels [20]. Therefore, instead of using the input features, we extract high level features more relevant to the classification task to detect vulnerable records.

Specifically, when attacking neural networks (e.g., see Figure 2 for a two-layer fully connected neural network), we construct new feature vectors by concatenating the outputs of the last layer before the Softmax function from the reference models (Figure 3), as the deeper layers in the network are more correlated with the classification output [12]. For other classification models without intermediate layers, the new feature vector can be generated by concatenating the model's prediction vectors. We then measure the unique influences of each record using its new feature vector. Let $\mathbf{f}$ be the the new feature vector of the record r. We call two records r_1 and r_2 *neighbors* if the cosine distance between their feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ is smaller than a *neighbor-threshold* α.

Note that the neighboring records are difficult to be distinguished by MIA because they have similar influence on the model. When a neighbor of r occurs in the training dataset, the model may behave as if r is used to train the model, leading to the incorrect membership inference result. Our goal is to select the vulnerable records in the entire record space with fewer or no neighbors likely to be present in the training set (assuming the training records are independently drawn from the record space) as putative targets of MIA.

Given a training dataset with N records and a reference dataset with N' records, both sampled from the same record space, and a target record r, we count the number of neighbors of r in the reference dataset, denoted as N'_n. Then, the expected number of neighbors of r in the training dataset, N_n, can be estimated as $\mathbb{E}[N_n] = N'_n \times \frac{N'}{N}$.

A record r is considered to be potentially vulnerable (and as the attack object), only if $\mathbb{E}[N_n] < \beta$, where β is the *probability-threshold* for target record selection. We stress that the approach for vulnerable records selection presented here relies only on the record space (represented by the reference records accessible by an adversary) and the reference models (built using reference records), and is independent of the target model; as a result, the computation can be done off-line even when used to attack a machine learning as a service (MLaaS).

4.4. Direct Inference

In a pragmatic attack, the goal of the adversary is to achieve high precision on the selected target records instead of all the records. Therefore, we attack each target record separately by computing the deviation between its output given by the target model and those given by the reference models. We expect that each training record has a unique influence on the model, which can be measured by comparing the target model's output with the output

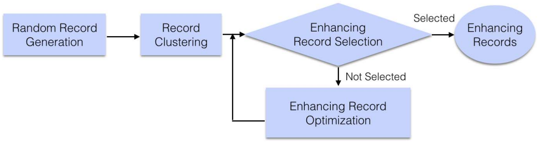


Figure 4: Steps for generating enhancing records.

of reference models (trained without the target record) on the record. We quantify the difference between the outputs using the log loss function. Given a classifier M and a record r with class label y_r, let p_{y_r} be M's output probability of class label y_r. The log loss function [23] $\mathcal{L}(M,r)$ is defined as:

$$\mathcal{L}(M,r) = -\log p_{y_r}.$$

The log loss function is commonly used as a criterion function [23] when training neural network models. $\mathcal{L}(M,r)$ is small when M gives high probabilities on correct labels.

Given a target model M, a target record r, and k reference models, we first obtain the log loss of all the reference models on r as $L_1, L_2, \ldots, L_k$. We view these losses as samples independently drawn from a distribution $\mathcal{D}(L)$, and estimate the empirical cumulative distribution function (CDF) of $\mathcal{D}_L$ as $F(L)$, which takes a real-valued loss L as input. We use the shape-preserving piecewise cubic interpolation [32] to smooth the estimated CDF. Based on the log loss of the target model M on the target record r, $\mathcal{L}(M,r)$, we estimate the confidence of r to be present in the training set by performing a left-tailed hypothesis test: under the null hypothesis H_0, r is not present in the training set (i.e., $\mathcal{L}(M,r)$ is randomly drawn from $\mathcal{D}(L)$), while under the alternative hypothesis r is used to train M (i.e., $\mathcal{L}(M,r)$ is smaller than samples in $\mathcal{D}(L)$ because of the influence of r in the training). Therefore, we calculate the p-value as:

$$p = F(\mathcal{L}(M,r)),$$

which gives the confidence that r is used for training M only if p is smaller than a threshold (e.g. 0.01) so that the null hypothesis is rejected.

4.5. Indirect Inference

Besides reducing a model's loss on its own, a training record also influences the model's outputs on other records. This influence is desirable to improve model generalization: in order to give correct predictions on unseen records, a model needs to use the correlation it learns from a training record to make predictions on queries with similar features. On the other hand, however, these influences can be exploited by an adversary to obtain more information about the target record through multiple queries to enhance MIA. Interestingly, we show that MIA can be achieved by queries of records seemingly uncorrelated with the target record, making the attack hard to detect and defend against.

The key challenge for inference without querying the target record is to efficiently identify the *enhancing records* whose outputs from the target model are expected to be influenced by the target record. To address this problem, we develop a method consisting of the following steps: random record generation, record clustering, enhancing record selection, and enhancing records optimization (as shown in Figure 4).

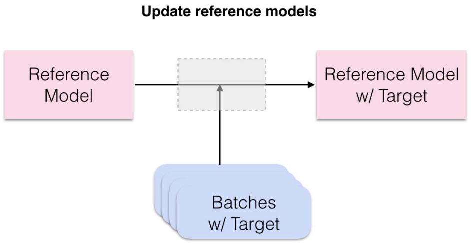


Figure 5: Building positive reference models by updating the model with the training set including the reference records plus the target record.

Random Record Generation. To start with, we randomly generate records from which the enhancing records are selected. Specifically, we adopt one of the following two methods for random record generation: (1) when the feature space is relatively small, we uniformly sample records from the whole feature space; (2) when the feature space is large, since the chance of getting enhancing records by uniform sampling is slim, we generate random records by adding noise to pre-selected vulnerable target records. We use Gaussian noise for numerical attributes and candle noise [38] for categorical attributes.

Enhancing Record Selection. To identify records whose target model's output may be influenced by the target record r, we approximate the target model's behavior using a group of *positive reference models* that are trained using reference records plus the target record r. To save the effort of retraining the positive reference models, we add the target record into batches sampled from the original reference dataset and update the reference models by training on the batches plus the target record. Figure 5 shows the process of updating reference models.

We select the *enhancing records* by comparing the predictions between the positive reference models (i.e., "in models") and the original reference models (that are trained without the target records, i.e., "out models"). We denote the ith original and the ith positive reference model as M_{ref_i} and $M^r_{\mathrm{ref}_i}$, respectively. Given a record r with class label y_r and another arbitrary record q, let $M(q, y_r)$ be the model M's output probability of y_r on the query q. We calculate r's influence on q as follows:

$$I(r,q) = \frac{1}{k}\sum_{i=1}^{k} \mathrm{t}\left(M^r_{\mathrm{ref}_i}(q, y_r) - M_{\mathrm{ref}_i}(q, y_r)\right), \quad (1)$$

where k is the total number of original (or positive) reference models, and t is a threshold function defined as follows:

$$\mathrm{t}\left(x\right)=\begin{cases}1 & \text{if} \quad x>0,\\ 0 & \text{otherwise.}\end{cases}$$

Algorithm 1 Enhancing Records Selection Algorithm

```
procedure $\mathrm{select}_\theta(q)$          ▷ Input a random query
    $I\left(r,q\right) \leftarrow \sum_{i=1}^{k} \mathrm{t}\left(M^r_{\mathrm{ref}_i}\left(q,y_r\right) - M_{\mathrm{ref}_i}\left(q,y_r\right)\right)/k$
    if $I > \theta$ then
        Accept $q$                        ▷ Use $q$ in MIA
    else
        Reject $q$
```

We identify a randomly generated record q as an enhancing record for the record r if $I(r,q)$ approaches 1, which indicates that adding r to the training dataset *almost always* increase the models' output probability on the class label y_r for the query q. In practice, we use q in the MIA on the target record r only if $I(r,q)$ is greater than a threshold θ (e.g. 0.95). Algorithm 1 summarizes the entire algorithm for query selection.

Enhancing Record Optimization. When the target model has a large record space (e.g., with high-dimension feature vectors), the chance of finding an enhancing record among randomly generated records is slim. To address this issue, we propose an algorithm to search for enhancing records for a target record r by optimizing the following objective function:

$$\max_{q} I\left(r,q\right), \tag{2}$$

where $I\left(r,q\right)$ is the influence function defined in Equation 1. Optimizing $I\left(r,q\right)$ is time-consuming because $I\left(r,q\right)$ consists of a non-differentiable threshold function t. Therefore, instead of solving the optimization function in equation 2, For simplification, we approximate the maximization of $I\left(r,q\right)$ with the minimization of the sum of multiple hinge loss functions defined as follows [10]:

$$\min_{q} \sum_{i=1}^{k} \max\left(0, \gamma - \left(M^r_{\mathrm{ref}_i}\left(q,y_r\right) - M_{\mathrm{ref}_i}\left(q,y_r\right)\right)\right), \tag{3}$$

where γ is a parameter indicating the margin width. If a randomly generated record are rejected by the query selection algorithm, we minimize the objective function in Equation 3 using gradient descent [7] to check if the resulting record is acceptable as an enhancing record.

Record Clustering (Optional). Note that it is inefficient to repeat the query selection and optimization algorithms on all random records because the predictions of the models on most records are highly correlated: the models giving high output probabilities on some record are also likely to give high output probabilities on correlated records. To improve the efficiency of query selection, we propose an algorithm to identify the *least correlated* enhancing records from a large number of randomly generated records.

First, we estimate the correlation between records based on the model's predictions on them. We construct a feature vector $\mathbf{f}_q$ for a record q by concatenating the reference models' outputs on it (Figure 6). If two queries q_1 and q_2 have highly correlated feature vectors, the models' outputs on q_2 do not add much information to the models' outputs on q_1.

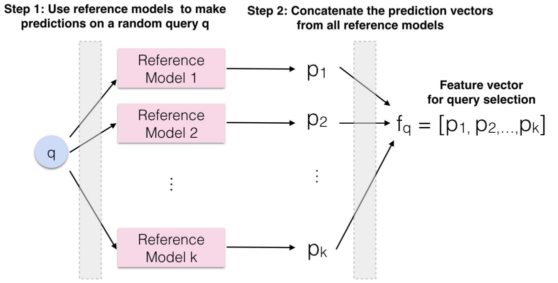

Step 1: Use reference models to make predictions on a random query q
Step 2: Concatenate the prediction vectors from all reference models
q
Reference Model 1
Reference Model 2
Reference Model k
p1
p2
pk
Feature vector for query selection
fq = [p1, p2,...,pk]

Figure 6: Generating query features for query selection.

Next, we formulate the problem of selecting a subset of least correlated records as a graph theoretical problem. We build a graph where records are the nodes and pairwise correlation between records is the weight on edges connecting the corresponding nodes. This allows us to recast our problem as the k-lightest subgraph problem [33], which is NP-hard. We obtain an approximate solution using hierarchical clustering [15]. For this, we cluster the records into k disjoint clusters based on their pairwise cosine distance. Finally, in each cluster, we select the record with least average cosine distance to all other records in the same cluster.

As shown in Figure 4, we use the enhancing record clustering algorxithm before the enhancing record selection and enhancing record optimization steps to improve the efficiency of the attack.

Indirect Inference with Multiple Queries. After identifying multiple enhancing records, we repeat the attack in section 4.4 by querying each of these records. Because the outputs on these queries may be correlated, we combine the resulting p-values using Kost's method [18], with the covariance matrix estimated from the query features generated in the query selection step (Figure 6).

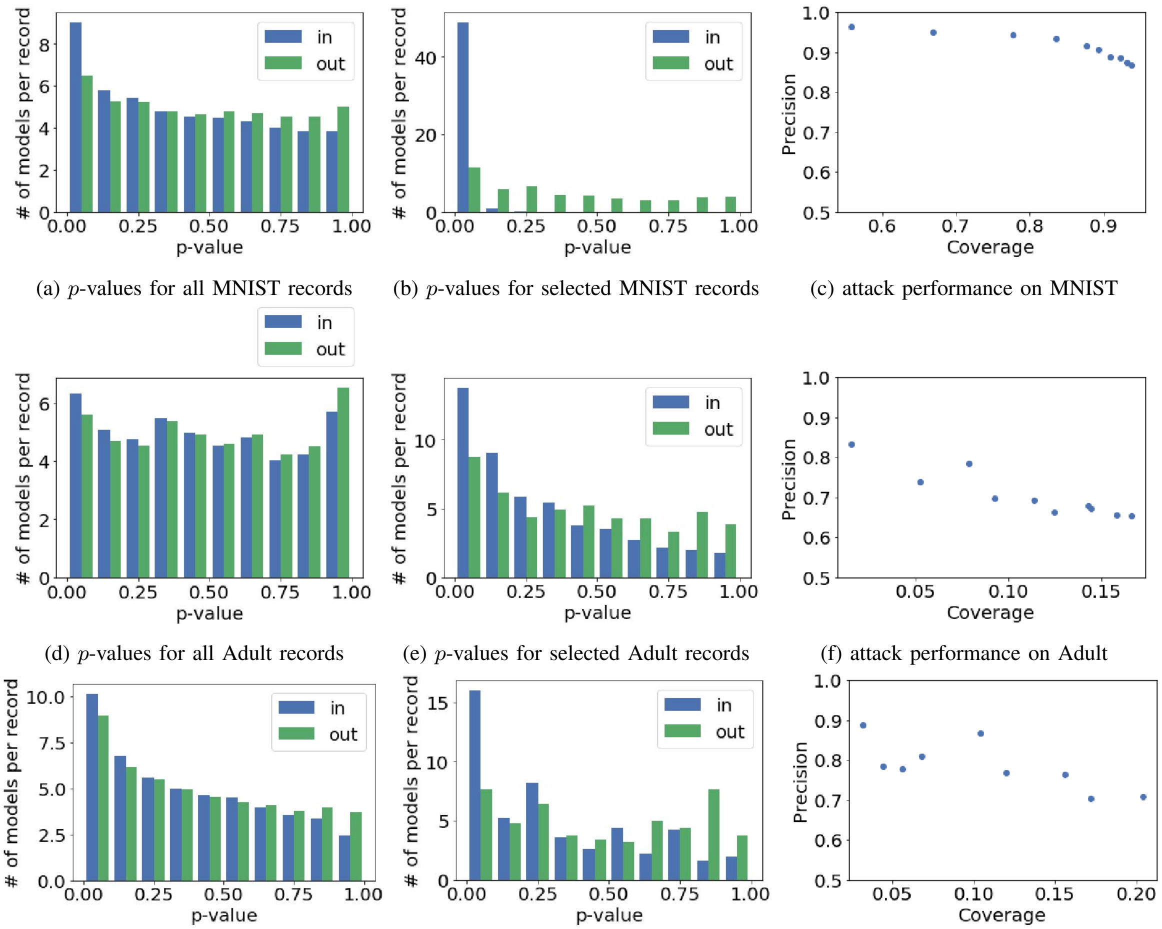
5. Evaluation

5.1. Experimental Setup

We evaluated the performance of our attack from the following three aspects: the precision of the attack, the coverage of the attack, and the effectiveness of vulnerable record selection method.

For each dataset, We constructed 100 target models. To get a better understanding of MIA's performance, we wanted the baseline precision to be 0.5 for each target record. That is, each target record should occur in 50 out of the 100 target models. Therefore, we generated training datasets by randomly splitting the target records into two datasets of the same size, each serving as a training set for a target model. We repeated this process 50 times and generated the training datasets for 100 target models.

The *precision* of the attack is the percentage of successful inferences (i.e., the target record is indeed in the training dataset) among all inferences. The *coverage* of the



(a) p-values for all MNIST records

(b) p-values for selected MNIST records

(c) attack performance on MNIST

(d) p-values for all Adult records

(e) p-values for selected Adult records

(f) attack performance on Adult

(g) p-values for all Cancer records

(h) p-values for selected Cancer records

(i) attack performance on Cancer

Figure 7: Evaluation of MIA on MNIST, Adult, and Cancer dataset. In (a), (d), and (g), we performed hypothesis testing on *all* records over 100 target models. There was no significant difference between the distributions of p-values for models trained with the target record (labeled "in") and models trained without the target record (labeled "out"). This result indicates that the attack cannot achieve high precision without the step of vulnerable record selection. In (b), (e), and (h), we performed the same hypothesis testing on the selected vulnerable records over 100 target models. Our attack was effective because there was a distinction between the p-value distributions of "in" models and "out" models. Most models with a small p-value were models trained with the target record (i.e. "in" models). (c), (f), and (i) show the varying precision and coverage with cut-off p-value ranging from 0.005 to 0.05. Our attack focused on achieving high attack precision because false accusations can be costly for attackers.

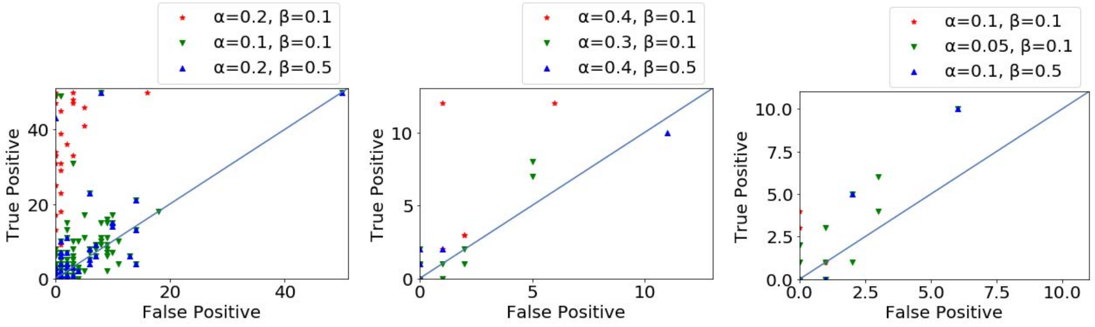
attack is the percentage of successful inferences among all the cases that the target record is in the training set (i.e. 50 times the number of records). In practice, a high precision is often more important than a high coverage because there is usually a high cost associated with making false inferences.

We define *true positive (TP)* to be the case that the target record is indeed in the training dataset when the adversary inferred it as in and *false positive (FP)* to be the case that the target record is *not* in the training dataset when the adversary inferred it as in. We evaluate the effectiveness of our vulnerable record selection method by looking at the true positives and false positives of each vulnerable record under different selection criteria.

5.2. Dataset

UCI Adult. The UCI Adult dataset [21] is a dataset extracted from 1994 American Community Survey. It contains 48,842 records and 14 attributes. The attributes are demographic features and the classification task is to predict whether an individual's salary is above $50K a year. We normalized the numerical attributes in the dataset and used one-hot encoding to construct the binary representation of categorical features. We randomly selected 20,000 records for training target models, and each training dataset contains 10,000 records. The remaining 28,842 records served as the adversary's background knowledge.

UCI Cancer. The UCI cancer dataset [21] contains 699 records and 10 numerical features ranging between



(a) MNIST Dataset

(b) Adult Dataset

(c) Cancer Dataset

Figure 8: Effectiveness of vulnerable record selection in a pragmatic attack. We evaluated the effectiveness of vulnerable record selection by plotting the number of true positive inferences and false positive inferences of each selected record. Each point in the figure represents a target record. Points can overlap because the attack can have the same performance on different records. For each target record, the attack was performed over 100 target models (50 "in" models and 50 "out" models). Records selected under different criteria were plotted with different colors and shapes. Records with cosine distance smaller than α were considered as neighbors. We selected records with probability threshold β. That is, the probability that the record's neighbor would occur in the training dataset was smaller than β. Points at the upper left corner are more vulnerable to MIA than those near the baseline. Most of the points are above baseline. This result demonstrates that our vulnerable record selection method is effective. Moreover, by adopting a strict selection criterion (large α and small β) we identified records that can be attacked with high precision (e.g., red points in the figures).

1 to 10. The features are characteristics of the cell in an image of a fine needle aspirate (FNA) of a breast mass. The classification task is to determine whether the cell is malignant or benign. We randomly selected 200 records for training, and each training dataset contains 100 records. The remaining 499 records served as the adversary's background knowledge.

MNIST Dataset. The MNIST dataset [19] is an image dataset of handwritten digits with 60,000 handwritten training examples and 10,000 testing examples. The images are normalized such that the digits are positioned at the center of the 28x28 pixel field. The classification task is to predict which digit is represented in an image. We randomly selected 20,000 images for training and 40,000 images as the adversary's background knowledge. Each training set for target models and reference models contains 10,000 images. We used the 10,000 testing images to calculate testing accuracy.

5.3. Models

Neural Network. For the Adult dataset, we constructed a fully connected neural network with 2 hidden layers with 10 units and 5 units respectively. We use `Tanh` as the activation function and `SoftMax` as the output layer. The model is trained with batchsize of 100 and 20,000 epochs. For the MNIST dataset, we constructed 2 convolutional layers with ReLu as the activation function, followed with max pooling layers. We then added a fully connected layer of 1,024 neurons, and we also used dropout techniques to reduce overfitting. Finally, we added an output layer and a Softmax layer. The model is trained with batchsize of 50 and 10,000 epochs. For the Cancer dataset, we used a vanilla neural network with no hidden layer. The model is trained with batchsize of 10 and 3,000 epochs.

Dataset (Model)	Vulnerable Records	Precision	Coverage
MNIST	27	95.05%	66.89%
Adult	13	73.91%	5.23%
Cancer	5	88.89%	3.20%
MNIST (Google)	1	100%	4%
Adult (Google)	7	80%	2.67%

TABLE 1: Performance of Direct Inference. We measured the performance of a direct inference attack by its precision and coverage. To achieve a high precision, we selected a few vulnerable records (neighbor threshold $\alpha = 0.2$ for MNIST, 0.4 for Adult, and 0.1 for Cancer; probability threshold $\beta = 0.1$), and made positive inferences only when attack confidence is high ($p \leq 0.01$).

Google ML Engine. Since the Google Predictions API [2] used in the prior attack is deprecated, we used Google ML Engine [1] to train target models on ML cloud. When training the model, we used the sample code provided by Google, which has pre-built model structures for training models on Adult dataset and MNIST dataset. Specifically, for Adult dataset, the sample code uses Google estimator [3] which hides low-level model structure from the user; for MNIST dataset, the sample code builds a neural network with 2 fully-connected hidden layers.

5.4. Direct Inference

We evaluated the performance of direct inferences by their precision and coverage on different datasets and models. We set a fixed vulnerable record selection criterion for each dataset. The neighbor threshold α was 0.2, 0.4, and 0.1 for MNIST, Adult, and Cancer respectively. This threshold represented the maximum cosine similarity between neighbors. Records with cosine similarity smaller

Dataset (Model)	Training Accuracy	Testing Accuracy
Adult	0.85 ± 0.01	0.85
Cancer	0.95 ± 0.04	0.94 ± 0.03
MNIST	0.99	0.98
Adult (Google)	0.84 ± 0.03	0.84 ± 0.02
MNIST (Google)	0.90	0.90

TABLE 2: Training and Testing Accuracy of Target Models. All the target models were well-generalized models with difference between training and testing accuracy smaller than 0.01.

than α were considered as neighbors. Therefore, this threshold varied for different datasets depending on the dimensionality of records. We evaluated the influence of this threshold later in this section. We selected records with probability threshold $\beta = 0.1$. That is, the likelihood that a neighbor of the record occurs in the training dataset was smaller than 0.1.

In Fig. 7 we plotted the average number of models per record with different attack p-values. The models trained with the target records are labeled as "in" and the models trained without the target records are labeled as "out". The attack was effective only when there was a distinction between the p-value distribution of "in" models and "out" models. The figure shows the necessity of selecting vulnerable records before doing the inferences. When the attack hypothesis testing was performed on *all* target records, the p-value distributions of "in" models and "out" models were indistinguishable. Therefore, the attack was unlikely to have a high precision no matter what p-value cutoff we selected. On the other hand, when we performed the same hypothesis testing on the selected vulnerable records, there was a clear distinction between the p-value distributions of "in" models and "out" models, which led to successful membership inference attacks.

The cut-off p threshold controls the trade-off between precision and coverage. We would only make a positive inference if the p-value obtained from the attack is smaller than the threshold. Since there is a cost for false inferences, we chose small p thresholds in the attack. In Fig. 7, we plotted the different attack precision and coverage obtained by cut-off p threshold varying between 0.005 to 0.05.

Our attack mechanism was less effective on the Adult Google ML model since we did not have access to the exact model structure due to the use of Google estimator. Instead, we used raw features to select target records. This limitation reduced the number of vulnerable target records we identified from 13 to 7. However, it also shows that the attack is possible even when the adversary does not know the model structure.

5.5. Selection of Vulnerable Records

The p-value distributions in Fig. 7 shows the importance of vulnerable records selection. We further explored how this step influenced the attack performance by changing the neighbor threshold α and the probability threshold β. Fig. 8 shows the records selected by different selection thresholds. Each point in the figure represents a target record. Points at the upper left corner are more vulnerable to MIA than those near the baseline. Smaller neighbor thresholds or higher probability thresholds increased the number of selected vulnerable target records. However, as we tried to attack more records at the same time, there was a higher chance that we would make false positive inferences due to the influence of a record similar to one of the target records, which decreased the attack precision.

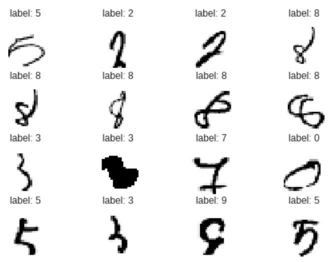


Figure 9: Vulnerable Examples in MNIST Dataset

Dataset	Cut-off p-value	Prec. (direct)	Coverage (direct)	Prec. (indirect)	Coverage (indirect)
Adult	0.01	-	0	1	14%
	0.1	70.83%	34%	75%	24%
Cancer	0.01	1	6%	-	0
	0.1	66.67%	52%	88.89%	16%
MNIST	0.01	96.15%	1	1	2%
	0.1	89.29%	1	52.38%	22%

TABLE 3: Comparison between direct and indirect inferences. We performed the attack on the same selected record with direct inference and indirect inference. The result indicates that membership inference attack is feasible without directly querying the target record. On the Adult dataset, indirect inferences even outperformed direct inferences.

To study what kinds of records are vulnerable, we plotted the vulnerable target records selected from MNIST dataset with $\alpha = 0.2$ and $\beta = 0.1$ (Figure 9). As we expected, some of the vulnerable target records are outliers in the dataset. However, some vulnerable examples actually increase model utility by providing rare but useful features for the classification task. For example, the images of the digit 8 written in different directions may help a model on recognizing similar written digits in testing examples. However, since these images are rare in the dataset, they have a unique influence on the target models, making them vulnerable to our attack, and the fact that this influence is useful in predicting unseen examples does not mitigate the risk.

5.6. Indirect Inference

For some vulnerable target records, we achieved the same level of attack performance by querying enhancing records. For each dataset, we randomly sampled $5,000$ records, selected 50 of them by record clustering, and tested them with the enhancing record selection algorithm. If less than 10 enhancing records were selected, we ran the enhancing record optimization algorithm to improve the records. The initial records for the Cancer dataset and the Adult dataset were randomly sampled from the feature space while the records for the MNIST dataset

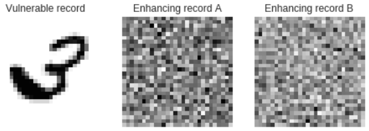


Figure 10: A vulnerable record from MNIST with its two enhancing records. In practice, it is difficult to find out what the target record is, by looking at the enhancing records used by an adversary.

were generated by adding noise to the target records due to the large feature space.

We selected 1 target record in each dataset. For the Cancer dataset, we selected 47 enhancing records whose euclidean distance to the target record range between 6 and 19.3 with a selection criterion $I\left(r,q\right) > 0.95$. Since the Cancer dataset has relatively low dimensional features, enough enhancing records were accepted, and enhancing record optimization was not needed. For the Adult dataset, we relaxed the enhancing record selection criterion to $I\left(r,q\right) > 0.9$ and found 15 enhancing records after the optimization step. For the MNIST dataset, we further relaxed the enhancing record criterion to $I\left(r,q\right) > 0.8$ due to the high dimensional feature space. We identified 41 enhancing records generated by adding noise to the target record.

Table 3 shows the performance of indirect inferences. For both the Cancer dataset and the Adult dataset, attacking with the enhancing records has compatible performance as querying the target record. Moreover, for the Adult dataset, querying the target record did not successfully infer any cases with a 0.01 cut-off p-value, but by combining the predictions on enhancing records, we achieved a precision of 1 and a coverage of 14%. For the MNIST dataset, we achieved a precision of 1 and a coverage of 2% when $p \leq 0.01$. Although this performance is less impressive compared to a direct inference on the same record (whose precision and recall are both close to 1) it's still an indication that membership inference attack can succeed without querying the target record.

The effectiveness of indirect inferences shows that prior defenses [16], [25] based on direct inferences could *fail* to eliminate the risk of membership inferences. Moreover, we plotted both the target record and the enhancing records and found that the enhancing records in no means represent the target record, indicating that our attack is hard to detect (Figure 10).

5.7. Influence of Regularization

Regularization is a common method for improving model generalization. It is shown to be an effective defense against prior MIAs [30]. To study its effectiveness on our attack, we applied L2 regularization on neural networks trained on MNIST set even though the models were *not overfitted*. In doing so, we limited the model capacity which increased the risk of underfitting. Specifically, when the regularization coefficient λ went from 0.001 to 0.01, testing accuracy decreased by 0.01 indicating that the model might be underfitted due to over regularization.

Regularization Coefficient λ	Training Acc.	Test Acc.	# of Target Records	Prec.	Coverage
0	0.99	0.98	52	90.84%	68.31%
0.001	0.99	0.99	1	1	54.8%
0.01	0.98	0.98	1	93.36%	4%

TABLE 4: Attack Performance w.r.t. Regularization ($\alpha = 0.2$, $\beta = 2$, $p \leq 0.01$) on MNIST dataset. We applied L2 regularization with varying coefficients λ. Experiment results show that applying regularization reduced, but did fully eliminate the privacy risk of a pragmatic adversary.

Dataset	Attack Confidence Threshold	Precision	Coverage
Cancer (3 records)	0.8 0.9	50.25% -	40% 0
Adult (13 records)	0.6 0.7	66.67% -	4.92% 0
MNIST (27 records)	0.6 0.7 0.8	50% 19.6% -	56.25% 6.25% 0

TABLE 5: Performance of the attack of Shokri et al. [30] on the same target models and the same target records. To imitate the attack strategy of a pragmatic adversary, we performed prior attack on the selected target records and made predictions only when the attack classifier has high confidence. However, the prior indiscriminative attack could not achieve high precision even under a low coverage.

Table 4 shows model accuracy and attack performance before and after applying L2 regularization with varying coefficients λ. Applying regularization reduced the number of vulnerable target records in the dataset, but did not completely eliminate the privacy risk. The remaining vulnerable records were attacked with high precision. Specifically, when L2 regularization was applied with coefficient $\lambda = 0.01$, we still identified 1 vulnerable target record, which was inferred with precision close to 1.

Applying regularization mitigated the model's privacy risk of *some* vulnerable individuals but did not eliminate the risk of *all* individuals. Moreover, since the most vulnerable record was identified with high precision, regularization may not be a good approach when the data owner wants to provide privacy protection for *all* individuals whose records are in the dataset.

5.8. Comparison with Indiscriminative Attacks

To compare with the attack proposed by Shokri et al. [30], we reproduced the attack on the same target models and the same vulnerable records in our attack. Specifically, we trained one attack classifier per class for each dataset. The attack classifiers are neural networks with one hidden layer of 64 units. We used `ReLU` as the activation function and `SoftMax` as the output layer. We only performed the attack when the probability given by the attack classifier was higher than a certain threshold (called attack confidence threshold). We evaluated the performance of the attack under various attack threshold as shown in Table 5. The attack precision was relatively low (e.g. $< 70\%$) on all three datasets even when a high attack confidence threshold was used.

6. Discussion

In this section, we explain the limitations of our attack, and further discuss the potential mitigation to the information leaks in machine learning models.

6.1. Limitations

We view our attack as preliminary because our techniques for identifying outliers cannot find all vulnerable instances: it is possible that some instances not considered to be outliers by our current design still exert unique influences on the model, which need to be better understood in the follow-up research. Moreover, the current way to search for enhancing records, through filtering out random queries, is inefficient, and often does not produce any results. More effective solutions could utilize a targeted search based upon a better understanding about the relations between the target record and other records. Fundamentally, it remains unclear how much information about the training set is leaked out through querying a machine learning model and whether more sensitive techniques can be developed to capture even small signals for a record's unique impact.

6.2. Mitigation

Differential Privacy. Differential privacy has been shown to limit the advantage of membership inference attacks [35]. However, the protection of differential privacy comes at the price of loss in model utility. Recent work has shown that current mechanisms for differentially private machine learning rarely offer acceptable utility-privacy trade-offs for complex learning tasks [16]. For example, the differentially private deep learning model [4] suffers a 10% accuracy loss on MNIST (under $\epsilon = 1$) and achieves only 65% accuracy on CIFAR10 (whereas state-of-art non-private models can achieve 99% accuracy). While differential privacy training of ML models is a viable option in some scenarios, the goal and main focus of our work is to provide a better understanding of non-differentially private models. This is an important research goal for two reasons. First, because the utility cost of differential privacy can only be justified if there is in fact a significant privacy risk when models are not trained to be differentially private. Second, because the research can shed light on other less costly privacy protection mechanisms such as privacy-preserving training record selection.

Adversarial Regularization. Prior research has used adversarial regularization [25] and adversarial examples [16] as defenses against membership inference attacks. However, both defenses assume an indiscriminate adversary that trains an attack classifier based on the model's *direct* prediction on the target records. In contrast, the pragmatic attack strategy in our work is more sophisticated, and our indirect queries are hard to identify. Therefore, it is challenging to model the attacker for privacy regularization or to generate adversarial examples for the attack.

Generalization and perturbation. As mentioned earlier, generalization has a limited effect on mitigating our more sophisticated membership inference attack: as demonstrated in our study, even after applying the L2 regularization (with a coefficient of 0.01), a vulnerable record in MNIST dataset can still be attacked with a precision of 1 (Section 5.7). Adding noise to the training set or to the model to achieve differential privacy can suppress this information leak [8]. However, in the presence of high-dimensional data, which is particularly vulnerable to our attack, perturbation significantly undermines the utility of the model before its privacy risk can be effectively controlled [17]. So we believe that a practical solution could be to apply generalization and perturbation together with proper training set selection, detecting and removing those vulnerable training instances.

Training record selection. We believe that there is a fundamental contention between selecting useful training instances, which bring in additional information, and suppressing their unique influence to protect their privacy. An important step we could take here is to automatically identify outliers and drop those not contributing much to the utility of the model. To this end, new techniques need to be developed to balance risk and utility for those risky instances. A machine learning model could be built to automatically decide whether an instance should be included in the training set or not.

7. Related Work

Membership Inference Attacks. Membership inference attacks were first proposed by Homer et al. [13] in the context of Genome-Wide Association Studies (GWAS). They successfully inferred membership status from aggregated statistics. As a consequence of this attack, NIH has removed all aggregate data of GWAS from public websites [36]. Shokri et al. [30] proposed the first black-box MIA on machine learning models. Following this work, there have been extensive research on membership inference attacks. Salem et al. [28] showed that MIA could succeed with less shadow models and weaker assumptions on the adversary. Yeom et al. [35] formalized MIA under the framework of a distinguishing game and defined the adversary's advantage. Nasr et al. [25] proposed white-box attacks for both centralized and federated learning models. However, these attacks are all evaluated under an indiscriminate adversary that naively attacks all the records. In this paper, we consider a pragmatic adversary that carefully selects the target records. We demonstrate that this more sophisticated adversary could achieve high precision on models which the inference from indiscriminate adversaries is close to random guessing.

Disparate Vulnerability. Prior work show that both membership inference attacks and differential privacy have disparate affect on under-represented groups. Yaghini et al. [34] show that under-represented groups have higher privacy risk. Bagdasaryan et al. [6] demonstrate that differentially private stochastic gradient descent incurs a higher accuracy loss on under-represented classes and subgroups. In this work, we demonstrate that attackers can leverage disparate vulnerability to perform high-precision membership inference attacks on selected records in a well-generalized model.

Defenses against MIA. Different defenses against MIA have been proposed. Shokri et al. [30] and Salem et al. [28] have both showed that model generalization techniques, such as regularization, dropout, and model stacking, are effective in protecting against indiscriminate membership inference attacks. However, our work shows that a pragmatic adversary can still achieve high precision on selected records even after regularization. Recent work proposed using adversarial training [24] and adversarial examples [16] as defenses for membership inference attacks. Nasr et al. [24] added a privacy regularization term to the loss function of a model, and formulated the model training process as a min-max problem. Jia et al. [16] added perturbations to make the prediction vector an adversarial example of the attack classifier. However, both defenses assume the adversary to train an attack classifier on the target record's predictions, which is equivalent to an indiscriminate adversary based on direct inferences. Yet, this assumption does not hold for our pragmatic adversary with more sophisticated attack strategies. For example, neither the privacy term in [24] or the attack classifier in [16] could represent our strategies for target record selection and indirect inferences.

Privacy and Model Generalization. There is a connection between privacy and model generalization. Differential privacy can improve model generalization when data is reused for validation [8]. Moreover, the prior membership inference attack [30] achieves high precision on highly overfitted models while barely works on non-overfitted ones. Previous research also points out that privacy leakage can happen on non-overfitted models *when the adversary has control over the training algorithm*. Specifically, the adversary can encode private information of the training dataset into the predictions of well-generalized models [31]. These two attacks [30], [31] can be formalized under a uniform theoretical framework [35]. The risk of membership inferences can be empirically measured based on the influence of each training record [22].

Privacy-Preserving Machine Learning. Differential privacy [8] is a prominent way to formalize privacy against membership inference. It has been applied to various machine learning models including decision trees [14], logistic regression [37], and neural networks [4], [29]. However, there are no generic methods to achieve differential privacy for all useful machine learning models. More importantly, even if these methods are developed, their applications to real-world machine learning problems may significantly decrease the accuracy of the models, and thus will reduce their utility [5].

8. Conclusions

In this paper, we take a step forward to better understanding information leaks from machine learning models. In contrast to prior work, we consider a more pragmatic adversary who carefully selects targets and makes predictions conservatively. We demonstrate new membership inference attacks allowing such an adversary to identify vulnerable targets, and we deploy a novel methodology to evaluate the risk. Our results show that this new methodology better reflects the privacy risk of membership inference. In fact, it highlights cases where prior work underestimates the risk, achieving low attack precision (barely above the random-guessing baseline), whereas our pragmatic adversary still achieves high precision (at the cost of lower coverage). Specifically, our study reveals that a pragmatic adversary can achieve high precision (e.g., 95.05% on MNIST) in cases where prior work's methodology implies only barely above-the-baseline precision (i.e., 51.7%). In addition, our study highlights the conflict between selecting informative training instances and preventing their identification through their unique influences on the model, and points to the direction of using training data analysis and selection to complement existing approaches.

This work was supported in part by NSF CNS 13-30491 (THaW). The views expressed are those of the authors only.

References

[1] "Google cloud machine learning engine," https://cloud.google.com/ml-engine/reference/rest/.

[2] "Google prediction api," https://developers.google.com/ prediction/.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[4] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[5] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "Differential privacy: On the trade-off between utility and information leakage." *Formal Aspects in Security and Trust*, vol. 7140, pp. 39–54, 2011.

[6] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 15 479–15 488. [Online]. Available: http://papers.nips.cc/paper/9681-differential-privacy-has-disparate-impact-on-model-accuracy.pdf

[7] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[8] C. Dwork, "Differential privacy in the 40th international colloquium on automata," *Languages and Programming*, 2006.

[9] B. Efron, *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.

[10] C. Gentile and M. K. Warmuth, "Linear hinge loss and average margin," in *Advances in Neural Information Processing Systems*, 1999, pp. 225–231.

[11] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: evaluating privacy leakage of generative models using generative adversarial networks," *arXiv preprint arXiv:1705.07663*, 2017.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] 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, p. e1000167, 2008.

[14] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 2009, pp. 114–121.

[15] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[16] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019, pp. 259–274.

[17] A. Johnson and V. Shmatikov, "Privacy-preserving data exploration in genome-wide association studies," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1079–1087.

[18] J. T. Kost and M. P. McDermott, "Combining dependent p-values," *Statistics & Probability Letters*, vol. 60, no. 2, pp. 183–190, 2002.

[19] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 609–616.

[21] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[22] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *arXiv preprint arXiv:1712.09136*, 2017.

[23] N. Murata, S. Yoshizawa, and S.-i. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 865–872, 1994.

[24] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 634–646.

[25] ——, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, 2019, pp. 739–753. [Online]. Available: https://doi.org/10.1109/SP.2019.00065

[26] M. Phillips and B. M. Knoppers, "The discombobulation of de-identification," *Nature biotechnology*, vol. 34, no. 11, p. 1102, 2016.

[27] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," 2018.

[28] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/ml-leaks-model-and-data-independent-membership-inference-attacks-and-defenses-on-machine-learning-models/

[29] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1310–1321.

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

[31] 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*. ACM, 2017, pp. 587–601.

[32] T. B. Sprague, "Shape preserving piecewise cubic interpolation," 1990.

[33] R. Watrigant, M. Bougeret, and R. Giroudeau, "The k-sparsest subgraph problem," 2012.

[34] M. Yaghini, B. Kulynych, and C. Troncoso, "Disparate vulnerability: On the unfairness of privacy attacks against machine learning," *arXiv preprint arXiv:1906.00389*, 2019.

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

[36] E. A. Zerhouni and E. G. Nabel, "Protecting aggregate genomic data," *Science*, vol. 322, no. 5898, pp. 44–44, 2008.

[37] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.

[38] Y. Zhang, Z. L. Liu, and M. Song, "Chinet uncovers rewired transcription subnetworks in tolerant yeast for advanced biofuels conversion," *Nucleic acids research*, vol. 43, no. 9, pp. 4393–4407, 2015.