Privacy Risks of Explaining Machine Learning Models

Reza Shokri, Martin Strobel, Yair Zick
{reza,mstrobel,zick}@comp.nus.edu.sg
National University of Singapore

ABSTRACT

Can an adversary exploit model explanations to infer sensitive information about the models' training set? To investigate this question, we first focus on *membership inference attacks*: given a data point and a model explanation, the attacker's goal is to decide whether or not the point belongs to the training data. We study this problem for two popular transparency methods: gradient-based attribution methods and record-based influence measures. We develop membership inference attacks based on these model explanations, and extensively test them on a variety of datasets. For gradient-based methods, we show that the explanations can leak a significant amount of information about the individual data points in the training set, much beyond what is leaked through the predicted labels. We also show that record-based measures can be effectively, and even more significantly, exploited for membership inference attacks. More importantly, we design *reconstruction attacks* against this class of model explanations. We demonstrate that they can be exploited to recover significant parts of the training set. Finally, our results indicate that minorities and outliers are more vulnerable to these type of attacks than the rest of the population. Thus, there is a significant disparity for the privacy risks of model explanations across different groups.

ACM Reference format:
Reza Shokri, Martin Strobel, Yair Zick. 2019. Privacy Risks of Explaining Machine Learning Models. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 17 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine learning models are making increasingly high-stakes decisions in a variety of application domains, such as healthcare, finance and law [14, 17, 24]; driven by the need for higher prediction accuracy, decision-making models are becoming increasingly more complex, and as a result, much less understandable to various stakeholders. In other words, decision-making models are often 'black-boxes': we have no access to their inner workings, but only to their outputs.

Applying black-box AI decision makers in high-stakes domains is problematic: model designers face issues understanding and debugging their code, and adapting it to new application domains [22]; companies employing black-box models may expose themselves to various risks (e.g. systematically mis-classifying some subgroup of their client base [7], or facing the negative consequences of an automated decision [23]) ; finally, clients (i.e. those on whom decisions are made) are at risk of being misclassified, facing unwarranted automatic bias, or simply frustrated at their lack of agency in the decision-making process leading e.g. to a right to explanation in the European data privacy regulation GDPR [15].

This lack of transparency has resulted in mounting pressure from the general public, the media, and government agencies; several recent proposals advocate for the use of (automated) *transparency reports* (also known as model explanations in the literature) [15]. The machine learning (and greater CS) community has taken up the call, offering several novel explanation methods in the past few years (see Section 6). Transparency reports offer users some means of understanding the underlying model and its decision making processes[1]. By and large, they do so by offering users additional *insights*, or *information* about the model, with respect to the particular decisions it made about them (or, in some cases, about users like them).

Releasing additional information is a risky prospect from a privacy perspective; however, despite the widespread work on the design and implementation of model explanations, there has been little effort to address any privacy concerns that arise due to their release. This is where our work comes in. We begin our investigation by asking the following question.

> *Can an adversary leverage model explanations in order to infer private information about the training data of the underlying model?*

Our Contributions. We provide a comprehensive analysis of information leakage through feature-based and record-based model explanations. We identify what causes the leakage, and design inference algorithms to identify members of the training set. Our algorithms are able to reconstruct a large fraction of the training data. To the best of our knowledge, this paper is the first to analyze the privacy risks of model explanations for the training set of the underlying models.

We focus on two major attacks: *membership inference attacks* [33] to infer the presence of individual data points in the training set, and *reconstruction attacks* which aim at recovering the training data points. We analyze feature-based explanation algorithms, with an emphasis on gradient-based methods [36], and record-based algorithms, with an emphasis on methods that report influential data points [19].

We analyze what information gradient-based explanations can leak about the training data, and compare it what an attacker can extract from black-box decisions (which serve as a baseline). We also compare our approach to attacks that have access to the model's loss on data points, serving as the strongest membership inference attack (Section 5.3.1). We show that the **1-norm of the gradient is a considerable distinguisher between the members of the**

[1] See https://distill.pub/2018/building-blocks/ for a particularly intuitive and interactive explanation method for neural network architectures.

training set, and other data points from the same distribution. The reason is that as the training algorithm converges, the gradient decreases on all members of the training set, whereas for non-members the variance/1-norm can remain high.

Our experiments on synthetic datasets show that the dimensionality of the data has a large influence on the connection between 1-norm of the gradient and the membership inference accuracy (Section 4.4). This is related to the observation that very low dimensional data is generally more resistant to overfitting, whereas models on high dimensional data have a poorer performance on the test set. We also show that a variant of gradient based explanations, the SmoothGrad [38], which are designed to improve the performance of gradient-based model explanations, can reduce the information leakage by adding noise to the explained points (Section 4.5). This is achieved through substantially reducing the variance of the gradient's 1-norm, and making the distribution of this statistic more indistinguishable on members versus non-members. Although this is not a provably secure defense mechanism, it indicates that **it is possible to achieve both privacy and accuracy for model explanations**.

Record-based model explanations provide explain decisions by outputting the most influential data points in the training set for the decision on a particular point of interest. This presents an obvious leak of training data. In particular, membership inference attacks become simple as training points are frequently used to explain their own predictions (Section 5.1).

However, a simple attack of randomly querying the model results in a poor coverage in terms of reconstructing the training data. This is because a few certain training data records — especially outliers and mislabeled training points — have greater influence over most of the input space. Thus, after a few queries, the set of reconstructed data points converges, recovering no additional points.

We design an optimal algorithm that identifies and constructs regions of the input space where previously recovered points will not be influential(Section 5.2). This minimizes re-discovering previously revealed points, thus increasing the coverage of the algorithm. The algorithm is optimal in the sense that for worst case scenarios it recovers all possible points. Through empirical evaluations on data with different dimensionality, we show that **an attacker can reconstruct (almost) the entire dataset for high dimensional data**.

For datasets with low dimensionality, we demonstrate our heuristic's adaptability: using recovered points, one can recover up to 25% of the training set. As we show, the graph structure, induced by the influence function over the training set, tends to have a large strongly connected component and the **attacker is likely to recover at least all points in the large connected component**. Complementary, as unusual points tend to have a larger influence on the training process we show that **minorities have a high risk of being revealed**.

In the supplementary material, we further study the effectiveness of membership inference attacks based on additional feature-based explanations (including Integrated Gradients [40] and DeepLIFT [34]). These membership inference attacks achieve comparable, albeit weaker, success than gradient-based attacks. We also present the result of studying the influence of dataset size on the success of membership inference for influence based explanations.

2 PRELIMINARIES AND PROBLEM FORMULATION

Let us first describe some basic notation. We write vectors as $\vec{x}$. Given an integer m, we write $[m] = \{1, \ldots, m\}$. We are given a *dataset* $\mathcal{X} \subseteq \mathbb{R}^n$, labeled with *true data labels* given by $\ell : \mathbb{R}^n \to [k]$. It is assumed that $\mathcal{X}$ is sampled from a *target distribution*; one commonly used distribution simply samples a random subset of points from a given database, partitioning it into *training* and *test* data. The labeled dataset is used to train a *model* c, mapping each *datapoint* $\vec{x} \in \mathcal{X}$ — as well as other unobserved points in $\mathbb{R}^n$ — to a distribution over k *labels*; when $k = 2$ we often refer to the labels as ± 1, and to c as a *binary classifier*. The n coordinates of the data are referred to as *features*. While the model c outputs a distribution over labels — indicating its belief that a given label fits the datapoint $\vec{x}$ — it often reveals a single label to a user; this is simply the label deemed most likely to fit $\vec{x}$.

Families of models are often *parameterized*, with each possible model defined by a set of parameters θ taken from a *parameter space* Θ; for example, the family of logistic regression models is parameterized by a coefficient w_i for each feature and a bias b, thus $\Theta = \mathbb{R}^n \times \mathbb{R}$. We denote the model as a function of its parameters as c_θ. A model is trained to empirically minimize a *loss functions* on the training data. The loss function $L : \mathcal{X} \times \Theta \to \mathbb{R}$ takes as input the model parameters θ and a point $\vec{x}$, and outputs a real-valued loss $L(\vec{x}, \theta) \in \mathbb{R}$. Simple loss functions include the square loss for binary classification — $L_2(\vec{x}, \theta) \triangleq (c_\theta(\vec{x}) - \ell(\vec{x}))^2$ — or include additional regularization parameters over θ (see [32] for an overview).

The objective of a machine-learning algorithm is to identify an *empirical loss minimizer* over the parameter space Θ:

$$\hat{\theta} \in \operatorname{argmin}_{\theta \in \Theta} \frac{1}{|\mathcal{X}|} \sum_{\vec{x} \in \mathcal{X}} L(\vec{x}, \theta) \tag{1}$$

For many machine learning models (e.g. neural networks, logistic regression) it is infeasible to exactly obtain the global optimum point $\hat{\theta}$ in practice, and heuristics are used instead (e.g. stochastic gradient descent) to find the approximate optimal parameters. For these types of models we assume that we have a training algorithm A and say A induces a parameter $\hat{\theta}$ given a data set $\mathcal{X}$

$$A(\mathcal{X}) = \hat{\theta}. \tag{2}$$

3 MODEL EXPLANATIONS - AN OVERVIEW

In this section, we briefly overview some of the algorithms for explaining the machine learning models, notably the ones that we evaluate in this work.

Generally speaking, transparency reports explain model decisions on a given *point of interest* (POI) $\vec{y} \in \mathbb{R}^n$. An explanation ϕ takes as input the dataset $\mathcal{X}$, labels over $\mathcal{X}$ — given by either the true labels $\ell : \mathcal{X} \to [k]$ or by a trained model c — and a *point of interest* $\vec{y} \in \mathbb{R}^n$. In addition, explanation methods sometimes assume access to additional information, such as active access to model queries (e.g. [2, 12, 29]), a prior over the data distribution [6], or knowledge of the model class (e.g. that the model is a neural

network [4, 34, 40], or that we know the source code [11]). The output of an explanation function $\phi(\mathcal{X}, c, \vec{y}, \cdot)$ can be quite diverse; in this work we focus on two explanation paradigms: *record-based* explanations [19][2], and *numerical* influence measures. More formally, record-based explanations output a set of points $\phi(\mathcal{X}, c, \vec{y}, \cdot) \subseteq \mathcal{X}$, whereas feature-based numerical influence measures output a vector in $\mathbb{R}^n$, where $\phi_i(\mathcal{X}, c, \vec{y}, \cdot)$ corresponds to the importance of the i-th feature in determining the label of $\vec{y}$. In particular, we focus on gradient-based methods [36]. In what follows we often refer to the explanation of the POI $\vec{y}$ as $\phi(\vec{y})$, omitting its other inputs when they are clear from context.

3.1 Feature-based Model Explanations

Numerical explanations assign numerical values to individual features. In this case, the explanation $\phi(\vec{y})$ is a vector in $\mathbb{R}^n$, where $\phi_i(\vec{y})$ is the degree to which the i-th feature influences the label assigned to $\vec{y}$. Generally speaking, high values of $\phi_i(\vec{y})$ imply a greater degree of effect; negative values imply an effect for *other labels*; if $\phi_i(\vec{y})$ is close to 0, this normally implies that feature i was largely irrelevant in producing the label of $\vec{y}$.

Gradient-Based Explanations. Simonyan et al. [36] introduce gradient-based explanations (i.e. the change of the output of a model with respect to changes to the input) to visualize image classification models; the authors utilize the absolute value of the gradient rather than the gradient itself; however, outside image classification, it is reasonable to consider negative values, as we do in this work. We denote gradient-based explanations as ϕ_{GRAD}. Shrikumar et al. [35] propose $\vec{x} \circ \phi_{GRAD}(\vec{x})$ as a method to enhance numerical explanations (here, $\vec{x} \circ \vec{y}$ denotes the Hadamard product, which results in a vector whose i-th coordinate is $x_i \times y_i$). Note that since an adversary would have access to $\vec{x}$, releasing $\vec{x} \circ \phi_{GRAD}(\vec{x})$ is equivalent to releasing $\phi_{GRAD}(\vec{x})$.

Many feature-based explanation techniques are implemented in the INNVESTIGATE library[3] [3] which we use in our experiments; a discussion of these measures and the relations between them can also be found in [5].

3.2 Record-Based Model Explanations

The approach proposed by Koh and Liang [19] aims at identifying influential *datapoints*; that is, given a point of interest $\vec{y}$, find a subset of points from the training data $\phi(\vec{y}) \subseteq \mathcal{X}$ that explains the label $c_{\hat{\theta}}(\vec{y})$, where $\hat{\theta}$ is a parameterization induced by a training algorithm A as per Equation (2). Koh and Liang propose selecting a training point $\vec{x}_{\text{train}}$ by measuring the importance of $\vec{x}_{\text{train}}$ for determining the prediction for $\vec{y}$.

In order to estimate the effect of $\vec{x}_{\text{train}}$ on $\vec{y}$, Koh and Liang measure the difference in the loss function over $\vec{y}$ when the model is trained with and without $\vec{x}_{\text{train}}$. More formally,

$$\theta_{\vec{x}_{\text{train}}} \triangleq A(\mathcal{X} \setminus \{\vec{x}_{\text{train}}\}) \tag{3}$$

In other words, $\theta_{\vec{x}_{\text{train}}}$ is induced by training algorithm A given the dataset excluding $\vec{x}_{\text{train}}$. The influence of $\vec{x}_{\text{train}}$ on $\vec{y}$ is then

$$I_{\vec{y}}(\vec{x}_{\text{train}}) \triangleq L(\vec{y}, \theta_{\vec{x}_{\text{train}}}) - L(\vec{y}, \hat{\theta}). \tag{4}$$

A record-based explanation releases the k points with the highest absolute value of influence according to the above definition. In the case of ties we assume a lexicographic tie-breaking over $\mathcal{X}$. Additionally, it might release the influence of these k points (the values of $I_{\vec{y}}(\vec{z})$ as per Equation 4), which allows users to gauge their relative importance.

4 MEMBERSHIP INFERENCE ATTACKS USING GRADIENT-BASED EXPLANATIONS

This section considers membership inference attacks targeted at gradient-based explanations. The main concepts are based on the attack developed by Shokri et al. [33].

4.1 Threat Model

We assume that an attacker has access to a set S of points from the target distribution (i.e. the set S is sampled from the same distribution that we sample $\mathcal{X}$ from) and has *query access* to the model; these queries return either (a) the predicted label; (b) the prediction vector (i.e. the probability of each class); (c) the model explanation; or combinations thereof. The attacker knows whether the points in S are a part of the training data or not. Furthermore, the attacker knows the target model, and its training procedure. This threat model operates under similar assumptions to those made by Shokri et al. [33], and is somewhat weaker than the model studied by Milli et al. [25].

4.2 Description of Attack

The fundamental idea is to cast membership inference as a *learning problem*: the attacker trains an *attack model* that, given the output $c(\vec{x})$ of a *target model* can predict whether the point $\vec{x}$ was used during the training phase of c. The main drawback of this approach is that it assumes that the attacker has partial knowledge of the initial training set in order to train the attack model. Shokri et al. [33] circumvent this by training *shadow models* (i.e. models that mimic the behavior of c on the data), and demonstrate that comparable results can be obtained even when the attacker does not have access to parts of the initial training set. The focus of this paper is the information leakage caused by explanations. Therefore, we assume a best-case scenario where the attacker has membership information of some datapoints, and forgo the additional step of training shadow models.

Explanations try to achieve transparency and so release additional information about a model. However they might inadvertently also release information about training set membership in form of different behavior for members vs. non-members. For example when training with an s-shaped activation function (e.g. sigmoid or tanh) the training process pushes points towards the flat parts of the s. This is reflected in the gradient of training points as they are lower. A machine learning model can pick up on this and similar patterns.

The specific attack model, in our paper, is a neural network inspired by the architecture of Shokri et al. [33]. The network consists of multiple sub-networks (as shown in Figure 11 in Appendix A).

[2]Koh and Liang [19] refer to their explanations as *influence measures*, which the current authors found to be too generic. The authors thank Pang-Wei Koh for a fruitful email discussion on the topic.
[3]https://github.com/albermax/innvestigate

The sub-networks used in a given implementation depend on the type of information we assume the attacker has access to. The first sub-network uses the one-hot encoded predicted label as input, and has fully connected layers of size $[k, 512, 64]$. This sub-network is always part of the model, given that at minimum, we assume that the attacker knows the predicted label. If the attacker also knows the actual ground-truth label $\ell(\vec{x})$, a second sub-network with the same architecture as the first is added. If the attacker also has access to the *entire* prediction probability vector, a third network is added with fully connected layers of sizes $[k, 1024, 512, 64]$, the fourth sub-network takes the explanation as input (if available) and has the same architecture as the third (except for the input dimension). The final part of the network combines the previous four; it has layers of sizes $[256, 64, 1]$. We use ReLu activations between layers and initialize weights in a manner similar to Shokri et al. [33] to ensure a valid comparison between the methods.

Recently, Sablayrolles et al. [31] have shown that an attacker can create an optimal black-box membership inference attack based only on the loss of the target model (under some assumptions). An attacker that has access to the prediction vector c as well as the true label l can fairly easily recover the loss L of the model. This is true even if the attacker has no knowledge of the precise loss function used by the target, since most commonly used loss functions are highly correlated.

We compare our attack model against gradient-based model explanations to the strong attacker that has access to the model's loss. We train a simple threshold model on the attacker's training set; intuitively, we rely on the findings that points in the training set, overall, exhibit low loss. We emphasize that having access to the loss function is a relatively strong assumption. First, the black-box model might only output the most likely label, and not the whole prediction vector. Besides, in many scenarios the ground-truth label of a point is hard to obtain. Consider, for example, a model that predicts a patient's susceptibility to cancer, based on his hospital records, for which there is no true label. In fact, labels being costly to obtain is the entire reason we train ML models in the first place.

Different assumptions on the attacker access to model data give rise to five different scenarios. In all scenarios the attacker has access to the predicted label, but may additionally have access to: (a) the gradient ∇c (b) the prediction c (c) the gradient ∇c and the prediction c or (d) the gradient ∇c and the true label l. We ignore the potential fifth scenario where the attacker only has access to the predicted label: methods relying solely on the predicted label offer little in terms of effective attack avenues. We also ignore scenarios where the attacker has access to the prediction as well the ground-truth label: they are equivalent to having access to the loss for most standard loss functions.

4.3 Experimental Evaluation

In this section, we evaluate membership inference attacks relying on gradient-based explanations.

4.3.1 Target datasets and Models. An overview of the target datasets for our experiments is presented in Table 1 and a more extensive description is in Appendix B. For each dataset and each target model, we sub-sample 10,000 points each for training and testing.

Name	#Points	#Features	Type	#Classes
Purchase	197,324	600	Binary	100
Texas	67,330	6,170	Binary	100
Cifar-100	60,000	3,072	Pixel	100
Hospital	101,766	127	Mixed	2
Adult	48,842	24	Mixed	2

Table 1: Overview of the target datasets for membership inference

Where possible we use the same target model and training configuration as used in [33], all of which are fully connected multi-layer networks with tanh activations. The CIFAR-100 network has two convolution layers following the input layer. The Diabetic Hospital dataset was not used previously to study membership inference; we use the same model architecture as used for the UCI Adult dataset, as they are most comparable. We only defer from the original training procedure by changing the number of training epochs as a method of controlling for overfitting (i.e. early stopping).

4.3.2 Training for Different Levels of Overfitting. It is known (see e.g. [26]) that the degree of overfitting to the training data significantly affects the efficacy of membership inference attacks based on model predictions. Informally, a perfectly generalized model exhibits the exact same behavior on training and test points, and is impervious to membership inference attacks. To investigate how overfitting influences explanation-based attacks, we train models with different degrees of overfitting; this is measured as the difference between training and test accuracy, a standard measure of overfitting.

To ensure that different instances are as comparable as possible, we leave the target model architecture and training regime fixed, with the exception of the number of epochs we train the model. To achieve additional comparability between the different datasets, we train each model to achieve comparable degrees of overfitting between 0 (i.e. identical training and test accuracy) and 0.25 (i.e. the training accuracy is 25% higher than the test accuracy). For the Diabetic Hospital dataset, the degree of overfitting never exceeds 0.1; the Adult dataset exhibits negligible overfitting and is excluded from this part of the analysis. We hypothesize that the small dimensionality of the latter two datasets, as well as their binary prediction task, makes them less susceptible to overfitting.

Before we turn to our evaluation results, we wish to note a few points regarding the training regime. For a given dataset, overfitting strongly correlates with training accuracy and number of training epochs. Training accuracy steadily increases with the number of epochs, while the increase in test accuracy eventually plateaus. The accuracy and number of epochs needed to obtain a certain degree of overfitting gap differs widely between datasets. For the purchase dataset, the model reaches almost perfect training accuracy after 25 epochs, while the Texas dataset requires 200 epochs to achieve a training accuracy of 85%.

4.3.3 Results and Evaluation. Figure 1 shows the attack model accuracy for different targets. The main observations can be summarized as follows:

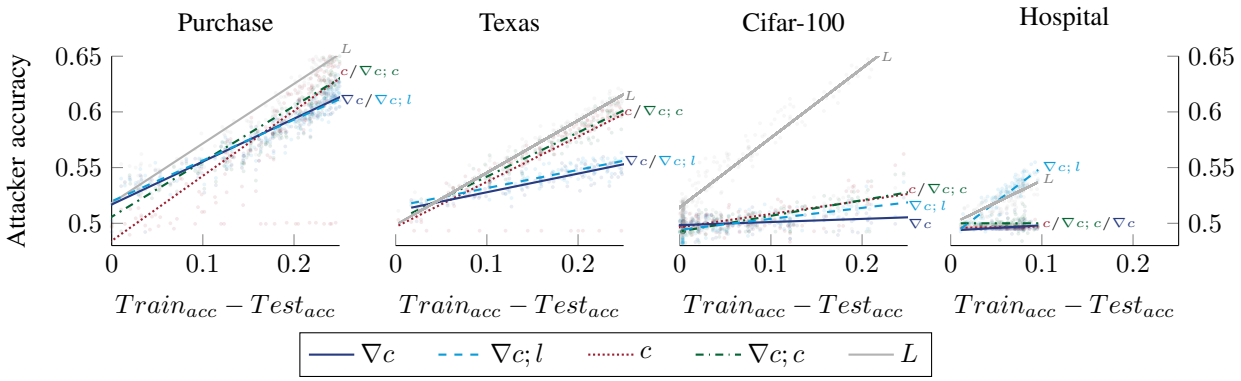


Figure 1: The results of the baseline attack model where the attacker uses a neural network for membership inference and has access to different types of information (a) the gradient ∇c (b) the gradient and the true label $\nabla c; l$ (c) the prediction vector c (d) the gradient and the prediction vector $\nabla c; c$ (e) or the Loss of the target model L.

Observation 4.1 (Overfitting)**.** Overfitting affects all attacks in a similar manner: overfitted models are more vulnerable, with vulnerability growing linearly in the degree of overfitting.

Observation 4.2 (Performance)**.** The type of information exploited by the attacker varies between datasets. Unsurprisingly, loss-based attacks offer the best performance. For the purchase dataset, all attacks behave in a similar manner, only slightly underperforming the loss-based attack. For the Texas dataset, gradient-based attacks underperform the prediction-based attack. For Cifar-100, all attacks perform relatively poorly, with access to *only* gradients offering the worst performance guarantee. For the hospital readmission datasets, the attack with access to the true label l and the gradient ∇c are the only attacks outperforming a random guess. Gradient-based attacks even beat the loss-based attack (this could also be an artifact of the linear approximation).

Observation 4.3 (Substitution)**.** Information gains from gradient access and prediction vector access are marginal: there is *no significant gain from the attacker having access to both the prediction vector and the gradient.*

Next, we analyze the factors that lead to performance differences, and discuss the roles different types of information play. We observe two main factors affecting our results: the structure of the target model, and the dimensionality of the target dataset.

Hypothesis 4.4. *When* tanh *(or, similarly sigmoid/softmax) is used as the activation function (the 1-norm of) the gradient is a proxy for the variance in the prediction vector.*

Variance in the prediction vector acts as a strong signal for membership: models make much more certain predictions on training points. Direct access to the prediction vector (as is the case in [33]), or indirect access to the same information via the gradient offers an avenue for the attacker. We verify this hypothesis in several ways.

(1) If the information leakage actually comes from the difference in $||\nabla c(\vec{x})||_1$, a simple model based only on the 1-norm achieves comparable results. This is in fact the case (see Figure 2). A decision tree as inference attack model trained only on the 1-norm achieves a competitive attack accuracy compared with a neural network trained on the entire gradient for the Purchase and Texas datasets. For Cifar-100 the decision tree *outperforms* the neural network attacker, though only by a small margin.
(2) If the model utilizes the information leaked from the difference in variance, then a model trained on normalized gradients would exhibit considerably worse performance. This is indeed the case: attacks trained on normalized gradients do not even outperform the random baseline.
(3) If the difference in 1-norm is connected to prediction variance, there should be a high correlation between the 1-norm of the gradient and prediction variance, as well as these two properties and membership in the training set. This is illustrated in Figure 3. For the Purchase dataset, there is a correlation between $Var(c(\vec{x}))$ and $||\nabla c(\vec{x})||_1$, and both correlate with training set membership. For the Texas dataset, the correlation between $Var(c(\vec{x}))$ and membership is negative. We suspect that this is due to high confidence (which corresponds to higher variance) on training points is not achieved (with only 85% training accuracy). Yet, there is still a clear correlation, which is a signal an attacker exploits. The results for Cifar are less clear-cut. There is some correlation between the variance/1-norm and test data membership; however, this correlation is inverted as overfitting increases.

Both number of features n, and the number of data labels k, have significant effect on the effectiveness of our attack model. We examine the effects of these parameters on synthetic datasets, for which we can control the values of n and k, in Section 4.4.

4.4 The Influence of the Input Dimension

The experiments in Section 4.3 indicate that $||\nabla c(\vec{x})||_1$ predicts training set membership. In other words, high absolute gradient values at a point $\vec{x}$ signal that $\vec{x}$ is *not* part of the training data: the classifier is uncertain about the label of $\vec{x}$, paving the way towards a potential attack; indeed, Shokri et al. [33] show how classifier uncertainty

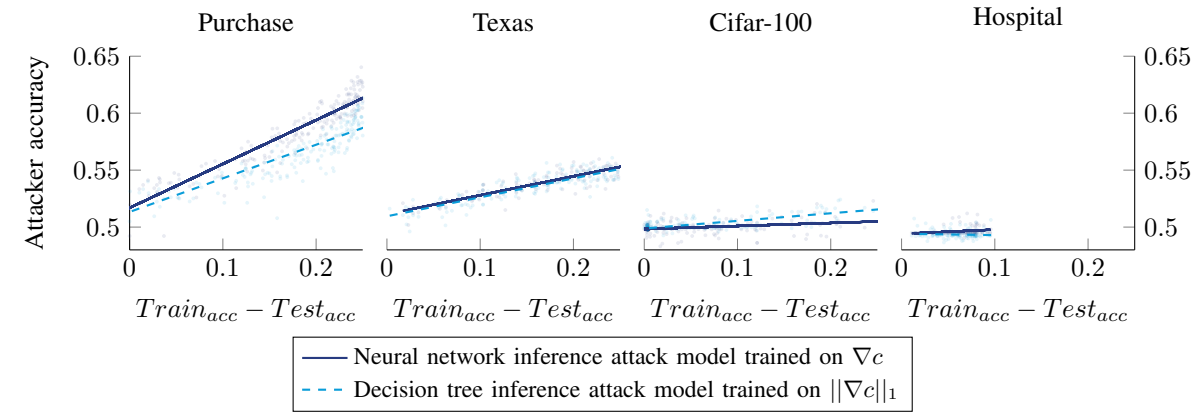


Figure 2: Attacks with a neural network trained on the entire gradient vs. a decision tree inference model using only the 1-norm of the gradient. While for the Purchase dataset the network slightly outperforms the tree, the results for the Texas dataset are almost identical. For Cifar-100, the decision tree model actually outperforms the neural network.

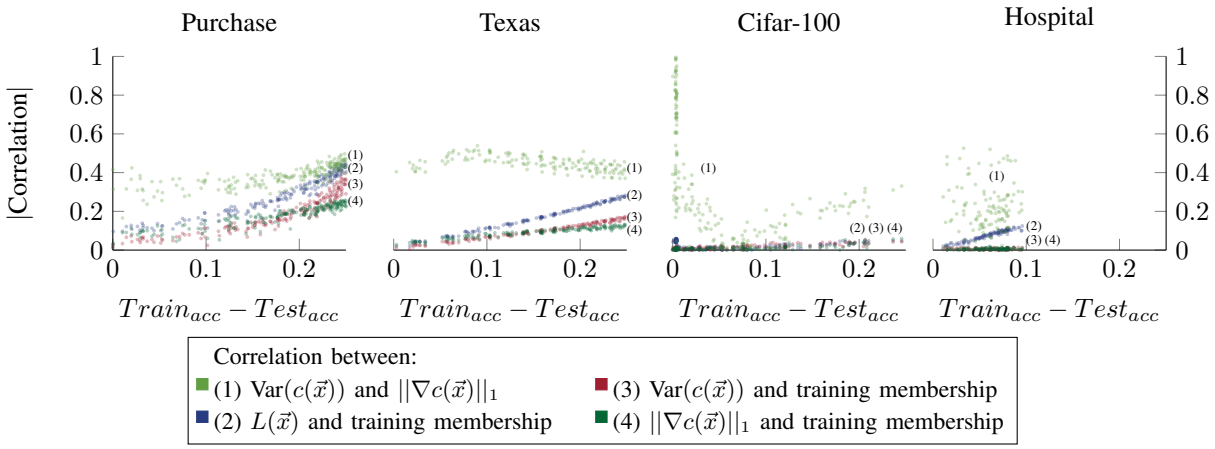


Figure 3: The Pearson correlation coefficient between training set membership and several key signals an attacker may observe. We display the absolute value for ease of comparison ($Var(c(\vec{x}))$ is negatively correlated with membership). Notably, even though for our attack $c(\vec{x})$ and $\nabla c(\vec{x})$ seem to be redundant, $Var(c(\vec{x}))$ and $||\nabla c(\vec{x})||_1$ are not strongly correlated.

can be exploited for membership attacks, further reinforcing this intuition. Let us next study this phenomenon on synthetic datasets, and the extent to which an adversary can exploit model gradient information in order to conduct membership inference attacks. The use of artificially generated datasets offers us control over the problem complexity, and helps identify important facets of information leaks.

To generate datasets, we use the Sklearn python library.[4] For n features, the function creates an n-dimensional hypercube, picks a vertex from the hypercube as center of each class, and samples points normally distributed around the centers. In our experiments, the number of classes is either 2 or 100 while the number of features is between 1 to 10,000 in the following steps,

$$n \in \{1, 2, 5, 10, 14, 20, 50, 100, 127, 200, 500, 600, 1000, 2000, 3072, 5000, 6000, 10000\}.$$

For each experiment, we sample 20,000 points and split them evenly into training and test set. We train a fully connected neural network with two hidden layers with fifty nodes each, the tanh activation function between the layers, and softmax as the final activation. The network is trained using Adagrad with learning rate of 0.01 and learning rate decay of 1e−7 for 100 epochs. Figure 13 in Appendix D contains an illustration of the generated data.

Increasing the number of features does not increase the complexity of the learning problem as long as the number of classes is fixed. However, the dimensionality of the hyper-plane increases, making its description more complex. Furthermore, for a fixed sample size, the dataset becomes increasingly sparser, potentially increasing

[4]the `make_classification` function https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html

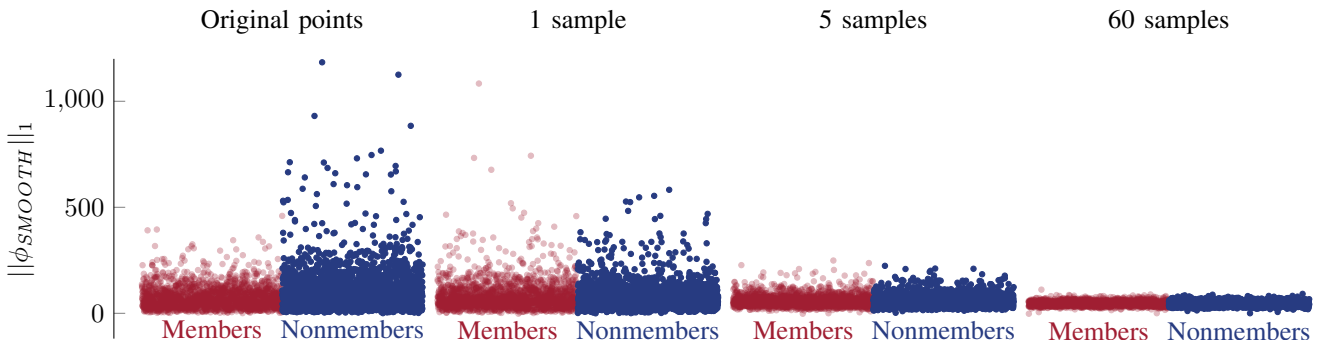


Figure 4: The 1-norm of the smooth gradient as a function of the number of samples used for smoothing. Each point in the plot above represents a single datapoint. One additional sample results in distributions close enough together as to make membership inference attacks fail. Only after a larger number of samples, all points have a low 1-norm.

the number of points close to a decision boundary. Increasing the number of classes increases the complexity of the learning problem.

Figure 5 shows the correlation between $||\nabla c(\vec{x})||_1$ and training membership. For datasets with a small number of features ($\leq 10^2$) there is almost no correlation. This corresponds to the failure of the attack for Adult and the Hospital dataset. When the number of features is in the range ($10^3 \sim 10^4$) there is a correlation, which starts to decrease when the data dimension is further increased. The number of classes seems to play only a minor role; however, a closer look at training and test accuracy reveals that the actual behavior is quite different. For two classes and a small number of features training and testing accuracy are both high (almost 100%), around $n = 10^2$ the testing accuracy starts to drop (the model over-fits) and at $n = 10^3$ the training accuracy starts to drop as well reducing the over-fitting. For 100 classes the testing accuracy is always low and only between $10^3 \leq n \leq 10^4$ the training accuracy is high, leading to over-fitting, just on a lower level. We also conduct experiments with networks of smaller/larger capacity, which have qualitatively similar behavior. However, the interval of n in which correlation exists and the amount of correlation varies (see Figure 14 in Appendix E).

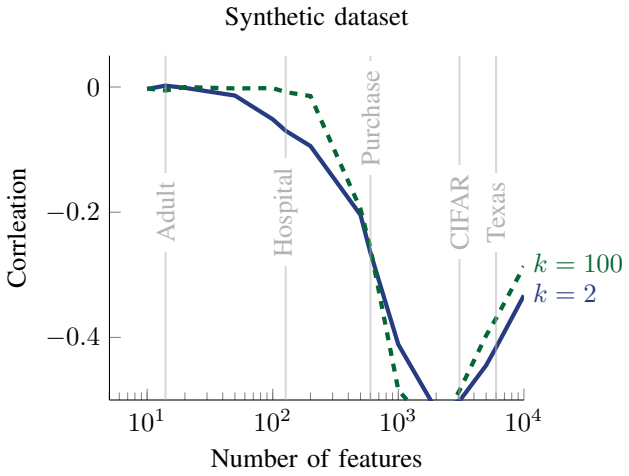


Figure 5: The correlation between $||\nabla c(\vec{x})||_1$ and training membership for synthetic datasets for increasing number of features n and different number of classes $k \in \{2, 100\}$

4.5 Protecting Against Gradient-Based Membership Inference Attacks: Differentially Private Models and Beyond

While the focus of this work is on attack models, let us briefly discuss potential defenses against gradient-based attacks. Differential private training [1] has been proposed as a provable defense against membership inference attacks based on model predictions. Indeed, differentially private models would be immune to gradient-based attacks: gradient-based explanations only interact with the model, and not with the underlying training set. However, currently differentially private training comes at a cost to model accuracy for large deep networks [1, 16] and training time; in addition, it may require a larger dataset [28].

Of all attribute based explanations considered in Appendix C, only the smoothed gradient is resistant to our attack models. Instead of releasing the gradient of a point, the smoothed gradient releases the average gradient of multiple samples, where the samples are created by adding Gaussian noise to the original point. As discussed above, the 1-norm of the gradient can be exploited as a signal for membership: non-members might have a high gradient. However, adding noise leads to a similar chance for members and nonmembers alike to have a high norm and the averaging brings down the overall norm (see Figure 4). This approach can be used to develop effective defenses while still releasing gradient-based explanations. However, future research is needed as we are aware that the failure of one attack is no proof of general privacy protection.

5 DATASET RECONSTRUCTION ATTACKS AGAINST RECORD-BASED MODEL EXPLANATIONS

As seen in Section 4.3.3, gradient-based explanations can be effectively used to conduct membership inference attacks. Let us next

now turn to record-based explanations. In this section we analyze how many training points an attacker can recover when having access to record based explanations. While technically the approaches discussed in previous sections could by applied to this setting as well, record based explanations allow for stronger attacks. Specifically, they explicitly reveal training points, and so offer the attacker certain knoweledge about points' training set membership (i.e. no false positives). Formally, we say a point $\vec{y}$ *reveals* point $\vec{x}$ if $\forall z \in \mathcal{X}$

$$|I_{\vec{y}}(\vec{x})| \geq |I_{\vec{y}}(\vec{z})|$$

In other words, $\vec{x}$ will be offered if one requests a record-based explanation for $\vec{y}$. Similarly, $\vec{y}$ *k-reveals* point $\vec{x}$ if there is a subset $S \subseteq \mathcal{X}, |S| = k-1$ such that $\forall z \in \mathcal{X} \backslash S$

$$|I_{\vec{y}}(\vec{x})| \geq |I_{\vec{y}}(\vec{z})|$$

In other words, $\vec{x}$ will be one of the points used to explain the prediction of $\vec{y}$ if one releases the top k most influential points. A point $\vec{x} \in \mathcal{X}$ that (k-)reveals itself, is called *(k-)self-revealing*.

5.1 Revealing membership for record based explanations

While for gradient-based explanations the attacker needs to rely on indirect information leakage to infer membership, for record based explanations, the attacker's task is relatively simple. Intuitively, a training point should be influential for its own prediction, so a record based explanation of a training set member is likely to contain the queried point, revealing membership. We test this hypothesis via experiments, i.e. for every point $\vec{x} \in \mathcal{X}$ in the training set we obtain the $k \in \{1, 5, 10\}$ most influential points for the prediction $f_\theta(\vec{x})$ and see if $\vec{x}$ is one of them.

5.1.1 Experimental setup. While the theoretical framework of influence functions described in Section 2 can be applied to an arbitrary classification task, it requires the training of as many classifiers as there are points in the training set in practice. Koh and Liang [20] propose an approximation method, but currently we only have access of its implementation for binary logistic regression models. However, for logistic regression, retraining the model (i.e. computing an *exact solution*) is actually faster than running the approximation algorithm. For larger models and datasets, these explanation methods seems intractable at the moment. This limits our experiments to the Adult and Diabetic Hospital datasets from our previous experiments for which we train binary logistic regression models. Furthermore, given the relatively long time it takes to compute all models, we reduce the size of the training set to 2,000 points, so we can run one experiment within a few hours (see Appendix G for an exploration of the effect of dataset size). Koh and Liang [20] use a specifically created dataset containing 2400 299 × 299-pixel dog and fish images, extracted from ImageNet [30]. The images were pre-processed by taking the output of the penultimate layer of a neural network trained on ImageNet[5]. These latent representations were then used to train a linear regression model. For more variety we include this dataset here as the *Fish vs. Dog* dataset. Arguably, this dataset doesn't contain particularly private information, but it can be seen as representative of image data in general. It also allows us to attack a pre-trained deep neural network which last layer was fine tuned. This type of transfer learning for small dataset becomes increasingly popular. Given the small size of this dataset we randomly split it into 1800 training and 600 test points for each experiment. For each dataset we repeat the experiment 10 times.

[5]Specifically, the authors used an Inceptionv3 architecture, for which pre-trained models are available for Keras https://keras.io/applications/#inceptionv3.

Figure 6: % of training points revealed as their own explanation, when $k \in \{1, 5, 10\}$ most influential points are revealed.

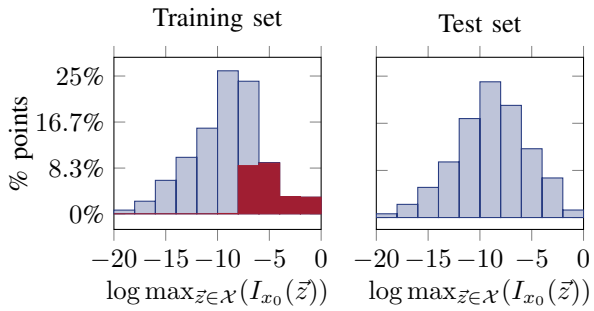


Figure 7: Histogram of the influence of the most influential points for every point in the training set(left) and test set (right) on a logarithmic scale (for one instance of Dog vs. Fish). The points in the training set for which the membership inference is successful (i.e. $\vec{x}_0 = \text{argmax}_{\vec{z}\in\mathcal{X}}(I_{x_0}(\vec{z}))$) are highlighted in red (dark).

5.1.2 Evaluation. Figure 6 shows the percentage of training points that would be revealed by explaining themselves. For the standard setting where the top 5 most influential points are revealed, a quarter of each dataset is revealed on average. For the hospital dataset, two thirds of training points are revealed. Even when just the most influential point would be released for the Adult dataset (which exhibits the lowest success rates), 10% of the members are revealed through this simple test.

As mentioned in Section 3.2, the influence score of the most influential points might be released to a user as well. In our experiments, the influence scores are similarly distributed between training and test points (i.e. members and non-members); however, the distribution is significantly different once we ignore the revealed training points. Figure 7 illustrates this for one instance of the Dog vs. Fish dataset; similar results hold for the other datasets. An attacker can exploit these differences, using techniques similar to those discussed in Section 4; however, we focus on other attack models in this work.

Minority and Outlier Vulnerability to Inference Attacks. Before turning our attention to dataset reconstruction attacks, we highlight an interesting finding. Visual inspection of datapoints for which membership attacks were successful indicates that outliers and minorities are more susceptible to being part of the explanation. Images of animals (a bear, a bird, a beaver) eating fish (and labeled as such) were consistently revealed (as well as a picture containing a fish as well as a (more prominent) dog that was labeled as fish).

	#points	$k=1$	$k=5$	$k=10$
Whole dataset	100%	16%	24%	28%
Birds	0.6%	64%	84%	86%
Clownfish	1%	14%	30%	33%
Lion fish	1%	9%	31%	43%

(a) Disclosure likelihood by type in the dog/fish dataset.

	% of data	$k=1$	$k=5$	$k=10$
Whole dataset	100%	33%	63%	75%
Age 0 -10	0.2%	61%	100%	100%
Age 10 -20	0.7%	23%	62%	95%
Caucasian	75%	32%	62%	74%
African American	19%	36%	66%	78%
Hispanics	2%	40%	61%	77%
Other race	1.5%	30%	60%	79%
Asian American	0.6%	33%	70%	95%

(b) Disclosure likelihood by age and race in the hospital dataset.

	% of data	$k=1$	$k=5$	$k=10$
Whole dataset	100%	10%	21%	28%
Age 10 -20	5%	0%	1%	1%
White	86%	10%	21%	28%
Black	10%	9%	16%	19%
Amer-Indian-Eskimo	1%	8%	23%	32%
Other	0.8%	16%	32%	43%
Asian-Pac-Islander	3.%	21%	40%	48%

(c) Disclosure likelihood by age and race in the adult dataset.

Table 2: Minority populations are more vulnerable to being revealed by the Koh and Liang method.

We label three "minorities" in the dataset to test the hypothesis that pictures of minorities are likelier to be revealed (Table 2a).

With the exception of $k = 1$ (for lion and clown fish), minorities are likelier to be revealed. While clownfish (which are fairly "standard" fish apart from their distinct coloration) exhibit minor differences from the general dataset, birds are more than three times as likely to be revealed. The hospital dataset exhibits similar trends (Table 2b). Young children, which are a small minority in the dataset, are revealed to a greater degree; ethnic minorities also exhibit slightly higher rates than Caucasians. This is mirrored (Table 2c) for the Adult dataset, with the exception of the age feature and the "Black" minority. Young people are actually particularly safe from inference. Note however, that young age is a highly predictive attribute for this classification. Only 3 out of the 2,510 entries aged younger than 20 have an income of more than 50K and none of them made it in any of our training set. Similarly only 12% of the Black people in the dataset (vs. 25% over all) have a positive label, making this attribute more predictive an easier to generalize.

While our findings are preliminary, they are quite troubling in the authors' opinion: transparency reports aim, in part, to *protect* minorities from algorithmic bias; however, data minorities are exposed to privacy risks through their implementation. Our findings can be explained by earlier observations that training set outliers are likelier to be "memorized" and thus less generalized [8]; however, such memorization leaves minority populations vulnerable to privacy risks.

5.2 Analysis For Dataset Reconstruction

Let us now turn our attention to a stronger mode of attack; rather than inferring the membership of specific datapoints, we try to recover the *entire training dataset*. We begin by describing our proposed attack models.

Given a parameter $\theta = (\vec{w}, b) \in \mathbb{R}^n \times \mathbb{R}$ a logistic regression model is defined as

$$f_\theta(\vec{x}) = \frac{1}{1 + e^{-\vec{w}^T \vec{x} + b}}.$$

For a point $\vec{y} \in R^n$ with label $l(\vec{y}) \in \{0, 1\}$ we define the loss of $f(\vec{y})_\theta$ as

$$L(\vec{y}, \theta) = (1 - f_\theta(\vec{y}))^{l(\vec{z})} (f_\theta(\vec{y}))^{1 - l(\vec{z})},$$

which corresponds to the standard maximal likelihood formulation. As logistic regression does not allow for a closed form optimum, it is generally trained via gradient ascent. We assume we are given a fixed training regime A (i.e. fixed parameters for number of steps, learning rate etc.).

Let f_θ be the model induced by $\mathcal{X}$ and

$$\mathcal{F}_\mathcal{X} = \{f_{\theta_{\vec{x}}} | f_{\theta_{\vec{x}}} \text{ is induced by } \mathcal{X} \backslash \{\vec{z}\}, \vec{x} \in \mathcal{X}\}$$

be the set of functions induced by omitting the training points. We can reformulate the influence of point $\vec{x}$ on point $\vec{y}$ (assuming $l(\vec{y}) = 0$) as

$$\begin{aligned} I_{\vec{y}}(\vec{x}) &= L(y, \theta) - L(y, \theta_{\vec{x}}) \\ &= f_\theta(y) - f_{\theta_{\vec{x}}}(y) \\ &= \frac{1}{1 + e^{-\vec{w}^T \vec{y} + b}} - \frac{1}{1 + e^{-\vec{w}_{\vec{x}}^T \vec{y} + b_{\vec{x}}}} \end{aligned}$$

The condition that $\vec{y}$ reveals $\vec{x}$ is thus equivalent to ensuring that for all $\vec{z} \in \mathcal{X}$, $|I_{\vec{y}}(\vec{x})| \geq |I_{\vec{y}}(\vec{z})|$. In the case of linear regression this simply implies that

$$\begin{aligned} &\left| \frac{1}{1 + e^{-\vec{w}^T \vec{y} + b}} - \frac{1}{1 + e^{-\vec{w}_{\vec{x}}^T \vec{y} + b_{\vec{x}}}} \right| \\ \geq &\left| \frac{1}{1 + e^{-\vec{w}^T \vec{y} + b}} - \frac{1}{1 + e^{-\vec{w}_{\vec{z}}^T \vec{y} + b_{\vec{z}}}} \right| \end{aligned}$$

which can be simplified to a linear constraint (whose exact form depends on whether the terms in absolute values are positive or negative).

Lower and upper bounds. It is relatively easy to construct examples in which only two points in the dataset can be revealed (see Figure 8). In fact, there are specific instances in which only a single datapoint can be revealed (see analysis in Appendix F), however these cases are neither particularly insightful nor do they reflect real-world settings. On the other side, there exists datasets where, independent of the number of features, the entire dataset can be recovered. The right side of Figure 8 illustrates such an example. The formal analysis is in Appendix F; actual datasets do not exhibit either extreme.

We present a practical algorithm with which the attacker can iteratively reveal new points. Algorithm 1 consists of two main

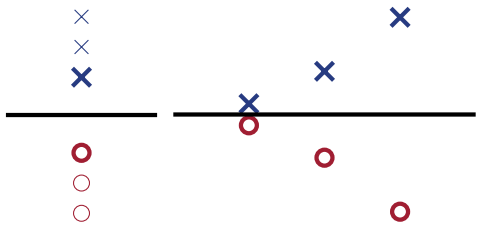


Figure 8: Illustrations of datasets for which only two (left) or all (right) points can be revealed under standard training procedures.

steps (1) Sample a point in the current subspace and (2) find a new subspace in which this point (as well as all already discovered points) has zero influence, and continue in this subspace.

Theorem 5.1 offers a lower bound on the number of points that Algorithm 1 discovers. The theorem holds under the mild assumption on the sampling procedure, namely that when sampling from a subspace R of dimension k, the probability of sampling a point within a given subspace Q of dimension $< k$ is 0. This assumption is fairly common, and holds for many distributions, e.g. uniform sampling procedures. We say that a set of vectors Q is d-wise linearly independent if every subset $D \subseteq Q$ of size d is linearly independent.

Theorem 5.1. *Given a training set $\mathcal{X} \subset \mathbb{R}^n$, let $\theta = (\vec{w}, b)$ be the model parameters induced by $\mathcal{X}$. Let $d \in \mathbb{N}$ be the largest number such that the vectors in $W = \{\vec{w}_{\vec{x}} | \vec{x} \in \mathcal{X}\}$ are d-wise linearly independent, then if all vectors in W are linearly independent of $\vec{w}$, Algorithm 1 (with query access to model predictions and record based explanations) reveals at least d points in $\mathcal{X}$ with probability 1.*

Proof. We need to show two statements (a) given an affine subspace R_q of dimension $n-q$ in which the influence of the first $q < d$ points is zero, with probability 1 a new point is revealed and (b) Algorithm 1 constructs a new subspace R_{q+1} in which the influence of the first $q+1$ points is zero, with probability 1.

For $q = 0$ $R_q = \mathbb{R}^n$ and statement (a) is trivially true: querying any point will reveal a new point. Now, when $q > 0$ and the attacker queries a point $\vec{y} \in R_q$, the only reason that no new point is revealed is that for all points $\vec{x} \in \mathcal{X}$ we have $I_{\vec{y}}(\vec{x}) = 0$. Note that $I_{\vec{y}}(\vec{x}) = 0$ iff $-\vec{w}^T\vec{y} + b = -\vec{w}_{\vec{x}}^T\vec{y} + b_{\vec{x}}$. Thus, if $I_{\vec{y}}(\vec{x}) = 0$ for all $\vec{x} \in \mathcal{X}$, then for all $\vec{x}_1, \vec{x}_2 \in \mathcal{X}$, $-\vec{w}_{\vec{x}_1}^T\vec{y} + b_{\vec{x}_1} = -\vec{w}_{\vec{x}_2}^T\vec{y} + b_{\vec{x}_2}$. By the d-wise independence assumption of $\vec{w}_i$ this can only happen in an $n-d$-dimensional subspace of $Z \subset \mathbb{R}^n$. The probability of sampling a point in the intersection of this subspace with R_q is 0 as $\dim(R_q \cap Z) \leq n - d < n - q = \dim(R_q)$.

To prove Statement (b) we first note that the attacker can recover θ (i.e. the original function) by querying f_θ for $n+1$ suitable — i.e., n of these points must be linearly independent, and the last one is different but otherwise arbitrary point — points and solving the corresponding linear equations. Knowing θ, the attacker can recover $|\theta_{\vec{x}}|$ for a vector $\vec{x} \in \mathcal{X}$, given the values of $|I_{\vec{y}_i}(\vec{x})|$ for $\vec{y}_i \in \mathbb{R}^n, i \in [n+1]$ suitable points. Similarly, the attacker can reconstruct the entire behavior of $|I_{\vec{y}_i}(\vec{x})|$ on an $n-q$ dimensional affine subspace R_q from $n-q+1$ suitable points.

Lines 9 to 18 in Algorithm 1 find these points. Since $|I_{\vec{y}}(\vec{x})|$ is continuous, the argmax in Line 7 is constant in a small region around $\vec{y}$, as long as there is no tie. Hence, ϵ will eventually become sufficiently small so that all points $\vec{y} + \epsilon v$ reveal the same point as $\vec{y}$. If there is a tie for the argmax at $\vec{y}$, let S be a open set of points around $\vec{y}$ such that $\forall \vec{y}' \in S\colon |\operatorname{argmax}_{\vec{z} \in \mathcal{X}}(|I_{\vec{y}}(\vec{z})|)| > 1$. Now either $\dim S = n - q$ in which case there exists an ϵ-ball B that is a subset of S around y such that all points in B reveal the same point as $\vec{y}$, or $\dim S < n - q$ in which case the probability of selecting S or any point in it is zero. So, with probability 1, Algorithm 1 finds $n-q+1$ suitable points that all reveal the same $\vec{x} \in \mathcal{X}$. From this the attacker can determine $|I_{\vec{y}}(\vec{x})|$ for all points $\vec{y} \in R_q$. In particular, the attacker can recover $\{\vec{y} \in R_q : |I_{\vec{y}}(\vec{x})| = 0\}$ which, again by the assumption of d-wise independence, is an $n-(q+1)$-dimensional affine subspace of R. □

Algorithm 1 recovers large parts of the dataset for high-dimensional data. If $d \geq |\mathcal{X}|$, it can recover the entire dataset. For $d \ll |\mathcal{X}|$, only a small part of the dataset will be recovered. Algorithm 1 is optimal in the sense that there are datasets for which only the points discovered by the algorithm can be discovered (see Figure 8 (Left)). However, there are many situations where the algorithm falls short of discovering all possible points (see Figure 8 (Right)). Furthermore, the algorithm does not exploit the fact that in practical situations the k most influential points would be revealed, instead of just one. In fact, incorporating the information about the k-most influential points into Algorithm 1 yields negligible benefits. Let us next discuss a heuristic that offers no theoretical guarantees, but work well even when $d \ll |\mathcal{X}|$, as demonstrated in our experiments.

5.2.1 Querying Revealed Points. This heuristic is relatively simple: use previously revealed points to reveal new points. When

Algorithm 1 Recovering the training set by minimizing the influence of already recovered points

```
1:  linearlyIndependent ← True
2:  q ← 0
3:  R_0 ← ℝ^n
4:  revealedVectors ← []
5:  while linearlyIndependent do
6:      y⃗ ← pickRandomPointIn(R_q)
7:      x⃗ ← argmax_{z⃗∈X}(|I_{y⃗}(z⃗)|)
8:      revealedVectors.append(x⃗)
9:      vectorsNotFound ← getBasisOf(R_q)
10:     spanningVectors ← [y⃗]
11:     while vectorsNotFound ≠ ∅ do
12:         for v⃗ ∈ vectorsNotFound do
13:             if x⃗ = argmax_{z⃗∈X}(|I_{y⃗+εv⃗}(z⃗)|) then
14:                 spanningVectors.append(y⃗ + εv⃗)
15:             end if
16:         end for
17:         ε ← ε/2
18:     end while
19:     if ∃r ∈ ℝ: ∀v⃗ ∈ spanningVectors: |I_{y⃗+εv⃗}(x⃗)| = r then
20:         linearlyIndependent ← False
21:     else
22:         R_{q+1} ← {y⃗ ∈ R_q | |I_{y⃗}(x⃗)| = 0}
23:         q ← q + 1
24:     end if
25: end while
26: return revealedVectors
```

querying a point the Koh and Liang measure returns k points from the training set (as well as their influence) as explanations. The attacker can now use these revealed points in new queries and obtain up to k new points in each of these queries. In fact, this naturally defines a structure over the training set, which we refer to as the *influence graph*. Every training datapoint is a node v in a graph G, with k outgoing edges towards the k nodes outputted by the Koh and Liang measure. The influence graph structure then governs the degree to which one can adaptively recover the training set. For example, if the graph contains only one strongly connected component, an attacker would be able to traverse (and thus recover) the entire training set from a single starting point. The following metrics are particularly informative:

Number of strongly connected components (SCCs): a high number of SCCs implies that the training set is harder to recover: an adaptive algorithm can only extract one SCC at a time. It also implies that the underlying prediction task is fragmented: the labels in one part of the dataset are independent from the rest.

Size of the SCCs: a small number of large SCCs help the attacker: they are more likely to be discovered, and recovering just some of them already results in recovery of significant portions of the training data.

Distribution of in-degrees: the greater a node's in-degree is, the likelier its recovery will be; for example, nodes with zero in-degree may be impossible for an attacker to recover. Generally speaking, a uniform distribution of in-degrees makes the graph easier to traverse.

Knowing these metrics helps to understand how successful an attacker can be, without any prior knowledge and a very limited number of queries.

As the attacker is traversing a graph, the order of queries does not affect the overall success of the attack, eventually every point reachable from the start will be revealed. There are, however, situations in which each query might be costly and an attacker tries to minimize the number of queries used. In order to benchmark the performance of our attacker, who has no knowledge of the influence graph structure, we compare it to an omniscient attacker who knows the graph structure and is able to optimally recover the SCCs; however, this problem is known to be NP-complete (the best known constant factor approximation factor is 92) [9]. We thus compare our approach to a greedy omniscient attacker, which selects the node that is connected to the most unknown points.

5.3 Dataset Reconstruction for Record-Based Explanations

We evaluate our reconstruction algorithms on the same datasets as the membership inference for record based explanations (see Section 5.1). We start with some general baselines.

5.3.1 Baseline Attack. A *baseline* attack model generates a static batch of transparency queries, i.e. new queries are not based on the attacker's past queries. An attacker who has some prior knowledge on the dataset structure can successfully recover significant chunks of the training data; in what follows, we consider three different scenarios.

Dataset	n	$\lvert\mathcal{X}\rvert$	# points recovered	% of $\lvert\mathcal{X}\rvert$
Dog vs. Fish	2048	1,800	1790	99.4
Adult	104	2,000	91.5	4.6
Diabetic Hospital	127	2,000	81.1	4.0

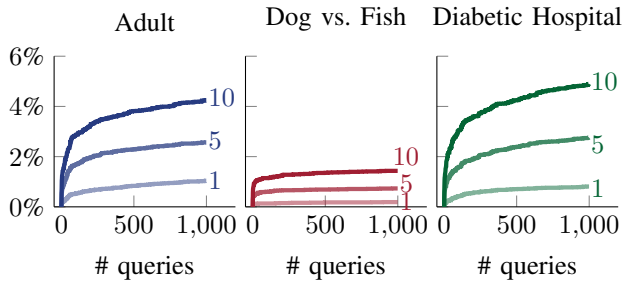
Table 3: The number of points recovered using our attack based on subspace reduction. For small sized, high dimensional data, like Dog vs. Fish, the attack can recover (almost) the entire dataset.

Uniform samples. With no prior knowledge on data distributions, an attacker samples points uniformly at random from the input space; this attack model is not particularly effective (Figure 9a): even after observing 1,000 queries with 10 training points revealed per transparency query, less than 2% of the Dog vs. Fish dataset and $\sim$ 3% of the hospital dataset are recovered. Moreover, the recovered images are unrepresentative of the data: since randomly sampled images tend to be white noise, the explanation images offered for them are those most resembling noise.

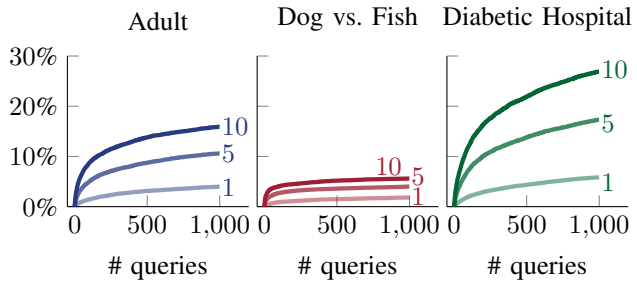
Marginal distributions. In a more powerful attack scenario, the attacker knows features' marginal distributions, but not the actual data distribution. In the case of images, the marginal distributions of individual pixels are rather uninformative; in fact, sampling images based on individual pixel marginals results in essentially random images. That said, under the Inception model, an attacker can sample points according to the marginal distribution of the latent space features: the weights for all nodes (except the last layer) are public knowledge; an attacker could reconstruct images using latent space sampling. Figure 9b shows results for the hospital dataset, and the Dog vs. Fish dataset under the inception model. This attack yields far better results than uniform sampling; however, after a small number of queries, the same points tended to be presented as explanations, exhausting the attacker's capacity to reveal additional information.

Actual distribution. This attack model offers access to the actual dataset distribution (we randomly sample points from the dataset that were not used in model training). This reflects scenarios where models make predictions on publicly available data. Using the actual data distribution, we can recover significant portions of the training data (Figure 9c). We again observe a saturation effect: additional queries do not improve the results significantly.

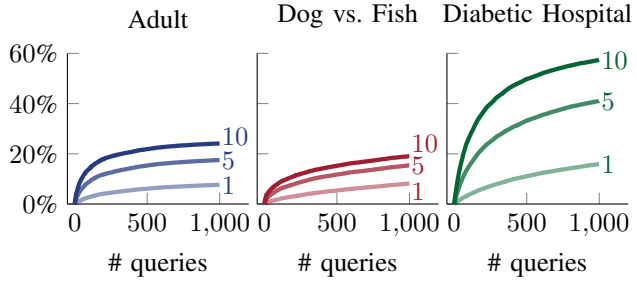
5.3.2 Attack based on subspace reduction. Table 3 summarizes the results from our attack based on Algorithm 1. For Dog vs. Fish (almost) the *entire* dataset can be recovered. The number of recovered points for Adult and Diabetic Hospital is small compared to the size of the dataset, but especially for Adult close to the dimensionality of the data, which is the upper bound for the algorithm. In our actual implementation, rather than constructing the subspaces and sampling points in them we recovered the entire weight vectors and sampled points by solving a least squares optimization problem with some additional constraints to increase numerical stability. These additional constraints bound the absolute values of the sampled points as well as ensuring that the value of the original function is not to close to 0 or 1.



(a) Uniform datapoint sampling.



(b) Marginal feature distribution sampling (for the Dog vs. Fish dataset points are sampled in the latent space).



(c) True point distribution sampling.

Figure 9: % of training data revealed by an attacker using different sampling techniques, with $k \in \{1, 5, 10\}$ explanation points revealed per query.

5.3.3 Adaptive heuristic attack. Table 4 summarizes some key characteristics of the influence graphs for the training datasets (we assume in this section that $k = 5$, slightly larger values of k lead to similar results). Remarkably, the influence graphs are fragmented, comprising of a large number of connected components, most of which have size 1. Each graph seems to have one larger SCC, which most of the smaller SCCs point to. This implies that an attacker starting at a random point in the graph is likely to recover only slightly more than the largest SCC. In fact, a significant amount of points (up to 75% for the Adult dataset) have in-degree 0: they cannot be recovered by this adaptive attack.

Figure 10 illustrates the percentage of the dataset recovered when traversing the influence graph from a random starting point. For the Dog vs. Fish dataset, the largest SCC is revealed every time. For the Adult and Diabetic Hospital datasets the largest SCC is not revealed for some points, resulting in limited success.

	Dog vs. Fish	Adult	Diabetic Hospital
#SCC	1709	1742	1448
#SCC of size 1	1679	1701	1333
Largest SCC size	50	167	228
Max in-degree	1069	391	287
#node in-degree=0	1364	1568	727

Table 4: Some key characteristics of the influence graphs induced by the record-based explanations (for $k = 5$). Averaged over 10 random initializations of the training set.

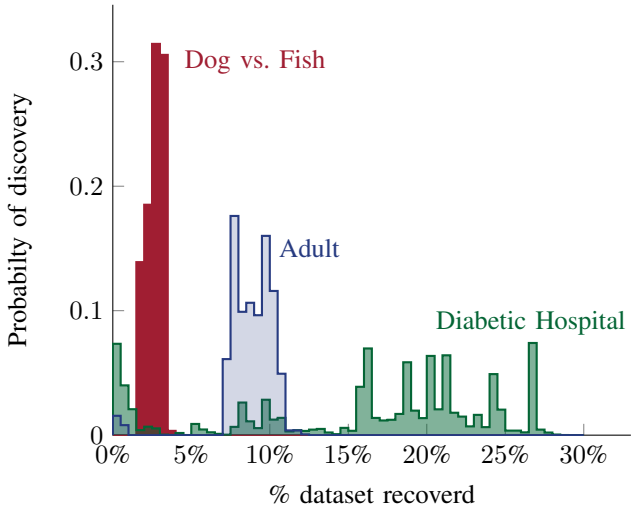


Figure 10: Distribution over the size of the training set that can be revealed when starting from an random point in the influence graph. Obtained by averaging over 10 initiations of the training set. Each time roughly the size of the largest SCC is recovered. For the Diabetic Hospital dataset the size of the SCC varies most, hence the multimodal distribution.

The number of queries required to traverse the discoverable part of the influence graph is invariant under changes to the query schedule (e.g. BFS, DFS, random walk and several heuristics based on the influence resulted in similar performance). Compared to an omniscient baseline (i.e. an attacker that knows the graph structure) and uses an greedy approach (see Section 5.2.1) to traverse the graph, our approaches require roughly twice as many queries.

6 RELATED WORK

Our work studies the vulnerability of transparency reports to membership inference attacks. We primarily focus on two types of transparency reports: record-based explanations, proposed by Koh and Liang [19], and feature-based explanations [6, 10, 12, 29, 37]. Datta et al. [12] show that their proposed measure, QII, is differentially private; however, similar guarantees have not been established for any of the other measures proposed in the literature. Indeed, in a recent paper, Milli et al. [25] show that gradient-based explanations can be used to reconstruct the underlying model; this serves as additional evidence of the vulnerability of transparency reports.

Ancona et al. [5] provide a recent overview of feature-based explanations (also called attribution methods). This approach can be divided into perturbation-based methods, which measure feature influence by altering (also removing or masking) the original input and measuring the difference in output, and backpropagation-based methods which rely on a small number of back-propagations through a model. Gradient-based methods fall within this category, with several proposed variants (see Appendix C). These variants tend to be noisier and harder to interpret.

In the category of perturbation-based methods fall occlusion based methods [42], LIME [29] which trains a local model, and QII [12] which computes the Shapley value of each feature. These methods rely on data sampling, making them comparatively slow and prone to querying counterfactuals (i.e. data points that could never actually occur). However, their explanations tend to be less noisy; moreover, their sampling techniques can be seen as a natural defense against privacy loss. This is an interesting direction for future study.

The attack scenario we adopt has been proposed by Shokri et al. [33]. Shokri et al. [33] use model predictions for data with known membership to train classifiers that accurately predict training set membership. However, Shokri et al. [33] assume access to the full probabilistic model prediction over the data points, as well as the true label; we assume a more realistic scenario, where one has access to the labels, and a given transparency report. The true label is unknown. Furthermore, some of our attacks models are rather simple, requiring only the 1-norm of the explanation as input.

Our analysis of record-based explanations indicates that outliers are more vulnerable to membership inference attacks than other datapoints, due to their distinctive characteristics. This is in line with existng works showing that overfitting may cause information leaks [41].

There exists some work on the defense against privacy leakage. Nasr et al. [26] use adversarial regularization, while Papernot et al. [27] and Papernot et al. [28] create a framework for differentially private training of machine learning models. However, these techniques are not yet widely adopted: their effect on model transparency is unknown.

7 CONCLUSIONS AND FUTURE WORK

In this work, we study membership inference attacks of transparent machine learning models based on two major types of model explanations. Both record and feature-based explanations can be successfully exploited to infer training set membership. For record-based explanations, we extract major parts of the training set via adaptive data queries. Our work is one of the first to show that releasing transparency reports can result in significant privacy risks; whatâĂŹs worse, for record based explanations minority populations face a significant risk of being exposed. While the authors are supportive of the call to algorithmic transparency, we strongly believe that it is the duty of the privacy community — indeed, the CS community at large — to ensure that policy makers and advocacy groups are aware of the risks and tradeoffs involved in offering greater model transparency.

Our results are just a first step towards a better understanding of transparency-based privacy attacks; several interesting open problems remain. For record based membership inference, it is unclear how one can offer safety guarantees. While low dimensional datasets are relatively robust against our subspace reduction attack, they are vulnerable to influence graph based attacks. While our theoretical analysis is restricted to logistic regression, the attacks can be extended to more complicated models. The general principle of minimizing the influence of discovered points allows for a model-agnostic gradient-descent basd attack. The influence graphs used in our analysis might be of separate interest: their characteristics highlight how a model "structures" the training set; their further study may lead to insights about the behavior of different model types, and provide an avenue for better record-based model explanations. Finally, designing safe transparency reports is an important research direction: one needs to release explanations that are both safe, and formally useful. For example, releasing no explanation (or random noise) is guaranteed to be safe, but is clearly not useful; record-based explanations are useful, but are not safe. Quantifying the quality/privacy tradeoff in model explanations will help us understand the capacity to which one can explain model decisions, while maintaining data integrity.

REFERENCES

[1] M Abadi, A Chu, I Goodfellow, H B McMahan, I Mironov, K Talwar, and L Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[2] P Adler, C Falk, S A Friedler, G Rybeck, C Scheidegger, B Smith, and S Venkatasubramanian. 2018. Auditing black-box models for indirect influence. In *Knowledge and Information Systems*.

[3] M Alber, S Lapuschkin, P Seegerer, M Hägele, K T Schütt, G Montavon, W Samek, K Müller, S Dähne, and P Kindermans. 2018. iNNvestigate neural networks! *arXiv preprint arXiv:1808.04260* (2018). arXiv:1808.04260 http://arxiv.org/abs/1808.04260

[4] M Ancona, E Ceolini, A Cengiz Öztireli, and M H Gross. 2017. A unified view of gradient-based attribution methods for Deep Neural Networks. *CoRR* (2017). arXiv:1711.06104

[5] M Ancona, E Ceolini, C Öztireli, and M Gross. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. arXiv:1711.06104 http://arxiv.org/abs/1711.06104

[6] D Baehrens, T Schroeter, S Harmeling, M Kawanabe, K Hansen, and K Mueller. 2009. How to Explain Individual Classification Decisions. *Journal of Machine Learning Research* 11 (2009), 1803–1831. arXiv:0912.1128 http://arxiv.org/abs/0912.1128

[7] J Buolamwini, T Gebru, and A Hubert. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classificatio. In *Proceedings of the 1st ACM Conference on Fairness, Accountability, and Transparency (ACM FAT*)*. 598–617.

[8] N Carlini, C Liu, J Kos, Ú Erlingsson, and D Song. 2018. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. *arXiv preprint arXiv:1802.08232* (2018). arXiv:1802.08232 http://arxiv.org/abs/1802.08232

[9] J Daligault and S Thomassé. 2009. On Finding Directed Trees with Many Leaves. In *Parameterized and Exact Computation*. Springer Berlin Heidelberg, 86–97.

[10] A Datta, A Datta, A D Procaccia, and Y Zick. 2015. Influence in Classification via Cooperative Game Theory. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*.

[11] A Datta, M Fredrikson, G Ko, P Mardziel, and S Sen. 2017. Use Privacy in Data-Driven Systems: Theory and Experiments with Machine Learnt Programs. In *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[12] A Datta, S Sen, and Y Zick. 2016. Algorithmic Transparency via Quantitative Input Influence. In *Proceedings of the 37th IEEE Conference on Security and Privacy (Oakland)*.

[13] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[14] C L Dunis, P W Middleton, A Karathanasopolous, and K Theofilatos. 2016. *Artificial Intelligence in Financial Markets: Cutting Edge Applications for Risk Management, Portfolio Optimization and Economics*. Springer.

[15] B Goodman and S R Flaxman. 2017. European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine* 38, 3 (2017), 50–57.

[16] B Jayaraman and D Evans. 2019. Evaluating Differentially Private Machine Learning in Practice. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security)*. 1895–1912.

[17] F Jiang, Y Jiang, H Zhi, Y Dong, H Li, S Ma, Y Wang, Q Dong, H Shen, and Y Wang. 2017. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology* 2, 4 (2017), 230–243.

[18] F Klauschen, K Müller, A Binder, G Montavon, W Samek, and S Bach. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *Plos One* (2015). https://doi.org/10.1371/journal.pone.0130140 arXiv:1606.04155

[19] P W Koh and P Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

[20] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. *arXiv preprint arXiv:1703.04730* (2017). arXiv:1703.04730 http://arxiv.org/abs/1703.04730

[21] A Krizhevsky, G Hinton, et al. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.

[22] C Lan and J Huan. 2017. Discriminatory Transfer. *arXiv preprint arXiv:1707.00780* (2017).

[23] St Lowry and G Macpherson. 1988. A blot on the profession. *British medical journal (Clinical research ed.)* 296, 6623 (1988), 657.

[24] L T McCarty. 2018. Finding the right balance in artificial intelligence and law. In *Research Handbook on the Law of Artificial Intelligence*. Edward Elgar Publishing.

[25] Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. 2018. Model Reconstruction from Model Explanations. *arXiv preprint arXiv:1807.05185* (2018). arXiv:1807.05185 http://arxiv.org/abs/1807.05185

[26] M Nasr, R Shokri, and A Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS)*. arXiv:1807.05852 http://arxiv.org/abs/1807.05852

[27] N Papernot, M Abadi, Ú Erlingsson, I Goodfellow, and K Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. arXiv:1610.05755 http://arxiv.org/abs/1610.05755

[28] Ni Papernot, S Song, I Mironov, A Raghunathan, K Talwar, and Úl Erlingsson. 2018. Scalable Private Learning with PATE. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. arXiv:1802.08908 http://arxiv.org/abs/1802.08908

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. https://doi.org/10.1145/1235 arXiv:1602.04938

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y arXiv:1409.0575

[31] A Sablayrolles, M Douze, Y Ollivier, C Schmid, and H Jégou. 2019. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

[32] S Shalev-Shwartz and S Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[33] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *Proceedings of the 38th IEEE Conference on Security and Privacy (Oakland)*. https://doi.org/10.1109/SP.2017.41 arXiv:1610.05820

[34] A Shrikumar, P Greenside, and A Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. arXiv:1704.02685

[35] A Shrikumar, P Greenside, and A Kundaje. 2017. Not just a black box: Learning Important Features Through Propagating Activation Differences. *arXiv preprint arXiv:1605.01713* (2017). https://doi.org/10.1109/IALP.2010.4 arXiv:1704.02685

[36] K Simonyan, A Vedaldi, and A Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034* (2013). arXiv:1312.6034 http://arxiv.org/abs/1312.6034

[37] J Sliwinski, M Strobel, and Y Zick. 2019. Axiomatic Characterization of Data-Driven Influence Measures for Classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*. arXiv:1708.02153 http://arxiv.org/abs/1708.02153

[38] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg. 2017. SmoothGrad : removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017). arXiv:arXiv:1706.03825v1

[39] B. Strack, J. P. Deshazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. 2014. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International* (2014). https://doi.org/10.1155/2014/781670

[40] M Sundararajan, A Taly, and Q Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

[41] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2017. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *arXiv preprint arXiv:1709.01604* (2017). arXiv:1709.01604 http://arxiv.org/abs/1709.01604

[42] M D Zeiler and R Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*.

A MEMBERSHIP INFERENCE ATTACK NETWORK STRUCTURE

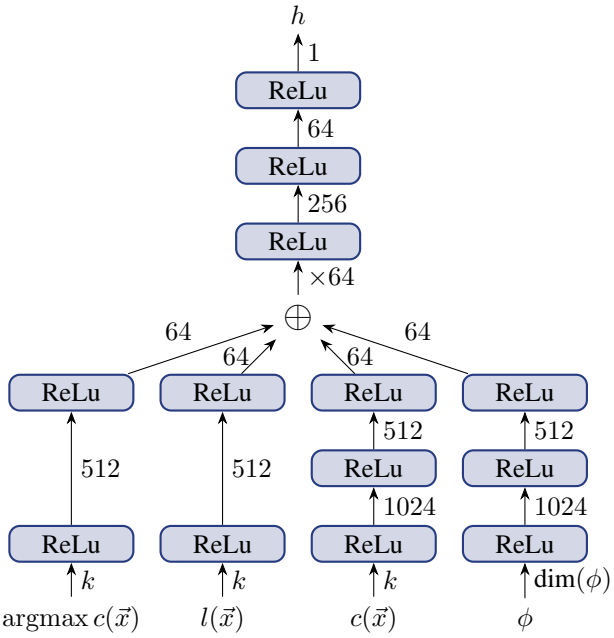


Figure 11: The design of the neural network attack model.

B TARGET DATASETS

B.1 Purchase dataset

The dataset originated from the "Acquire Valued Shoppers Challenge" on Kaggle[6]. The goal of the challenge was to use customer shopping history to predict shopper responses to offers and discounts. For the original membership inference attack, Shokri et al. [33] create a simplified and processed dataset, which we use as well. Each of the 197,324 records corresponds to a customer. The dataset has 600 binary features representing customer shopping behavior. The prediction task is to assign customers to one of 100 given groups (the labels). This learning task is rather challenging, as it is a multi-class learning problem with a large number of labels; moreover, due to the relatively high dimension of the label space, allowing an attacker access to the prediction vector — as is the case in [33] — represents significant access to information.

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

B.2 Texas hospital stays

The Texas Department of State Health Services released Hospital Discharge Data public use files spanning from 2006 to 2009.[7] The data is about inpatient status at various health facilities. There are four different groups of attributes in each record: general information (e.g. hospital id, length of stay, gender, age, race), the diagnosis, the procedures the patient underwent and the external causes of injury. The goal of the classification model is to predict the patientâĂŹs primary procedures based on the remaining attributes (excluding the secondary procedures). The dataset is filtered to include only the 100 most common procedures. The features are transformed to be binary resulting in 6,170 features and 67,330 records.

B.3 CIFAR-100

CIFAR-100 is a well known benchmark dataset for image classification [21]. It consists of 100 classes of $32 \times 32 \times 3$ color images, with 600 images per class. The dataset is usually split in 50,000 training and 10,000 test images. To enable multiple different experiments we reshuffle these two sets before sampling.

B.4 UCI Adult (Census income)

this dataset is extracted from the 1994 US Census database [13]. It contains 48,842 datapoints and based on 14 features (e.g. age, workclass, education) the goal is to predict if the yearly income of a person is above 50,000 $. We transform the categorical features into binary form resulting in 104 features.

B.5 Diabetic Hospital

The dataset contains data on diabetic patients from 130 US hospitals and integrated delivery networks [39]. We use the modified version described in [20] where each patient has 127 features which are demographic (e.g. gender, race, age), administrative (e.g., length of stay) and medical (e.g., test results); the prediction task is readmission within 30 days (binary). The dataset contains 101 766 records from which we sub-sample balanced datasets (i.e. with equal numbers of patients from each class).

C EXPERIMENTS FOR OTHER ATTRIBUTION BASED METHODS

Besides the gradient, several other explanation methods based on the gradient and or back propagation have been proposed. We conducted the attack described in Section 4 replacing the gradient with some other popular of these explanation methods. The techniques below are all implemented in the INNVESTIGATE library[8] [3]. A in depth discussion of some of these measures and the relations between them can also be found in [5].

Integrated Gradients. Sundararajan et al. [40] argue that instead of focusing on the gradient it is better to compute the average gradient on a linear path to a baseline $\vec{x}_{BL}$ (often $\vec{x}_{BL} = \vec{0}$). This approach satisfies three desirable axioms: sensitivity, implementation invariance and a form of completeness. Sensitivity means that given a point $\vec{x} \in \mathcal{X}$ such that $x_i \neq x_{BL,i}$ and $c(\vec{x}) \neq c(\vec{x}_{BL})$, then $\phi_i(\vec{x}) \neq 0$; completeness means that $\sum_{i=1}^{n} \phi_i(\vec{x}) = c(\vec{x}) - c(\vec{x}_{BL})$. Mathematically the explanation can be formulated as

$$\phi_{INTGRAD}(\vec{x})_i \triangleq (x_i - \vec{x}_{BL,i}) \cdot \int_{\alpha=0}^{1} \frac{\partial c(\vec{x}^{\alpha})}{\partial \vec{x}_i^{\alpha}} \Bigg|_{\vec{x}^{\alpha}=\vec{x}+\alpha(\vec{x}-\vec{x}_{BL})}.$$

Layer-wise Relevance Propagation (LRP). Klauschen et al. [18] use backpropagation to map *relevance* back from the output layer to the input features. Let l be a layer in the network and the number of layers be denoted by L. Then the relevance $r_i^{(l)}$ of the i-th neuron in the l-th layer can be computed as:

$$r_i^{(L)}(\vec{x}) \triangleq c_i(\vec{x})$$
$$r_i^{(l)}(\vec{x}) \triangleq \sum_j \frac{z_{ji}}{\sum_{i'}(z_{ji'} + b_j) + \epsilon \cdot \text{sign}(\sum_{i'}(z_{ji'} + b_j))} r_j^{(l+1)}$$

Here z_{ji} is the weighted activation of neuron i to neuron j in the next layer and b_j is the bias added to neuron j. The summations are over all neurons in the respective layers. Finally, the ϵ is added to avoid numerical instabilities. In words, LRP defines the relvance in the last layer as the output itself and in each previous layer the relevance is redistributed according to the weighted contribution of the neurons in the previous layer to the neurons in the current layer. The final attributions for the input $\vec{x}$ are defined as the attributions of the input layer: $\phi_{LRP}(\vec{x})_i \triangleq r_i^{(1)}(\vec{x})$.

DeepLift. The method proposed by Shrikumar et al. [35] combines the two main ideas in previous methods. Like LRP, it propagates attribution backwards through the network; like integrated gradients, it uses a baseline reference point $\vec{x}_{BL}$. Analogous to the weighted activations z_{ji} for the point $\vec{x}$ during a forward pass the weighted activations $\bar{z}_{ji}$ for the reference point $\vec{x}_{BL}$ are calculated. The attribution of neuron i in layer l is recursively defined as

$$\bar{r}_i^{(L)}(\vec{x}) \triangleq c_i(\vec{x}) - c_i(\vec{x}_{BL})$$
$$\bar{r}_i^{(l)}(\vec{x}) \triangleq \sum_j \frac{z_{ji} - \bar{z}_{ji}}{\sum_{i'} z_{ji'} - \sum_{i'} \bar{z}_{ji'}} \bar{r}_j^{(l+1)}$$

The measure is defined as the attribution on the input layer

$$\phi_{DEEPLIFT}(\vec{x})_i \triangleq \bar{r}_i^{(1)}(\vec{x}).$$

DeepLift with the recursion as defined above satisfies completenss by design; the recursion is referred to as the "Rescale Rule". A different version called "Reveal-Cancel" [35] is not considered in this work.

Smooth gradient. Smilkov et al. [38] introduced SmoothGrad to sharpen the images created when using the gradient as an explanation in image classification tasks. The basic idea is to average several gradients which are sampled around the point of interest, for the sampling Gaussian noise is added to the input. For a given variance σ and number of samples k the SmoothGrad is defined as

$$\phi_{SMOOTH}(\vec{x}) \triangleq \frac{1}{k} \sum_{1}^{k} \phi_{GRAD}(\vec{x} + \mathcal{N}(k, \sigma)).$$

Figure 12 shows the attack accuracy on the purchase dataset for the different explanation methods. While the performance using $\phi_{INTGRAD}$ is very similar to ϕ_{GRAD} the attacj performs worse for LRP and Deeplift. In fact these two methods are further away from

[7] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm
[8] https://github.com/albermax/innvestigate

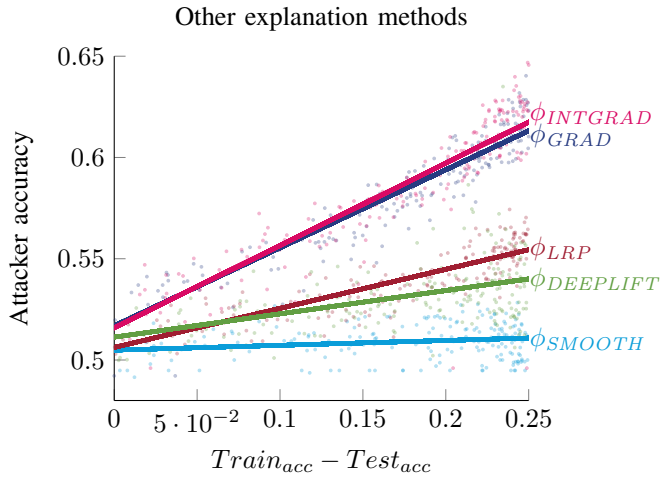


Figure 12: A comparison of the accuracy of the membership inference attack on the purchase dataset for different explanation methods the attacker might exploit.

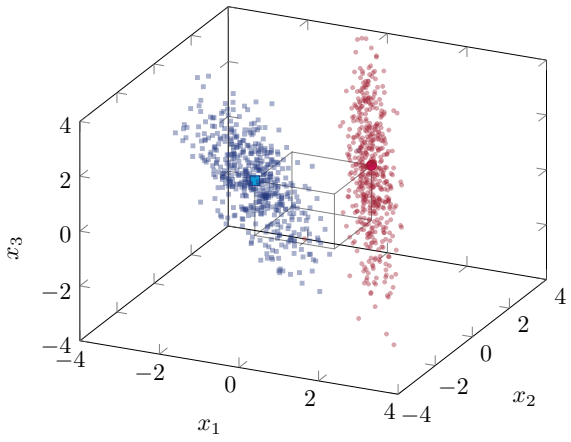


Figure 13: An illustration of the dataset generation process with $n = 3$.

the original gradient and it is less clear what is the exact signal leakage here. The attack fails for SmoothGrad. In fact the sampling used to generate this explanation mimics the practice of adding noise to obtain differential privacy, this can be seen as a natural defense mechanism against the attack. A precise analysis might be interesting future work.

D GENERATION OF SYNTHETIC DATASETS

To generate datasets, we use the `make_classification` function of the Sklearn python library. For n features, the function creates a n-dimensional hypercube, picks a vertex from the hyper-cube as center of each class, and samples points normally distributed around the centers. See Figure 13 for an illustration of the $n = 3$ case.

E VARYING THE CAPACITY OF THE NEURAL NETWORK FOR SYNTHETIC DATASETS

Figure 14 illustrates how the correlation between $||\nabla c(\vec{x})||_1$ and training membership is influenced by the capacity of the target network.

F BOUNDS ON NUMBER OF REVEALABLE POINTS

The following Lemma characterizes the situations in which only a single point of the dataset can be revealed for $X \subseteq \mathbb{R}$. The conditions for higher dimensions follow from this.

Lemma F.1. *Given a training set $\mathcal{X}$ let f_θ and $\mathcal{F}_\mathcal{X}$ be induced by $\mathcal{X}$ with $|\mathcal{F}_\mathcal{X}| \geq 2$, then one of the following statements is true*

(1) $\forall \vec{x} \in \mathcal{X} : w = w_{\vec{x}}$ *and* $(b \geq b_{\vec{x}}, \forall x \in \mathcal{X}) \vee (b \leq b_{\vec{x}}, \forall x \in \mathcal{X})$ *(i.e all functions in $\mathcal{F}_\mathcal{X}$ are shifted in one direction of f_θ),*

(2) $\exists \vec{y} \in \mathbb{R}^n : \forall \vec{x} \in \mathcal{X} : w_{\vec{x}}\vec{y} + b_{\vec{x}} = w\vec{y} + b$ *(i.e all functions in $\mathcal{F}_\mathcal{X}$ intersect with in f_θ the same point) ,*

(3) $\forall \vec{x} \in \mathcal{X} : w = w_{\vec{x}}$ *and there exists a numbering of the points in $\mathcal{X}$ such that $b_1 \leq b_2 \leq \cdots \leq b_k \leq b \leq b_{k+1}, \ldots, b_m$ such that $b \leq \log(\frac{1}{2}(e^{b_1} + e^{b_m}))$.*

(4) at least two points can be revealed.

Proof. It is easy to see that in the first two situations only one point can be revealed (the one corresponding to the largest shift or largest angle at intersection). In the third case all functions in $\mathcal{F}_\mathcal{X}$ are shifts of f_θ, but not all in the same direction. Only, the left most and right most shift are candidates for being revealed as they clearly dominate all other shifts. Also we assume $b_1 < b < b_m$ (as soon as one shift coincidences with the original model the statement is trivially true). Some calculus reveals the condition for which the two points would have the same influence is

$$y = \ln\left(\frac{-2e^b + e^{b_1} + e^{b_m}}{e^{b+b_1} + e^{b+b_m} - 2e^{b_1+b_m}}\right)/w$$

, which is well defined when the expression inside the logarithm is positive and $e^{b+b_1} + e^{b+b_m} - 2e^{b_1+b_m} \neq 0$. The former is the case for $b < \log(\frac{1}{2}(e^{b_1} + e^{b_2}))$, which also ensures the latter condition. On the other hand if $b \leq \log(\frac{1}{2}(e^{b_1} + e^{b_m}))$ the equation has no solution and so only one point can be revealed.

It remains to show that in all other cases at least two points can be revealed. Let's assume that there is a single $\vec{x} \in \mathcal{X} : \vec{w} \neq \vec{w}_{\vec{x}}$. In this case $w_{\vec{x}}^T\vec{y} + b_{\vec{x}} = w^T\vec{y} + b$ can be solved and at the solutions $I(\vec{x}) = 0$ yet all other points have nonzero influence and one of them is revealed. Yet, since $w_{\vec{x}'}^T\vec{y} + b_{\vec{x}} - w^T\vec{y} + b$ is constant for all $\vec{x}' \in \mathcal{X}, \vec{x} \neq \vec{x}$ and $|w_{\vec{x}}^T\vec{y} + b_{\vec{x}} - w^T\vec{y} + b|$ can take arbitrary values, there exists $\vec{y} \in R^n$ such that $\vec{x} = \text{argmax}_{\vec{z} \in \mathcal{X}} |I_{\vec{y}}(\vec{z})|$. Finally, if there are multiple points with $\vec{x} \in \mathcal{X} : \vec{w} \neq \vec{w}_{\vec{x}}$ none of them can be revealed for all $\vec{y}$ as long as the condition in 2) is not satisfied. □

The following result states that there exist models and datasets for which every point can be revealed.

Lemma F.2. *For every $m \in \mathbb{N}$ there exists a dataset $\mathcal{X} \subset \mathbb{R}^n$ with $|\mathcal{X}| = m$ and a training procedure A so that any point in $\mathcal{X}$ can be revealed by record-based explanations.*

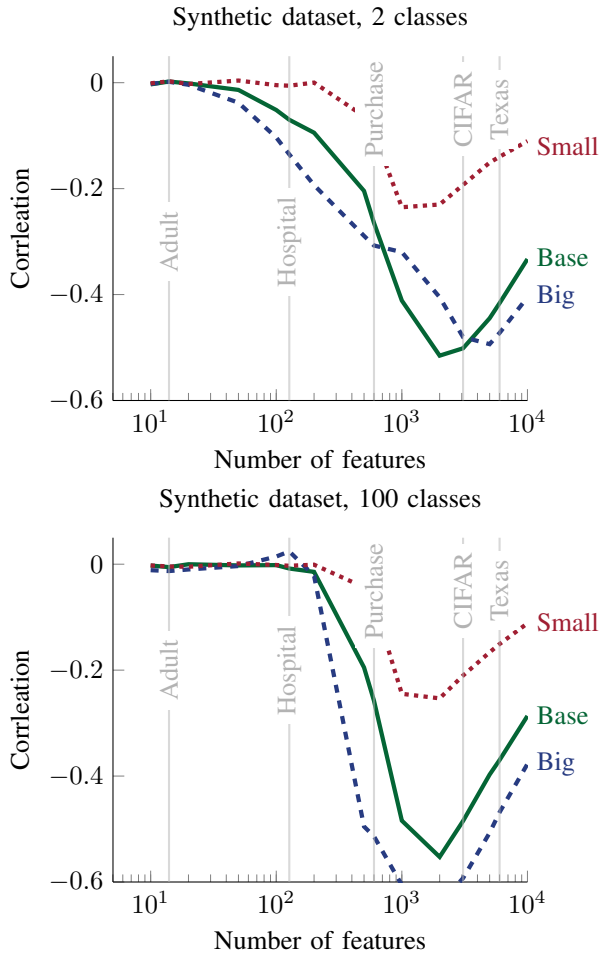


Figure 14: The correlation between $||\nabla c(\vec{x})||_1$ and training membership for synthetic datasets for increasing number of features n and different number of classes $k \in |2, 100$ for three different networks. The "Small" has one hidden layer with 5 nodes, "Base" has two layers with 50 nodes each, "Big" has 3 layers with 100 nodes each.

Proof. Given that we do not have restrictions on our training procedure A the claim is equivalent to the existence of a parameter θ and a series of parameters θ_k and points y_k such that $\forall k, m \in \mathbb{N}$:

$$|L(\vec{y}_k, \theta) - L(\vec{y}_k, \theta_k)| = \max_{i \in [m]} |L(\vec{y}_k, \theta) - L(\vec{y}_k, \theta_i)|$$

W.l.o.g. $n = 1$, let $w = b = 0$ and $\forall k\colon l(y_k) = 0$, then

$$|L(\vec{y}_k, \theta) - L(\vec{y}_k, \theta_k)| = \max_{i \in [m]} |L(\vec{y}_k, \theta) - L(\vec{y}_k, \theta_i)|$$
$$\Leftrightarrow L(\vec{y}_k, \theta_k) = \max_{i \in [m]} |L(\vec{y}_k, \theta) - L(\vec{y}_k, \theta_i)$$
$$\Leftrightarrow f_{\theta_k}(\vec{y}_k) = \max_{i \in [m]} f_{\theta_i}(\vec{y}_k) \Leftrightarrow w_k y_k - b_k = \max_{i \in [m]} w_i y_k - b_i$$

The above condition is satisfied for $w_k = 2k$, $y_k = k$ and $b_k = k^2$, as this describes the series of tangents of the strictly convex x^2 function. □

G VARYING DATASET SIZE FOR RECORD-BASED EXPLANATIONS

As illustrated in Figure 15 for increasing datasize the relative number of self-revealing points decreases. However, the decrease is relatively slow (the figure is in logarithmic scale) and the absolute number of revealed points is actually increasing.

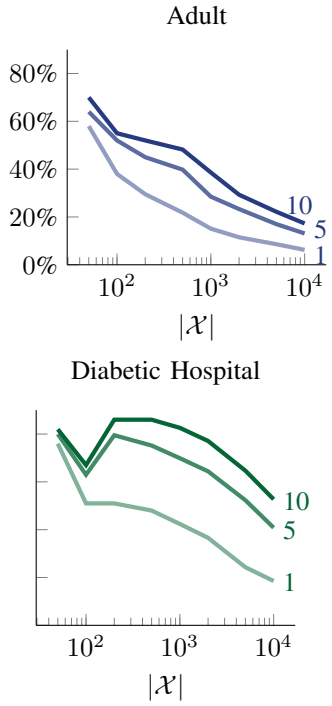


Figure 15: % of k-self-revealing points depending on the size of the dataset ($|\mathcal{X}|$) for $k \in \{1, 5, 10\}$