Exploiting Transparency Measures for Membership Inference: a Cautionary Tale

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

Abstract

Can an adversary exploit model explanations to infer information about the training set? To investigate this question, we focus on *membership inference attacks*: given a datapoint and a model explanation, the attacker's goal is to decide whether 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 develop an attack that can be executed by an attacker that has very limited resources, while maintaining comparable accuracy to existing membership inference attacks. We show that record-based measures can be effectively utilized for membership inference attacks; moreover, 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 majority groups.

1 Introduction

Machine learning models are making increasingly high-stakes decisions in a variety of application domains, such as healthcare, finance and law [14, 21, 11]; 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 inputs and 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 [19]; companies employing black-box models may expose themselves to various risks (e.g. systematically mis-classifying some subgroup of their client base [6], or facing the negative consequences of an automated decision [20]) ; 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 GDPR [12]. 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) [13]. The machine learning (and greater CS) community has taken up the call, offering several novel explanation methods in the past few years (see Section 7). Transparency reports offer users a means of understanding the underlying model and its decsion 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 transparency reports, there has been little effort to address any privacy concerns that arise due to the their release. This is where our work comes in.

Our Contributions We begin our investigation by asking the following simple question.

Can an adversary leverage transparency reports in order to infer private information?

We focus on inferring the presence of individual data points in training set of the model, using *membership inference attacks* [29] and *reconstruction attacks*. We analyze feature-based explanation algorithms, with the emphasis on gradient-based methods, and record-based algorithms, with the emphasis on methods that report influential data points. Our main contributions with respect to gradient based explanations are as follows:

- We design membership inference attacks under the assumption that the attacker has (additional) access to model explanations (Section 4).

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

- We study our attack model on several datasets commonly used in the privacy and interpretability community (Section 5). Our experiments show that in some settings, attacks based on explanations (given in the form of gradients) can achieve performance comparable to the original attack model proposed by Shokri et al. [29]; however, while Shokri et al. [29] assume that one has access to the distribution over labels (for the input vector $\vec{x}$, the attacker observes how likely the model believes it belongs to any class) and the true label, we study several weaker assumptions e.g. only access to the model's final decision, and a model explanation.
- We analyze the potential reasons for the efficacy of our attack model, and demonstrate that the main information leakage stems from the 1-norm of the gradient (Section 5.3). This has interesting implications for possible defenses.
- On synthetic datasets, we study the influence of the input dimension on the success of membership inference using gradient-based explanations (Section 6).

In the supplementary material we further study membership inference attacks based on additional feature-based explanations (including Integrated Gradients [36] and DeepLIFT [30]). These membership inference attacks achieve comparable, albeit weaker, success than gradient-based attacks.

In a *full version* [28] of this paper we include our results for record-based explanations.

- We show how to successfully implement a basic membership inference attack based on record-based explanations.
- We study two types of dataset reconstruction attacks based on record-based explanations, enabling an attacker with little prior knowledge to recover significant parts of the training data.
- Finally, we explore the vulnerability of minorities and outliers in the training data to being revealed for record-based explanations. Our explorations indicates that minorities are particularly vulnerable. This raises significant concerns for the actual deployment of the explanation methods in high-stakes domains.

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 linear models is parameterized by a coefficient w_i for each feature, thus $\Theta = \mathbb{R}^n$. We denote the model as a function of its parameters as c_θ. When picking a good model for our data, it is often useful to think in terms of *loss functions*; a 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 [27] for an overview).

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

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

2.1 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. [29], and is somewhat weaker than the model studied by [22].

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) $\vec{x}_0 \in \mathbb{R}^n$. 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* $\vec{x}_0 \in \mathbb{R}^n$. In addition, explanation methods sometimes assume access to additional information, such as active access to model queries (e.g. [1, 9, 26]), a prior over the data distribution [5], or knowledge of the model class (e.g. that the model is a neural network [3, 30, 36], or that we know the source code [8]). The output of an explanation function $\phi(\mathcal{X}, c, \vec{x}_0, \cdot)$ can be quite diverse; in this work we focus on two explanation paradigms: *record-based* explanations [16][2], and *numerical* influence measures. More formally, record-based explanations output a set of points $\phi(\mathcal{X}, c, \vec{x}_0, \cdot) \subseteq \mathcal{X}$, whereas feature-based numerical influence measures output a vector in $\mathbb{R}^n$, where $\phi_i(\mathcal{X}, c, \vec{x}_0, \cdot)$ corresponds to the importance

[2]Koh and Liang [16] 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.

of the i-th feature in determining the label of $\vec{x}_0$. In particular, we focus on gradient-based methods [32]. In what follows we often refer to the explanation of the POI $\vec{x}_0$ as $\phi(\vec{x}_0)$, 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 $\phi(\vec{x}_0)$ is a vector in $\mathbb{R}^n$, where $\phi_i(\vec{x}_0)$ is the degree to which the i-th feature influences the label assigned to $\vec{x}_0$. Generally speaking, high values of $\phi_i(\vec{x}_0)$ imply a greater degree of effect; negative values imply an effect for *other labels*; if $\phi_i(\vec{x}_0)$ is close to 0, this normally implies that feature i was largely irrelevant in producing the label of $\vec{x}_0$.

Gradient-Based Explanations Simonyan, Vedaldi, and Zisserman [32] introduce gradient-based explanations 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, Greenside, and Kundaje [31] 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] [2] which we use in our experiments; a discussion of these measures and the relations between them can also be found in [4].

3.2 Record-Based Model Explanations

The approach proposed by Koh and Liang [16] aims at identifying influential *datapoints*; that is, given a point of interest $\vec{x}_0$, find a subset of points from the training data $\phi(\vec{x}_0) \subseteq \mathcal{X}$ that explains the label $c_{\hat{\theta}}(\vec{x}_0)$, where $\hat{\theta}$ is a parameterization choice minimizing total loss as per Equation (1). Koh and Liang propose selecting a training point $\vec{z}_{\text{train}}$ by measuring the importance of $\vec{z}_{\text{train}}$ for determining the prediction for $\vec{x}_0$.

In order to estimate the effect of $\vec{z}_{\text{train}}$ on $\vec{x}_0$, Koh and Liang measure the difference in the loss function over $\vec{x}_0$ when the model is trained with and without $\vec{z}_{\text{train}}$. More formally, Koh and Liang define

$$\tilde{\theta}_{\text{train}} \triangleq \text{argmin}_{\theta \in \Theta} \frac{1}{|\mathcal{X}| - 1} \sum_{\vec{x} \in \mathcal{X} \setminus \{\vec{z}_{\text{train}}\}} L(\vec{x}, \theta) \quad (2)$$

In other words, $\tilde{\theta}_{\text{train}}$ minimizes empirical loss over the dataset excluding $\vec{z}_{\text{train}}$. The influence of $\vec{z}_{\text{train}}$ on $\vec{x}_0$ is then

$$I_{\vec{x}_0}(\vec{z}_{\text{train}}) \triangleq L(\vec{x}_0, \tilde{\theta}_{\text{train}}) - L(\vec{x}_0, \hat{\theta}). \quad (3)$$

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 $I_{\vec{x}_0}(\vec{z})$ as per Equation 3), which allows users to gauge their relative importance.

[3] https://github.com/albermax/innvestigate

4 Membership Inference Attacks Using Gradient-Based Explanations

This section describes our baseline membership inference attack, which is based on the attack developed by Shokri et al. [29]. The underlying idea is to capture membership inference as a learning problem; the attacker wants to train 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. [29] circumvent this by training *shadow models*, 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 we assume a best case scenario were the attacker actually has membership information of some datapoints, forgoing the additional step of training shadow models.

The attack model is a neural network inspired by the architecture of Shokri et al. [29]. The network consists of multiple sub-networks (see Figure 5 in Appendix A for illustration). Which sub-networks are used depends on the types of information 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. [29] to ensure a valid comparison between the methods.

The different assumptions on the attacker's access to model data the attacker give rise to seven different scenarios. We ignore the potential eighth 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.

5 Experimental Evaluation of Membership Inference Attacks Via Gradient-Based Explanations

In this section we analyze our experiments for membership inference attacks relying on gradient-based explanations.

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

5.1 Target datasets and Target Models

An overview of the target datasets for our experiments can be found in Table 1 and a more extensive description in Appendix B. For each dataset and each target model we subsample 10,000 points each for training and testing.

Where possible we used the same target model an training configuration as used in [29] all of which are fully connected multi-layer networks with $\tanh$ activations. The CiFAR-100 network has two convolutional layers following the input layer. The Diabetic Hospital dataset was not used previously to study membership inference. Therefore 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 overfitting (i.e. early stopping).

5.2 Training for Different Levels of Overfitting

It is known (see e.g. [23]) that the degree of overfitting to the training data significant 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. To investigate how overfitting influences explanation-based attacks, we train models with different degrees of overfitting, measured as the difference between training and test accuracy. The difference between training and test accuracy is a standard measure of overfitting.

To keep the different instances 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 amount of overfitting never exceeded 0.1; for the Adult dataset, the degree of overfitting was negligible, so we exclude it 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 the amount of overfitting strongly correlates with the training accuracy and number of epochs trained. The training accuracy steadily increases with the number of epochs, while the increase in test accuracy eventually plateaus. Between different datasets the accuracy and number of epochs needed to obtain overfitting gaps differs widely. 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%. Reporting the gap between training and testing accuracy allows in our eyes for most comparability between the different datasets. To avoid correlation within one training run affecting the result we train each target model indivdually (i.e. we use each independently seeded and trained model exactly once).

5.3 Results and Evaluation

Figure 1 displays the accuracy of the attacking networks for the different targets on an evaluation set.

The main observations can be summarized as follows:

Observation 5.1 (Overfitting)**.** Overfitting affects all attacks in a similar manner: overfitted models are more vulnerable, with the vulnerability apparently growing linearly in the degree of overfitting.

Observation 5.2 (Performance)**.** The type of information exploited by the attacker varies between datasets. For the purchase dataset, most attacks behave in a similar manner, with the only the attack based on the true label alone underperforming significantly. For the Texas dataset, gradient-based attacks underperform the prediction-based attack. For Cifar-100, all attacks perform relatively poorly, with access to gradients offering the worst performance guarantee. For the hospital readmission datasets, attacks with access to the true label considerably outperform the rest; this is remarkably not the case for the other datasets.

Observation 5.3 (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 are going to analyze which factors lead to the difference in performance and explain the substitutional behavior of the different types of information. We see two main factors for the results: The structure of the target model and the dimensionality of the target dataset.

Hypothesis 5.4. *When* $\tanh$ *(or, similarly sigmoid and 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 points that are part of the training data. Direct access to the prediction vector (as is the case in [29]), or indirect access to this information via the gradient offers an avenue for the attacker. We verify this hypothesis in several partially redundant ways.

1. If the information leakage actually comes from the difference in $||\nabla c(\vec{x})||_1$, a simple model base only on the 1-norm achieves comparable results. This is in fact the case (see Figure 2). A decision tree trained only on the 1-norm achieves a competitive accuracy compared with a neural network trained on the entire gradient for the purchase dataset and Texas dataset. For Cifar-100 the decision tree

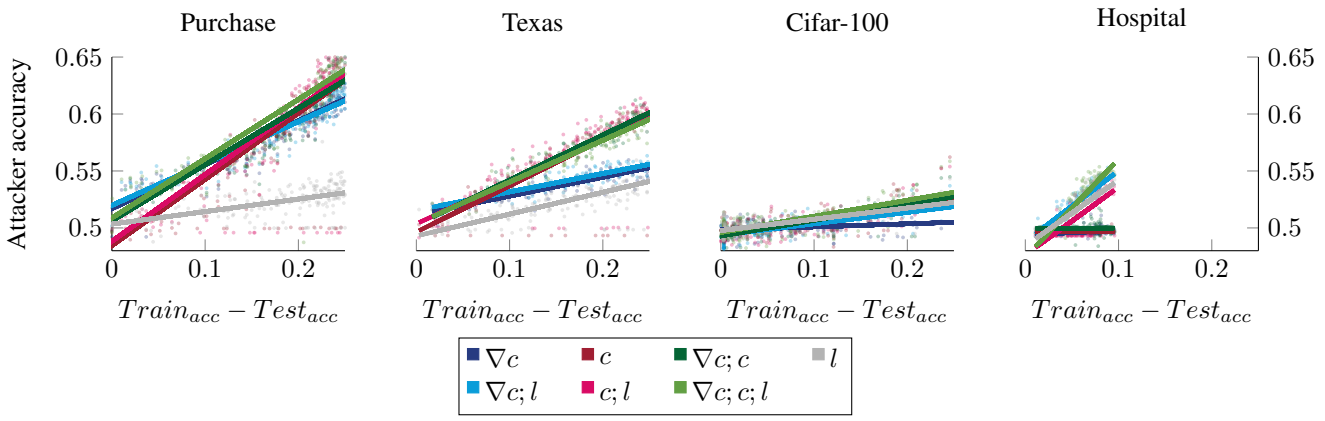

Figure 1: The results of the baseline attack model where the attacker uses a neural network for membership inference and has access to different types of information. The results for the Adult dataset are not presented, as no attack outperformed a random guess.

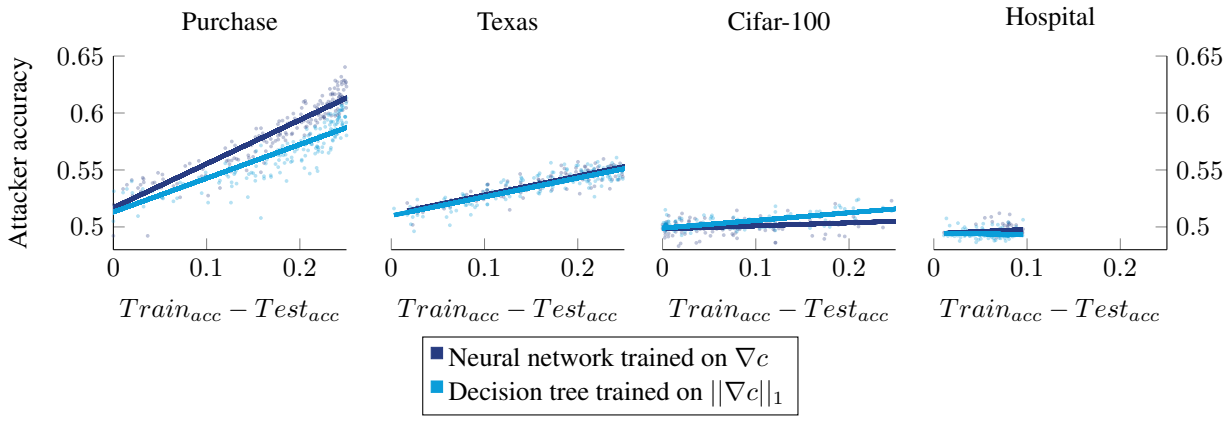

Figure 2: Comparison between an attack with a neural network trained on the entire gradient vs. a decision tree using only the 1-norm of the gradient. While for the Purchase dataset the network slightly outperforms the tree, the results for the Texas dataset are almost identical. For Cifar-100, the decision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

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 perform considerably worse. In fact, attacks trained on normalized gradients do not even outperform the random baseline.

3. If the difference in 1-norm is connected to variance in the prediction, 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 inverts as overfitting increasing.

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 6.

6 The Influence of the Input Dimension

The experiments in Section 5 indicate that $||\nabla c(\vec{x})||_1$ may indicate membership in the training set. 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. [29] show how classifier uncertainty can be exploited for membership attacks, further reinforcing this intuition. Let us next study this phenomenon on synthetic

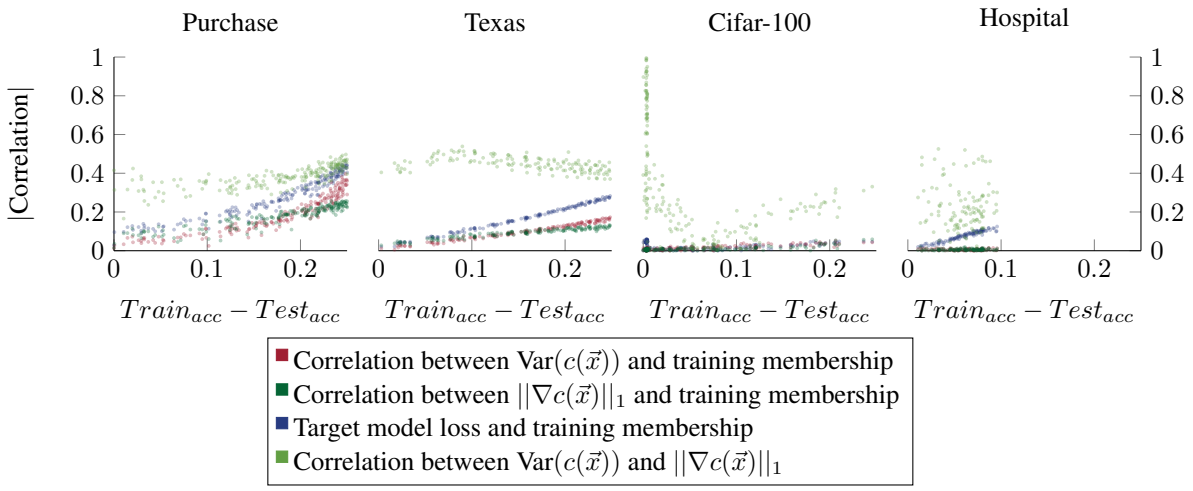

Figure 3: The correlation between training set membership and several key signals an attacker can observe. We display the absolute value for ease of comparison ($Var(c(\vec{x}))$ is negatively correlated with membership). Notably, even though for our attack $c(\vec{x})$ and $\nabla c(\vec{x})$ seem to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

datasets, and the extent to which an adversary can exploit model gradient information in order to conduct membership inference attacks. We use artificially generated datasets; this offers us control over the problem complexity, and helps identify important facets of information leaks.

To generate datasets, we use the `make_classification` function of the Sklearn python library.[4] 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. In our experiments, the number of classes is either 2 or 100 while the number of features increases in steps from 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 and 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 7 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 the number of points close to a decision boundary. Increasing the number of classes increases the complexity of the learning problem (e.g., as measured in VC-dimension).

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

Figure 4 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. For a the number of features the other datasets fall into ($10^3 \sim 10^4$) there is a correlation, which starts to decrease for even higher dimensions. For the correlation 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 $10^3 \leq n \leq 10^3$ the training accuracy is high, leading to over-fitting, just on a lower level. We also did 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 8 in Appendix E).

7 Related Work

Our work studies the vulnerability of transparency reports to membership inference attacks. We primarily focus on two types of transparency reports: datapoint-based influence measures using influence functions, proposed by Koh and Liang [16], and numerical influence measures [5, 7, 9, 26, 33]. Datta, Sen, and Zick [9] 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. [22] show that gradient-based model explanations can be used to reconstruct the underlying model with high accuracy; their work serves as additional evidence that transparency reports are vulnerable to inference attacks.

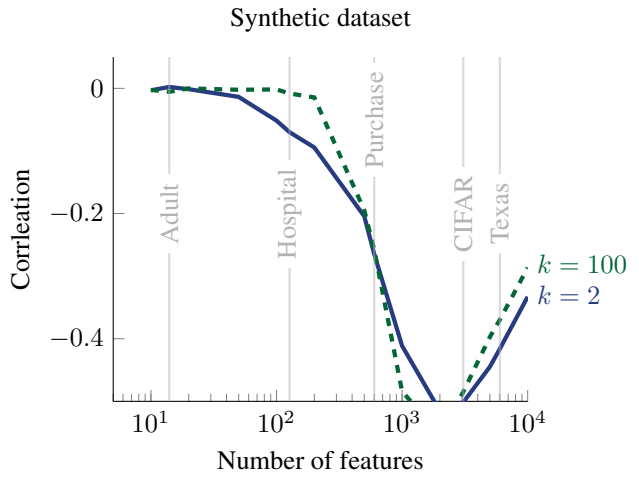

Figure 4: The correlation between $||\nabla c(\vec{x})||_1$ and training membership for synthetic datasets for increasing number of features n and different number of classes $k \in \{2, 100\}$

Ancona et al. [4] provide a recent overview of numerical influence measures (also called attribution methods). Generally this approach can be divided into perturbation-based methods which generate the influence of each feature by altering (also removing or masking) the original input and comparing the difference in the output and backpropagation-based methods which rely on a single (or very small number of) back-propagations through the network.

The intuition behind backpropagation-based methods is to map influence back from the output to the input. The most canonical example is the gradient, however several variations have been proposed. While these methods are generally fast, they tend to be more noisy and often harder to interpret.

In the category of perturbation-based methods fall occlusion based methods [38], but also LIME [26] which trains a simpler model with high local fidelity and QII [9] which computes the Shapley value of each feature. The reliance of these methods on sampling makes them comparatively slow and also prone to query counterfactuals (i.e. data points that could never actually occur). Yet, they tend to give more stable and less noisy explanations. Further, the sampling can be seen as a natural defense against privacy loss. Our analysis focuses on the former group leaving the latter for future study.

The attack scenario we adopt has been recently proposed by Shokri et al. [29]. Shokri et al. [29] use model predictions for data with known membership to train classifiers that predict training set membership with high accuracy. However, Shokri et al. [29] assume access to the full probabilistic prediction of the model over the datapoints as well as the true label; we assume more realistic scenario, where one has access to the datapoint labels, and a given transparency report and the true label is unknown. Further, a form of our attack doesn't require the training of a neural network and requires only the 1-norm of the explanation as input.

Our analysis for record-based explanations in the full version of this work indicates that outliers are more vulnerable to membership inference attacks than other datapoints: the attacker is likelier to identify them as part of the training set due to their distinctive characteristics. This is in line with exisitng results showing that overfitting may cause information leaks [37].

There exists some work on the defense against privacy leakage. Nasr, Shokri, and Houmansadr [23] use adversarial regularization, while Papernot et al. [24] and [25] create a framework for differentially private training of machine learning models. However, these techniques are not yet widely adapted and it is especially unknown how they affect the transparency of the trained models.

8 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. We show that feature-based explanations can be successfully exploited by an attacker to infer membership of the training set.

Our work is one of the first to show that releasing transparency reports can result in significant privacy risks. While we are supportive of the call to algorithmic transparency, we believe that it is the duty of the computer science community at large to ensure that policy makers and advocacy groups are aware of the risks and tradeoffs involved in offering greater model transparency.

In the full version of this paper [28] we included a dataset reconstruction attack that exploits the underlying structure of record-based explanations. For high dimensional data this attack, under mild constraints, allows the recovery of (almost) the entire dataset.

Our results are just a first step towards a better understanding of transparency-based privacy attacks; several interesting open problems remain. First, it is not clear what are sufficient conditions for dataset safety. Low dimensionality of the data seems beneficial, but that needs to be further analyzed.

Finally, designing safe transparency reports is an important research direction: in more detail, one needs to release explanations that are both *safe*, and *useful* (in some formal sense). 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 tradeoff between explanation quality and its privacy guarantees will help us understand the capacity to which we can explain model decisions, while maintaining data integrity.

Acknowledgment

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number AISG-RP-2018-009).

References

[1] Adler, P.; Falk, C.; Friedler, S. A.; Rybeck, G.; Scheidegger, C.; Smith, B.; and Venkatasubramanian, S. 2018.

Auditing black-box models for indirect influence. In *Knowledge and Information Systems*.

[2] Alber, M.; Lapuschkin, S.; Seegerer, P.; Hägele, M.; Schütt, K. T.; Montavon, G.; Samek, W.; Müller, K.; Dähne, S.; and Kindermans, P. 2018. iNNvestigate neural networks! *arXiv preprint arXiv:1808.04260*.

[3] Ancona, M.; Ceolini, E.; Öztireli, A. C.; and Gross, M. H. 2017. A unified view of gradient-based attribution methods for Deep Neural Networks. *CoRR*.

[4] Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[5] Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Mueller, K. 2009. How to Explain Individual Classification Decisions. *Journal of Machine Learning Research* 11:1803–1831.

[6] Buolamwini, J.; Gebru, T.; and Hubert, A. 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.

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

[8] Datta, A.; Fredrikson, M.; Ko, G.; Mardziel, P.; and Sen, S. 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)*.

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

[10] Dua, D., and Graff, C. 2017. UCI machine learning repository.

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

[12] Goodman, B., and Flaxman, S. 2017a. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38(3):50–57.

[13] Goodman, B., and Flaxman, S. 2017b. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38(3):50–57.

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

[15] Klauschen, F.; Müller, K.; Binder, A.; Montavon, G.; Samek, W.; and Bach, S. 2015. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *Plos One*.

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

[17] Koh, P. W., and Liang, P. 2017b. Understanding Black-box Predictions via Influence Functions. *arXiv preprint arXiv:1703.04730*.

[18] Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

[19] Lan, C., and Huan, J. 2017. Discriminatory transfer. *arXiv preprint arXiv:1707.00780*.

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

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

[22] Milli, S.; Schmidt, L.; Dragan, A. D.; and Hardt, M. 2018. Model Reconstruction from Model Explanations. *arXiv preprint arXiv:1807.05185*.

[23] Nasr, M.; Shokri, R.; and Houmansadr, A. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *Proceedings of the 25th ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[24] Papernot, N.; Abadi, M.; Erlingsson, Ú.; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

[25] Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, Ú. 2018. Scalable Private Learning with PATE. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[26] Ribeiro, M. T.; Singh, S.; and Guestrin, C. 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.

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

[28] Shokir, R.; Strobel, M.; and Zick, Y. 2019. Privacy risks of explaining machine learning models. *arXiv preprint arXiv:1907.00164*.

[29] Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership Inference Attacks Against Machine Learning Models. In *Proceedings of the 38th IEEE Conference on Security and Privacy (Oakland)*.

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

[31] Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017b. Not just a black box: Learning Important Features Through Propagating Activation Differences. *arXiv preprint arXiv:1605.01713*.

[32] Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*.

[33] Sliwinski, J.; Strobel, M.; and Zick, Y. 2019. Axiomatic Characterization of Data-Driven Influence Measures for Classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

[34] Smilkov, D.; Thorat, N.; Kim, B.; Viegas, F.; and Wattenberg, M. 2017. SmoothGrad : removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

[35] Strack, B.; Deshazo, J. P.; Gennings, C.; Olmo, J. L.; Ventura, S.; Cios, K. J.; and Clore, J. N. 2014. Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed Research International*.

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

[37] Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2017. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *arXiv preprint arXiv:1709.01604*.

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

A Architecture of attacker network

The attack model is a neural network inspired by the architecture of Shokri et al. [29]. The network consists of multiple sub-networks (see Figure 5 for illustration). Which sub-networks are used depends on the types of information 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.

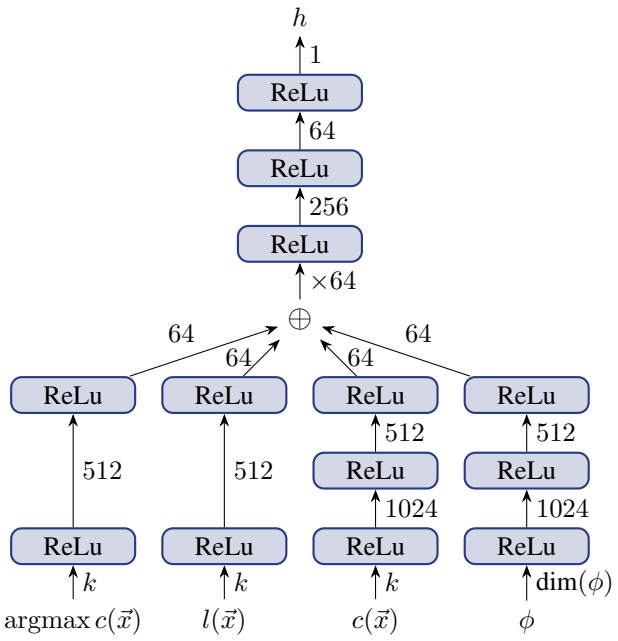

Figure 5: The design of the neural network attack model.

B Target datasets

B.1 Purchase dataset

The dataset originated from the "Acquire Valued Shoppers Challenge" on Kaggle[5]. 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. [29] 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 [29] — represents significant access to information.

B.2 Texas hospital stays

The Texas Department of State Health Services released Hospital Discharge Data public use files spanning from 2006 to 2009.[6] 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 patient's primary procedures based on the remaining attributes (excluding the sec-

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

[6] https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm

ondary 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 [18]. 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 [10]. 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 [35]. We use the modified version described in [17] 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[7] [2]. A in depth discussion of some of these measures and the relations between them can also be found in [4].

Integrated Gradients Sundararajan, Taly, and Yan [36] 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

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

[7] https://github.com/albermax/innvestigate

Layer-wise Relevance Propagation (LRP) Klauschen et al. [15] 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:

$$r_i^{(L)}(\vec{x}) \triangleq c_i(\vec{x})$$

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

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, Greenside, and Kundaje [31] 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" [31] is not considered in this work.

Smooth gradient Smilkov et al. [34] 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 6 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 attack performs worse for LRP and Deeplift. In fact these two

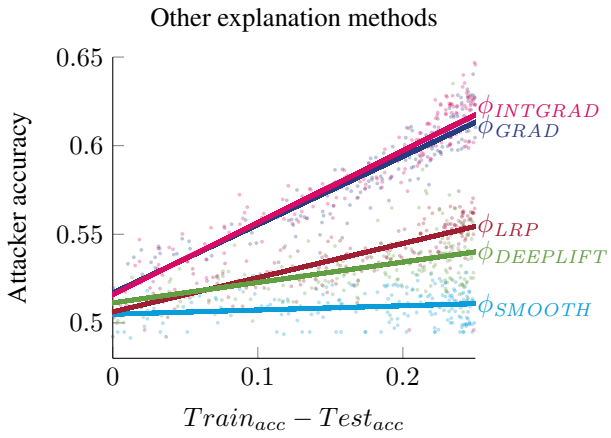

Figure 6: A comparison of the accuracy of the membership inference attack on the purchase dataset for different explanation methods the attacker might exploit.

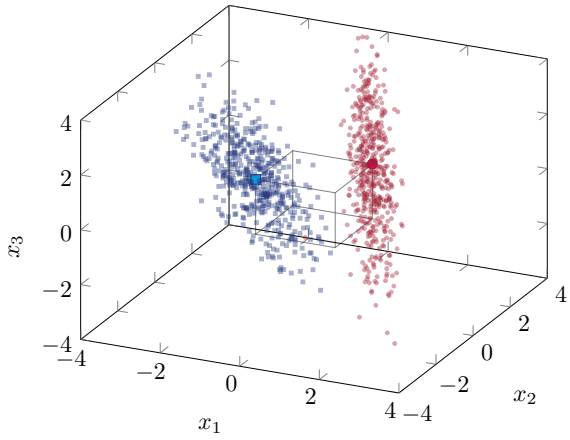

Figure 7: An illustration of the dataset generation process with $n = 3$.

methods are further away from 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 7 for an illustration of the $n = 3$ case.

E Varying the capacity of the neural network for synthetic datasets

Figure 8 illustrates how the correlation between $||\nabla c(\vec{x})||_1$ and training membership is influenced by the capacity of the target network.

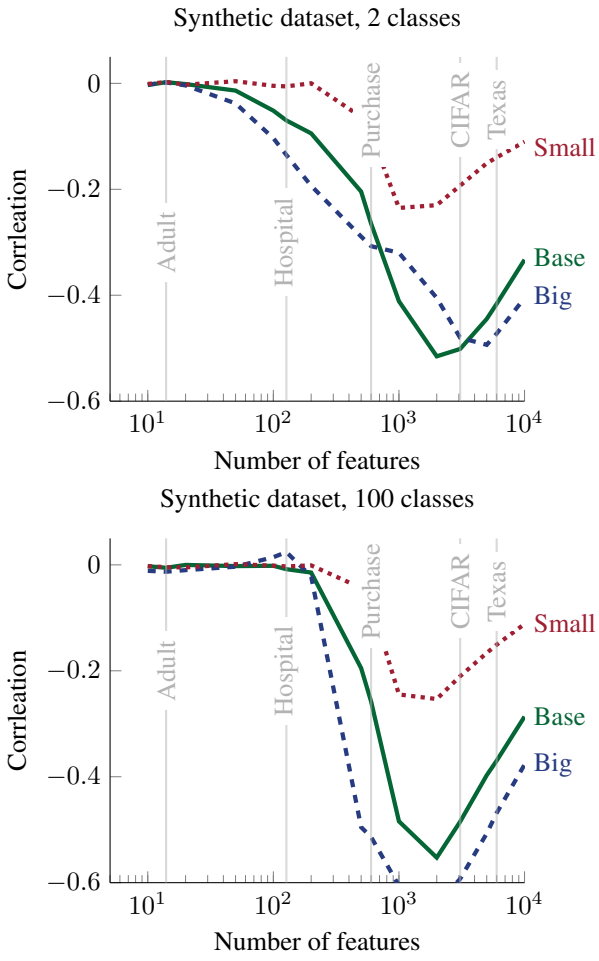

Figure 8: The correlation between $||\nabla c(\vec{x})||_1$ and training membership for synthetic datasets for increasing number of features n and different number of classes $k \in |2, 100$ for three different networks. The "Small" has one hidden layer with 5 nodes, "Base" has two layers with 50 nodes each, "Big" has 3 layers with 100 nodes each.