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 we trust black-box machine learning with its decisions? Can we trust algorithms to train machine learning models on sensitive data? Transparency and privacy are two fundamental elements of trust for adopting machine learning. In this paper, we investigate the relation between interpretability and privacy. In particular we analyze if an adversary can exploit transparent machine learning to infer sensitive information about its training set. To this end, we perform membership inference as well as reconstruction attacks on two popular classes of algorithms for explaining machine learning models: feature-based and record-based influence measures. We empirically show that an attacker, that only observes the feature-based explanations, has the same power as the state of the art membership inference attacks on model predictions. We also demonstrate that record-based explanations can be effectively exploited to reconstruct significant parts of the training set. Finally, our results indicate that minorities and special cases 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 [11, 14, 18]; 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. 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 [17]; companies employing black-box models may expose themselves to various risks (e.g. systematically mis-classifying some subgroup of their client base, or facing the negative consequences of an automated decision without having the capacity to explain it) [21]; 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 [21]. 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* [12]. 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 transparency measures, there has been little effort to address any privacy concerns that arise due to the release of transparency reports. This is where our work comes in.

Our Contributions. We start 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* [27] 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.

We show that gradient-based *numerical influence measures*, including saliency maps [30], Integrated Gradients [34] and DeepLIFT [28], can be used to accurately predict training set membership. We accomplish this by observing the difference in the 1-norm of the explanations for members and non-members. Our gradient-based attack model achieves similar performance to the original attack model proposed by Shokri et al. [27], while utilizing a much simpler label space: rather than observing the model's underlying label distribution (which contains a significant amount of information not normally observed), our attacker only observes the label, and its gradient-based transparency report. We achieve this by carefully adapting the original membership inference attack model to the transparency domain, and show how the information we employ is intrinsically related to key local parameters of the underlying model.

In the case of record-based explanation methods, the attack model is particularly appealing for our purposes because some transparency reports reveal the identity of prominent training set members as a mode of explanation [16]. We show that an attacker can exploit record-based explanations to do much more than infer the identity of a single point: it can infer a significant proportion of the training data, by carefully using the explanation method as an oracle to explore the training set.

For both explanation types, our exploration indicates that minorities and outliers in the training data are particularly vulnerable to being revealed; this raises significant concerns for the actual deployment of the explanation methods in high-stakes domains.

2 PRELIMINARIES

We use the following basic notation: vectors are written as $\vec{x}$; given an integer m, we write $[m] = \{1, \dots, m\}$. We are given a *dataset* $\mathcal{X} \subseteq \mathbb{R}^n$, which we wish to label by a *model* c, mapping each *datapoint* $\vec{x} \in \mathcal{X}$ 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

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

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 $\vec{x}$. In this work, we often distinguish between the labels assigned by a trained model c, and the true data labels, given by $\ell : \mathcal{X} \longrightarrow [k]$.

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 the weight w_i coefficient 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 $L : \mathcal{X} \times \Theta \longrightarrow \mathbb{R}$ takes as input the model parameters θ and a point $\vec{x}$, and outputs a real value $L(\vec{x}, \theta) \in \mathbb{R}$. Simple loss functions would include the square loss for binary classification — $(c_\theta(\vec{x}) - \ell(\vec{x}))^2$ where $\ell(\vec{x})$ is the true data label — or include additional regularization parameters over θ (see [26] for an overview).

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

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

3 EXPLAINING MACHINE LEARNING

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) $\vec{x}_0 \in \mathcal{X}$. An explanation ϕ takes as input the dataset $\mathcal{X}$, labels over $\mathcal{X}$ — given by either the true labels $\ell : \mathcal{X} \longrightarrow [k]$ or by a trained model c — and a *point of interest* $\vec{x}_0 \in \mathcal{X}$. In addition, explanation methods sometimes assume access to additional information, such as active access to model queries (e.g. [1, 10, 24]), a prior over the data distribution [5], or knowledge of the model class (e.g. that the model is a neural network [3, 28, 34], or that we know the source code [9]). 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 of the i-th feature in determining the label of $\vec{x}_0$. In particular, we focus on gradient-based methods [30]. 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 a 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*, and 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 et al. [30] introduced 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 et al. [29] 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 $\vec{x} \circ \phi_{GRAD}(\vec{x})$ and $\phi_{GRAD}(\vec{x})$ are equivalent.

Integrated Gradients. Sundararajan et al. [34] 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. [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:

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

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. [29] 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

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

[2] Koh and Liang [16] refer to their explanations as *influence measures*, which the current authors found to be too generic.

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" [29] is not considered in this swork.

The techniques above 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

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

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

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

In robust statistics, this technique is known as *influence functions* (hence its name in Koh and Liang [16]). If the value $I_{\vec{x}_0}(\vec{z}_{\text{train}})$ is positive, this means that $\vec{z}_{\text{train}}$ plays a significant role in determining the outcome of $\vec{x}_0$. Therefore, points for which (3) is high are presented to the user in order to explain the label of $\vec{x}_0$. The key challenge here is that retraining a model on $\mathcal{X} \setminus \{\vec{z}_{\text{train}}\}$ for every point $\vec{z}_{\text{train}} \in \mathcal{X}$ is prohibitively expensive, rendering this approach inapplicable; Koh and Liang suggest approximating (3) using a result about *influence functions* by Cook and Weisberg [7], as we do in our analysis.

Koh and Liang use a quadratic approximation of the empirical loss around the original parameters θ of the model, followed by a Newton step in the direction of removing a data point (Technically the weight of the loss at this point for the overall loss is decreased.). One can then apply the chain rule to compute how the change in the loss affects the function c at the POI.

$$\begin{aligned} I_{\vec{x}_0}(\vec{z}_{\text{train}}) &\approx \left.\frac{L(\vec{x}_0, \tilde{\theta}_{\text{train},\epsilon})}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta L(\vec{x}_0, \tilde{\theta}_{\text{train}})^T \left.\frac{\tilde{\theta}_{\text{train},\epsilon}}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta L(\vec{x}_0, \tilde{\theta}_{\text{train}})^T H_{\tilde{\theta}}^{-1} \nabla_\theta L(\vec{z}_{\text{train}}, \tilde{\theta}_{\text{train}}), \end{aligned}$$

where $\tilde{\theta}_{\text{train},\epsilon} \triangleq \text{argmin}_{\theta\in\Theta} \frac{1}{|\mathcal{X}|} \sum_{\vec{z}\in\mathcal{X}} L(\vec{z},\theta) + \epsilon L(\vec{z}_{\text{train}},\theta)$. The step from the second to the third line follows from a classic result by Cook and Weisberg [7] and $H_{\tilde{\theta}} \triangleq \frac{1}{|\mathcal{X}|} \sum_{\vec{z}\in\mathcal{X}} \nabla^2_\theta L(\vec{z},\tilde{\theta})$ is the Hessian which is, by assumption, positive definite. A record-based explanation releases the k most influential points according to the above definition, as well as the influence of these k points (the values of $I_{\vec{x}_0}(\vec{z})$ as per Equation 3).

4 INFORMATION LEAKAGE THROUGH GRADIENT-BASED EXPLANATIONS

An adversary obtained access to a dataset $\mathcal{X}' \subseteq \mathcal{X}$, and some additional information about $\mathcal{X}'$; can it use $\mathcal{X}'$ in order to infer information about the remaining points in the training set? The attack model proposed by Shokri et al. [27] assumes that the adversary has access to the probabilistic labels assigned to $\mathcal{X}'$ by some model c. We assume that the attacker only has access to *transparency queries*: for every point $\vec{x} \in \mathcal{X}'$ the adversary observes the transparency report $\phi(\mathcal{X}, c, \vec{x})$; however, rather than observing label distributions as per Shokri et al., we assume that the adversary only observes the assigned label of $\vec{x}$. We focus on two types of inference attacks. The first is *membership inference attacks*: given a point of interest $\vec{x}_0 \notin \mathcal{X}'$, can we determine w.h.p. whether $\vec{x}_0 \in \mathcal{X}$? We also consider *dataset reconstruction attacks*, whose objective is to recover as many points from $\mathcal{X}$ as possible.

Let us first analyze membership inference attack models based on gradient-based explanations. While it would appear intuitively clear that knowing the gradient of the model at some points should help, it is not immediately clear *how* and *to what extent*. We present a simple example illustrating how one might use the information in the model gradient to successfully deploy a membership inference attack . We consider a dataset $\mathcal{X} = \{-1, +1\}$ with two points, such that $l(1) = 1$, $l(-1) = 0$. We fit a single layer neural network with sigmoid activation to $\mathcal{X}$ (other activation functions such as tanh and softmax yield similar results); thus, our parameter class is $\theta \in \Theta = \mathbb{R}$ and $f_\theta(x) = \frac{1}{1+e^{-\theta x}}$. We use absolute loss to find the best-fitting f_θ; some simple calculations show that the gradient (i.e. the parameter update) on both points is the same and given as

$$\frac{\partial |c(x) - f_\theta(x)|}{\partial\theta} = -\frac{e^{-\theta}}{(1+e^{-\theta})^2}$$

Note that the update is always positive: θ is increasing in the number of epochs k. Further, for positive θ the update decreases exponentially.

Table 1 lists how θ evolves over training epochs assuming $\theta_0 = 0$ (a randomized choice for θ_0 yields similar behavior, given the positive update rule and the exponential decay in update). Table 1 also shows the gradient value with respect to the training point 1 and a non-training point $\frac{1}{2}$. While the two gradients start out quite similarly the gradient of 1 drops several orders of magnitude before the gradient of $\frac{1}{2}$. Figure 1 illustrates why this happens; the shape of the sigmoid function is such that the gradient decreases in the vicinity of training points, and grows dramatically at points close to the decision boundary. Sharp swings in decision boundary are arguably desirable from a training perspective: they imply that the classifier is relatively certain about the class of large portions of the data region. Note that the points outside the training set and close

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

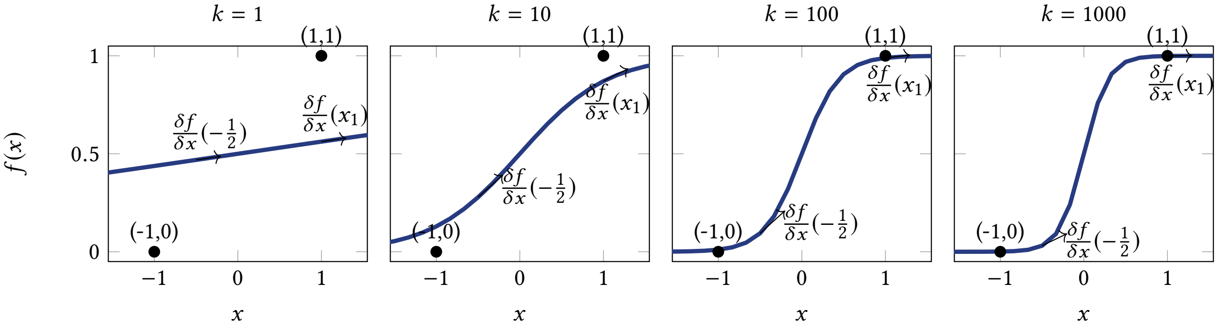


Figure 1: Gradient changes as the number of training epochs (k) grows; the gradient decreases for training points after a while (here 1 and -1), but the gradient for non-training points remains high for much longer (here illustrated at the non-training point $-\frac{1}{2}$).

k	θ	$-\frac{\partial\lvert 1-f_\theta(1)\rvert}{\partial\theta}(\theta)$	$f_\theta(1)$	$\frac{\partial f_\theta}{\partial x}(1)$	$\frac{\partial f_\theta}{\partial x}(\frac{1}{2})$
0	0.0000	0.2500	0.5000	0.0000	0.0000
1	0.2500	0.2461	0.5622	0.0615	0.0623
10	1.9069	0.1126	0.8707	0.2147	0.3829
100	4.5277	0.0106	0.9893	0.0479	0.3862
1000	6.8967	0.0010	0.9990	0.0070	0.2060

Table 1: The change of the single parameter θ during training for 1000 epochs of the toy example of Section 4. The gradient for the training point x_1 decreases many epochs before the gradient of x_3 a point outside the training set.

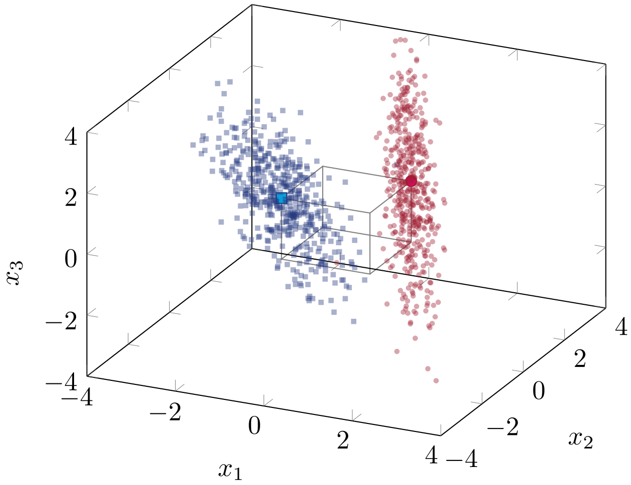


Figure 2: An illustration of the dataset generation process with $n_{inf} = 3$ and $n_{clus} = 1$; the features containing only noise are not shown.

to the decision boundary may end up having a gradient with high absolute values. In other words, high absolute gradient values at a point $\vec{x}$ serve as a signal that $\vec{x}$ is *not* part of the training data, indicating the classifier's uncertainty about the label of $\vec{x}$, and paving the way towards a potential attack; indeed, Shokri et al. [27] show how classifier uncertainty can be exploited for membership attacks, further reinforcing this intuition. Let us next study this phenomenon on complex 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_{inf} informative features, the function creates a n_{inf}-dimensional hypercube. For each class the function picks n_{clus} vertices from the hyper-cube as centers of clusters, and samples points normally distributed around the centers. We fix the number of classes to 2 (see Figure 2 for details). The remaining features are filled with random noise; The overall number of features stays fixed at 200. We increase the number of informative features $n_{inf} \in [1, 5, 50, 10, 199]$; we also vary the number of clusters per class $n_{clus} \in [1, 2, 5, 10]$. Increasing the number of informative features does not increase the complexity of the learning problem as long as $n_{clus} = 1$: an optimal separating hyper-plane offers a good solution to this problem. 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 clusters increases the complexity of the learning problem (e.g., as measured in VC-dimension).

For each experiment we sample 500 points per class and split them evenly into training and test set. We train a fully connected neural network with two hidden layers wit fifty nodes each and the tanh activation function between the layers, and sigmoid 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. The network consistently achieved perfect accuracy on the training set while the accuracy on the test set decreases for increasing n_{clus} from 0.88 for $n_{clus} = 1$ to 0.56 for $n_{clus} = 10$. We cannot detect a major influence of n_{inf} on the performance. We repeat each experiment 30 times with different random seeds.

Next, we compute ϕ_{GRAD} on both training and test points; if training point gradients significantly differ from those of test points, this

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

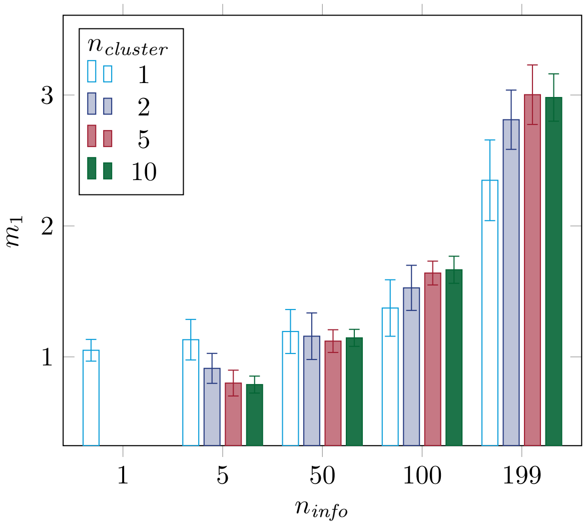


Figure 3: The results for increasing the number of informative features n_{inf} and clusters n_{clus} in artificially generated datasets. The information leakage about membership measured according to m_1 as defined in Equation 4 increases for larger n_{inf}.

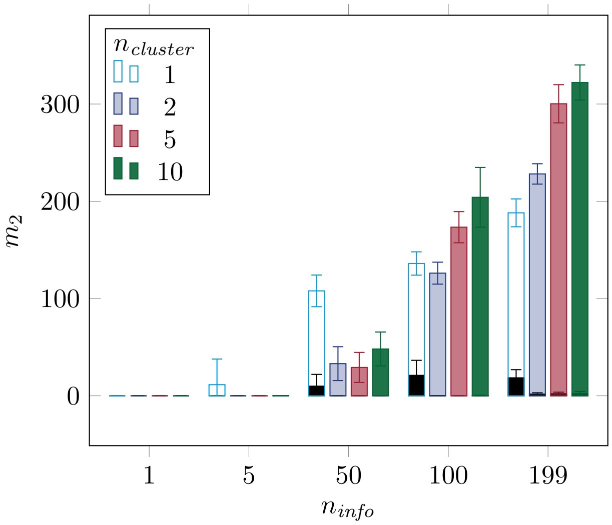


Figure 4: The number of points $\vec{x}$ in the test set which $||\phi_{GRAD}(\vec{x})||_1$ would be considered an outlier with respect to the training set. The number of such outliers in the the training set (black) is marginal.

would indicate a potential for an attack. To measure this, we define m_1: the ratio between the median 1-norm of ϕ_{GRAD} in training and test points.

$$m_1 = \frac{median\{||\phi_{GRAD}(\vec{x})||_1|\vec{x} \in \mathcal{X}_{\text{test}}\}}{median\{||\phi_{GRAD}(\vec{x})||_1|\vec{x} \in \mathcal{X}_{\text{train}}\}}. \tag{4}$$

Figure 3 illustrates how m_1 develops for increasing n_{inf} and n_{clus}. m_1 grows as with n_{inf} increases; this means that member and non-member distributions drift apart, resulting in differentiation in gradients. The effect of larger n_{clus} varies: for low n_{inf}, n_{clus} is negatively correlated with m_1; for large n_{inf} n_{clus} is positively correlated with m_1.

$||\phi_{GRAD}(\vec{x})||_1$ is decreasing in n_{inf} for both members and non-members. However the decrease is faster for members, resulting in an overall increase in m_1. For small n_{inf}, the datapoints have $200 - n_{inf}$ noise features, resulting in a highly noisy dataset; increasing n_{inf} leads to a decrease in noise, thereby making it more prone to overfitting; this leads to $||\phi_{GRAD}(\vec{x})||_1$ decreasing faster for training set members. For large values of n_{inf}, n_{clus} behaves according to our intuition: increasing n_{clus} increases function complexity, leading to greater overfitting and subsequently increasing m_1. For low n_{inf}, a higher n_{clus} also leads to more severe overfitting; however, this does not increase m_1. The value m_1 indicates the differentiation in label distributions for members and nonmembers; if $m1$ is close to 1 the distributions are close. Thus, a small value of m_1 is beneficial for the attacker as well.

The second metric we use to capture information leakage is the number of non-members that would be considered outliers, according to the distribution of $||\phi_{GRAD}(\vec{x})||_1$ for points in the training set. Following a common rule we consider points outliers that are above the threshold α_{outlier} of least 1.5 interquartile ranges iqr above the upper quartile uq [13]. Formally,

$$\alpha_{\text{outlier}} = uq(||\phi_{GRAD}(\mathcal{X}_{\text{train}})||_1) + \frac{3}{2} iqr(||\phi_{GRAD}(\mathcal{X}_{\text{train}})||_1$$

. Given that the number of such outliers in the training set itself is small (only for $n_{clus} = 1$ does it go beyond 3), m_2 indicates how many test points are close to a decision boundary.

$$m_2 = |\{||\phi_{GRAD}(\vec{x})||_1 \geq \alpha_{\text{outlier}}|\vec{x} \in \mathcal{X}_{\text{test}}\}|.$$

Figure 4 shows the values m_2 as well as the number of outliers in the training set. Like m_1, m_2 increases as n_{inf} increases. However, m_2 shows practical consequences for n_{inf}: about the $||\phi_{GRAD}(\vec{x})||_1$ value of half of the 500 points in the test set would be considered outliers with respect to the training set and so easily identifiable by an attacker.

Figure 5 illustrates on two examples how the distributions of $||\phi_{GRAD}(\vec{x})||_1$ for members and non-members drift apart for increasing n_{inf} and n_{clus}. For $n_{inf} = 1, n_{clus} = 1$ the distributions are relatively close. Yet, for $n_{inf} = 199, n_{clus} = 5$ the boxplots barely overlap. To conclude, the information classifier gradients can be used to deduce the difference in label distributions between test points and training points, making it potentially useful for the purpose of membership inference. In what follows, we demonstrate the effectiveness of gradient information in devising effective membership inference attacks on real data.

5 EXPERIMENTS FOR ATTACKS USING FEATURE-BASED EXPLANATIONS

Our discussion in Section 4 indicates that gradient-based explanations could potentially be used to devise effective membership inference attacks; armed with this intuition, let examine the efficacy of these attacks on actual data. We compare our explanation-based attack model to the state-of-the-art in two ways. First, we compare

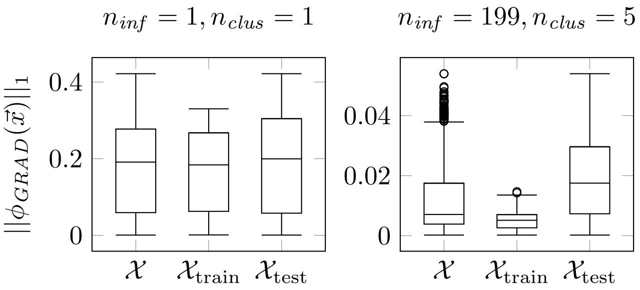


Figure 5: The boxplots illustrate how the distributions of $||\phi_{GRAD}(\vec{x})||_1$ drift apart for points $\vec{x}$ in the training set $\mathcal{X}_{\text{train}}$ and test set $\mathcal{X}_{\text{test}}$. For $n_{inf} = n_{clus} = 1$, the two boxplots are indistinguishable. However, for $n_{inf} = 200, n_{clus} = 5$ they barely overlap. Here $\mathcal{X} = \mathcal{X}_{\text{train}} \cup \mathcal{X}_{\text{train}}$.

our explanation-based approach to attacks using only the prediction *vector* and the predicted *label* as used by Shokri et al. [27] and Nasr et al. [20]. Next, we compare our approach to a simple attack model with limited access to the gradient (based on a one-level decision tree trained on the 1-norm of the explanation) to an attack model relying on a neural network. The latter is an adapted version of the attack model also used by [27]. For ease of comparison we focus our attention on the datasets used in [27].

5.1 Datasets and Target Models

5.1.1 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. [27] 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 (i.e. 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 was the case in [27]) represents a significant amount of information.

5.1.2 Classification Models. As a target model, we use a fully connected four-layer (with a [1024, 512, 256, 100] nodes configuration) neural network. We use the tanh activation function for hidden layers, and softmax in the output layer. We randomly initialize node weights with a 0-mean normal distribution and a 0.01 standard deviation; we train each model on 10 000 datapoints for 100 epochs using the ADAGRAD optimizer with a learning rate of 0.01 and a learning rate decay of 1e − 7.

The original models exhibit significant overfitting; however, this is not a major issue as our primary goal is comparing information leaked by explanations versus the prediction itself, rather than study attacks on perfectly generalized models. Nevertheless, to see how overfitting affects attack effectiveness we introduce a dropout behind each exiting layer. We increase dropout in 0.1 increments from 0 to 0.9. Figure 6 shows how different dropout values affect accuracy on the training and validation set. High levels of dropout reduce overfitting, but result in a performance tradeoff.

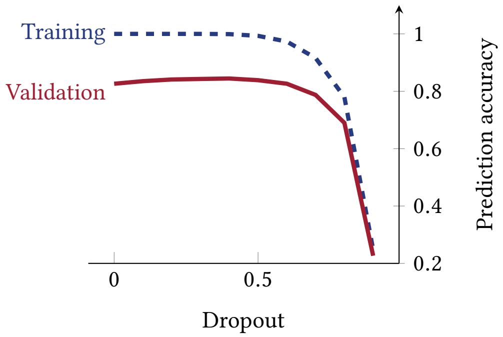


Figure 6: Target model accuracy for the training set (blue) and validation set (red) on the purchase dataset under increasing levels of dropout. High dropout levels reduce overfitting, but decrease accuracy.

5.2 Membership Inference Attack Models

We create two different attack models based on the explanations introduced in Section 3.1. The first is a simple one-level decision tree; the second is based on training a neural network, inspired by the attack model used for membership inference attacks in Nasr et al. [20]. As input for the tree model we use the 1-norm of the respective explanation. Each neural network model has access to the predicted label argmax $c(\vec{x})$ as well as the numerical explanation vector $\phi(\vec{x})$.

As shown in Figure 7, the network consists of three sub-networks. The first sub-network uses the one-hot encoded predicted label as an input and has fully connected layers with sizes [100, 256, 64]. The second network uses the explanation ϕ as input; its layer sizes are $[\dim(\phi), 1024, 512, 64]$, where $\dim(\phi)$ is the explanation dimension. The final part of the network combines the first two and has layers of sizes [256, 64, 1]. We use a ReLu activation between layers and initializing the weights the same way as the target models. Further, we train a neural network for attackers with access to the prediction vector $c(\vec{x})$ and to the actual label $l(\vec{x})$, which makes it more powerful than attackers using only the explanations and predicted label. Finally, we create two more one-level decision tree models as comparisons to the explanation-based decision tree attack models. The first uses the correctness of the target as input (i.e. argmax $c(\vec{x}) == l(\vec{x})$); here information leakage intuitively arises from the fact that target accuracy is higher on the training set. The second tree uses prediction variance as input; as described in [20], lower variance corresponds to prediction uncertainty, indicative of non-membership. Note the difference between 1-norm and variance. While for the explanations high absolute values are indications of being close to a decision boundary, the 1-norm of each prediction is the same.

We train our attack models for 50 epochs using the ADAGRAD optimizer with the learning rate of 10^{-3} and a learning rate decay

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

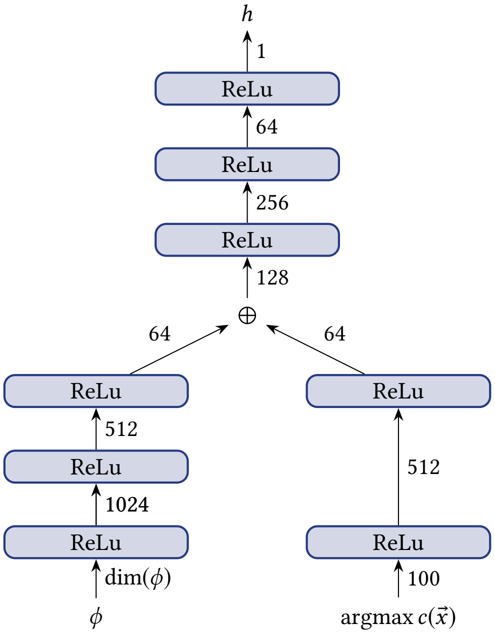


Figure 7: The design of the neural network attack model. It processes the predicted label argmax $c(\vec{x})$ **and the explanation** $\phi(\vec{x})$ **in two sub-networks before combining them in a final, third sub-network.**

of 1e − 6. The training data for the attack models consists of 14 000 labels and predictions/explanations, half of which is known belong to points in the training set, and the other half is known *not* to belong to the training set. The attack models are then are evaluated on the remaining 6 000 points; we repeat each experiment 30 times.

5.3 Empirical Results

The main results of our experiments are illustrated in Figure 8. Attacks with access to the prediction vector and the actual label were the most successful, followed by gradient-based attacks. Simpler tree-based models fared only slightly worse the neural network based attack models; this is despite the fact that the tree-based attackers had only access to the 1-norm of the gradient vectors. This supports the conjecture that the main source of gradient exploitability is its 1-norm. Results for ϕ_{LRP} and $\phi_{DEEPLIFT}$ are less conclusive: both explanation methods leak less information than gradient-based explanations; however, they do offer some advantage to an attacker. As is to be expected, increasing dropout during training decreases attack accuracy; indeed, for a dropout rate of 0.9 all attack models are only marginally better than a random guess. As seen in Figure 6, attack accuracy starts to decrease before there is a significant drop in model accuracy; the only exception to this is for attacks based on the predicted label argmax$(c(\vec{x}))$: this attack exploits the model prediction of the target model and so is only affected once model predictions become highly inaccurate. We only included the results for argmax$(c(\vec{x}))$ of the simple models, given that the only information the attacker can use is weather the prediction is correct, given this information a simple model is as good as an advanced. Further note that in some instances the neural network failed to train (i.e. training accuracy remained close to 50%) we excluded these attacks.

5.4 Class size and distance to the data

The number of training points per class influences how much information is leaked. For classes with only a small number of points the model tends to overfit more, allowing for more accurate attacks in these data regions. This has an unfortunate implication: members of minority classes (e.g. patients who underwent rare procedures) face a higher risk of their membership being exposed. The purchase data set contains groups of varying sizes; we used the output of the neural network/gradient-based explanation attack model to quantify the effect of class size on attacker accuracy.

While class size and attack accuracy are not strongly correlated (Pearson's $r \equiv -0.23$), Figure 9 shows that the maximal accuracy obtained for small classes is signifciantly higher than for larger classes; in other words, the attacker makes more certain predictions on smaller classes. To conclude, smaller classes are subject to a greater risk of membership inference attacks. We use the purchase data to investigate whether data outliers are at a particular risk. We measure the average Hamming distance of a point to the training set and grouped the points by their average Hamming distance. As illustrated in Figure 10 the accuracy of the model drops for increasing distance. In fact the 1-norm of the outliers is generally small and the attack models classify them as members of the training set.

6 ATTACKS ON RECORD-BASED EXPLANATIONS

As we have seen in Section 5, gradient-based explanations can be effectively used to conduct membership inference attacks. Let us next turn to record-based explanations.

6.1 Datasets and Target models

We focus on two datasets and three model types where record-based explanations had been demonstratably successful [16].

6.1.1 Fish vs. Dog. This dataset contains 2400 299 × 299-pixel dog and fish images, extracted from ImageNet [25]. We split it into a training set of 1800 points and a test set of 600 points.

InceptionV3. The inception network architectures are convolutional neural networks that where designed to overcome large variations in the size of salient parts in an image (i.e. the parts of the image containing relevant information). The networks contain convolutional filters of varying sizes on the same level working in parallel [35]. Pre-trained instances of the inception architecture are available in Keras.[6] We use a pre-trained network and retrained the last layer for the specific task of the classifying fish/dogs; our model obtained a 99% test accuracy.

RBF SVM. Support vector machines with radial base function kernels (RBF SVM) are a popular classification method. We use SmoothHinge [16] to overcome the non-differentiable loss function; the SVM model obtains a test accuracy of ∼ 80%.

[6] https://keras.io/applications/#inceptionv3

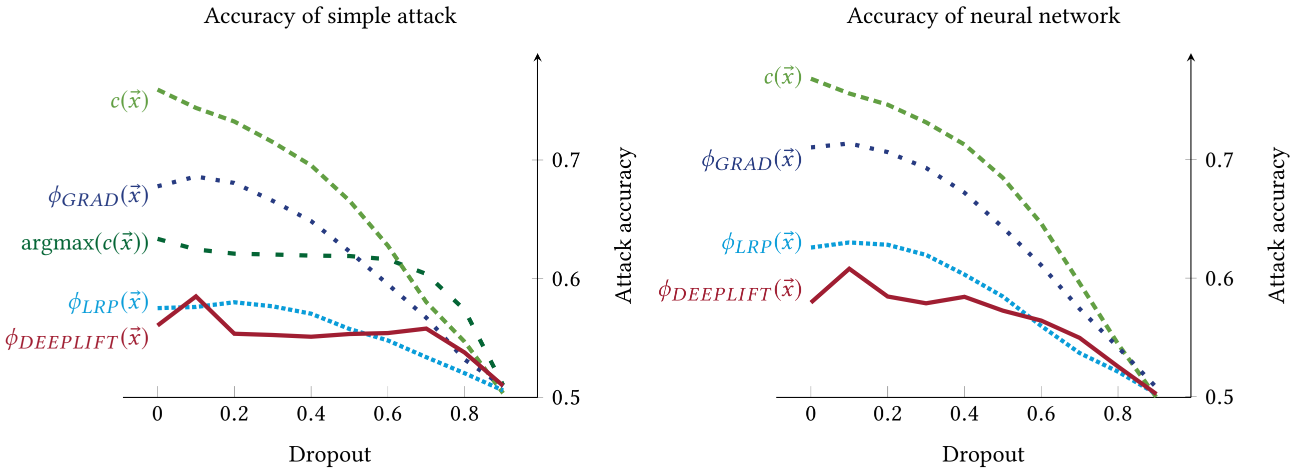


Figure 8: The accuracy of the attacks on the purchase dataset. The simple attack models which a based on the 1-norm of the explanation (left) do as good (or better) as the attack relying on the complete explanation and a neural network (right).

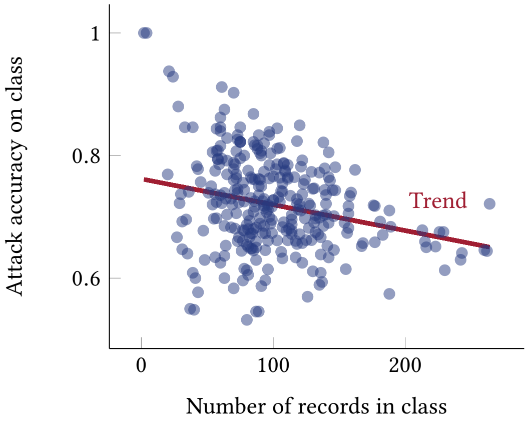


Figure 9: The accuracy of the membership inference attack against purchase classification models using the gradient as explanation. Each dot is the size of a class during training and the accuracy of the corresponding attack model. We only display a subset (300 of 2000) for clarity, the trend is computed on all points. No points below 0.5 are displayed there are 8, mostly for small class sizes.

6.1.2 Diabetic Hospital. The dataset contains data on diabetic patients from 130 US hospitals and integrated delivery networks [33]. We use the modified version described in Koh and Liang [16] 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. The dataset contains 101 766 records from which we sub-sample balanced datasets (i.e. with equal numbers of patients from each class) with 2 000 records from each of the two classes.

Linear Regression. A simple linear regression model obtains a test accuracy of ∼ 65%. Each experiment was repeated 20 times with randomly sampled datasets of 2 000 records; we report the average results.

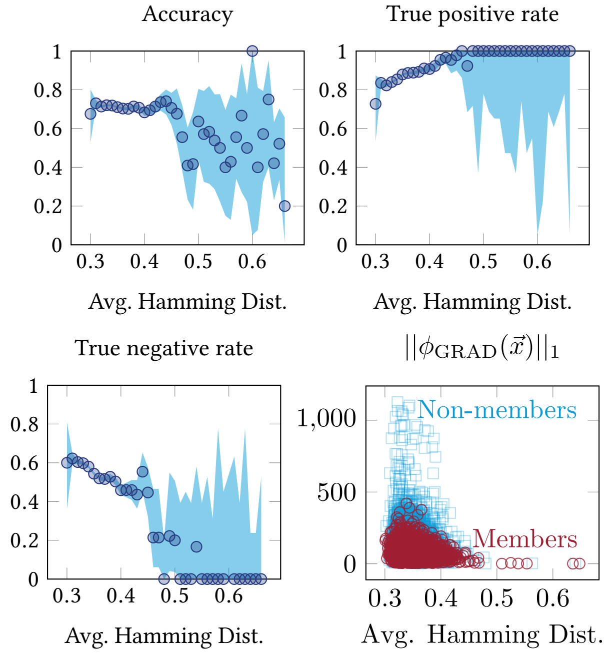


Figure 10: Attack accuracy on the purchase data set, as a function of the mean Hamming distance *d* of the point to the training set. Accuracy declines with increasing distance; in fact, the attacker classifies most outliers as training points, as seen in the change in true positive/negative rates.

6.2 Direct Membership Inference

While feature-based explanations indirectly leak membership information, record-based explanations do so in a much more straightforward manner; as described in Section 3.2, record-based explanations release the k most influential training points given a query. If a query is in fact part of training data, it is likely to be part of its own explanation. Our experiments confirm this intuition (Figure 11). For

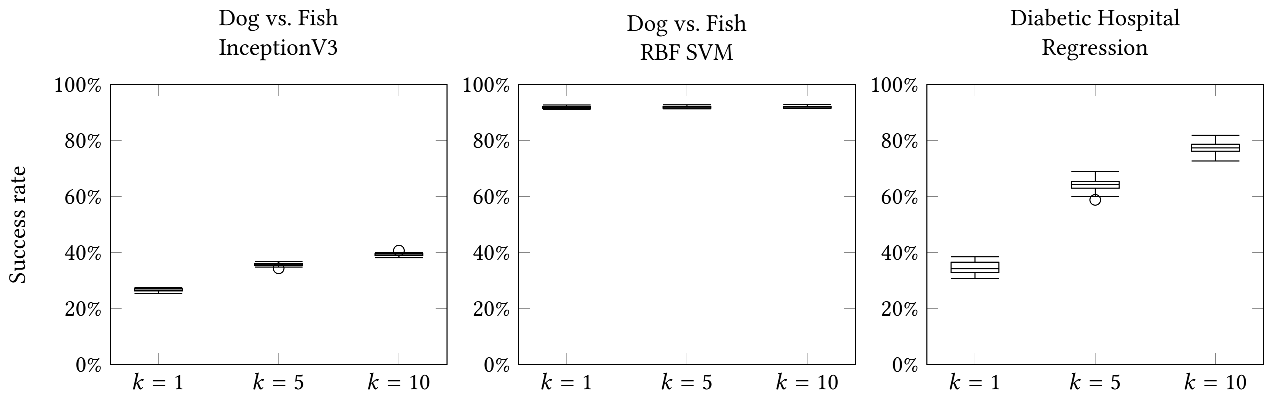


Figure 11: % of training points revealed as their own explanation, when $k \in \{1, 5, 10\}$ most influential points are revealed.

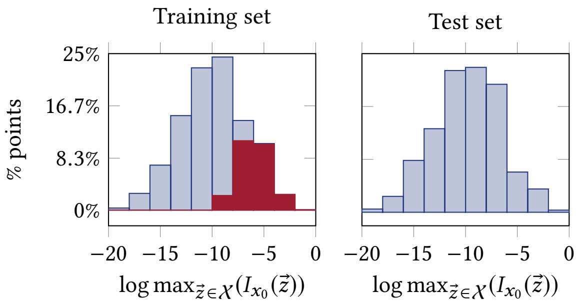


Figure 12: Histogram of the influence of the most influential points for every point in the training set(left) and test set (right) on a logarithmic scale. The points in the training set for which the membership inference is successful (i.e. $\vec{x}_0 = \text{argmax}_{\vec{z} \in \mathcal{X}}(I_{x_0}(\vec{z}))$) are highlighted in red.

all models we consider, at least 25% of the training points are the most influential points for their own prediction. For RBF SVM this figure spikes to 90%. When the top 5 (10) most influential points are released an attacker would be able to confirm 36% (39%) of the training set members for the inception model and 64% (79%) for the regression model.

The susceptibility to disclosure strongly depends on the target model, rather than the underlying dataset (e.g. SVM is especially vulnerable). A closer analysis shows that for a large majority (80%) of training points that are *not* revealed, the Koh and Liang method outputs *no datapoints*; the remaining unrevealed training points exhibited very low influences on the revealed points, suggesting approximation errors due to smooth hinge approximation or the numerical Hessian inversion. In other words, most training points that were not exposed by the Koh and Liang method were not offered an explanation to begin with!

The influence of the most influential points is similarly distributed between training and test points (assuming a normal distribution, the KL-divergence between the two distributions is 0.0007); however, the distribution is vastly different once we ignore the revealed training points. Figure 12 illustrates this for the Dog vs. Fish dataset with the inception model, but similar results hold for the hospital dataset. An attacker can exploit these differences, using techniques similar to those discussed in Section 5; we focus on other attack models in following sections.

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

	#points	$k = 1$	$k = 5$	$k = 10$
Whole data set	2400	26%	36%	39%
Clownfish	26	27%	37%	43%
Lion fish	29	9%	42%	51%
Birds	15	64%	85%	90%

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

	% of data	$k = 1$	$k = 5$	$k = 10$
Whole data set	100%	34%	64%	77%
Age 0 -10	<0.1%	67%	100%	100%
Age 0 -20	<1%	20%	58%	92%
Caucasian	74%	34%	64%	77%
African American	19%	38%	68%	81%
Hispanics	2%	39%	64%	76%
Unknown race	1%	35%	60%	77%
Asian American	<1%	25%	64%	89%

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

Minority and Outlier Vulnerability to Inference Attacks. Before turning our attention to dataset reconstruction attacks, we highlight an interesting finding. Visual inspection of images for which membership attacks were successful indicates that outliers and minorities are more susceptible of 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). 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 lion fish with $k = 1$, 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 twice 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.

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 [6]; however, such memorization leaves minority populations vulnerable to privacy risks.

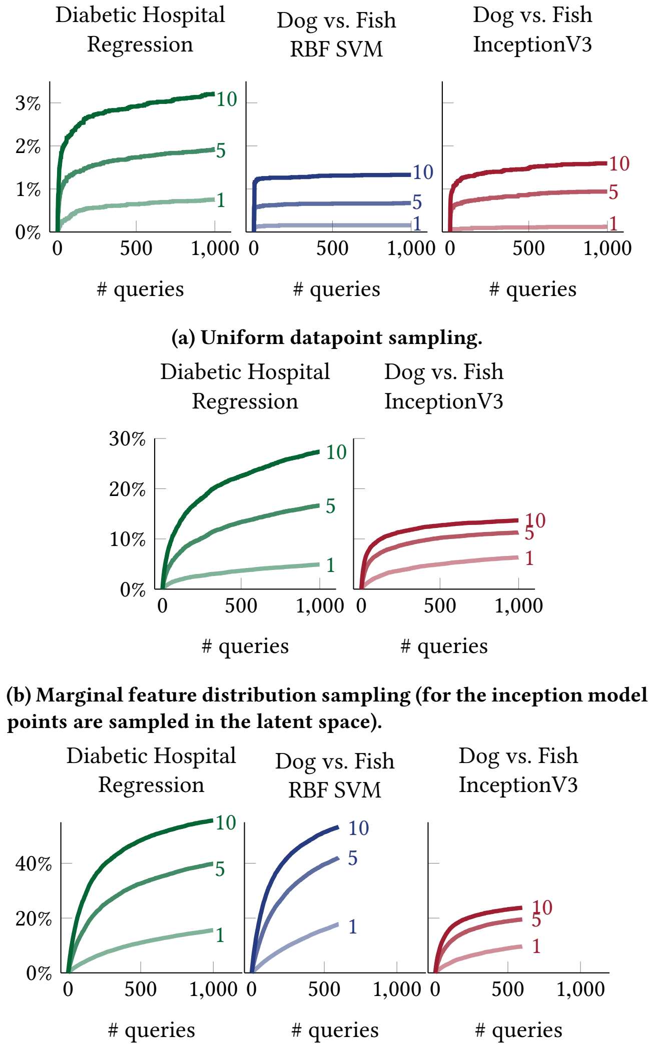
6.3 Naive Dataset Reconstruction Attacks

Record-based explanations can be exploited to conduct membership inference attacks with relative ease; let us set our sights on a more ambitious attack model, dataset reconstruction attacks: rather than recovering single training datapoints, an attacker tries to recover entire portions of the training data. A *naive* 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.

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 13a): even after observing 1 000 queries with ten training points revealed per transparency query, less than 2% of the dog/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 data distribution. Note that 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 13b shows results for the hospital dataset, and the dog/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, essentially exhausting the attacker's capacity to reveal 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 data. Using the actual data distribution, we can recover significant portions of the training data (Figure 13c). We again observe a saturation effect, where additional queries would not improve the results significantly.



(a) Uniform datapoint sampling.

(b) Marginal feature distribution sampling (for the inception model points are sampled in the latent space).

(c) True point distribution sampling.

Figure 13: % of training data revealed by an attacker using different sampling techniques, with $k \in \{1, 5, 10\}$ explanation points revealed per query.

6.4 Influence Graphs

Before considering more sophisticated dataset reconstruction attacks, we discuss a structure that naturally arises when studying influence in 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 indicates how easy it is to 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 especially interesting:
Number of strongly connected components (SCCs): a high number of SCCs implies that the training set is harder to recover:

	Diabetic Hospital	Dog vs. Fish	Dog vs. Fish
	Regression	InceptionV3	RBF SVM
#SCC	1426	1473	1090
#SCC of size 1	1295	1459	1042
Largest SCC size	173	300	646
Sum largest SCC	262	320	664
Max in-degree	258	407	125
#node degree=0	711	1154	145

Table 3: Some key characteristics of the influence graphs induced by the record-based explanations (for $k = 5$).

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.

The influence graphs induced by record-based explanations tend to have many small SCCs (see Table 3); however, each graph has one large SCC containing a considerable fraction (10% - 30%) of the training data. Furthermor, most nodes in the graph have outgoing edges to the large SCC, thus an attacker will almost surely discover the large SCC, and subsequently recover all points contained in it. However, a significant amount of the nodes has an in-degree of 0; these nodes are not influential for any point and will likely never be revealed to an attacker.

6.5 Advanced Dataset Reconstruction Attacks

We consider two advanced attack models for dataset recovery. First,we consider how an attacker can use recovered points to extract additional points from the training set, using a minimal number of queries (multiple model queries may be costly, or raise suspicions). Next, we study how knowledge about the target model can be exploited to improve attack accuracy.

Adaptive attacks. Using points in the input space allows the attacker to recover a relatively small part of the training set (Section 6.3); the attacker can use previously recovered training points as queries to effectively recover additional points (as discussed in Section 6.4, the attacker is simply traversing the influence graph). Attackers using recovered training points as queries can effectively recover the largest strongly connected component (SCC). In order to benchmark the performance of our attacker, who has no knowledge of the influence graph structure, we would like to compare it to an omniscient attacker who knows the graph structure and is able to optimally recover the SCC; however, this problem is known to be NP-complete (the best known approximation factor is 92) [32]. We thus compare our approach to a greedy omniscient attacker, which selects the node that is connected to the most unknown points. Our attacker explores the influence graph by DFS search. Figure 14 shows the gap (for a typical example) between the greedy algorithm and DFS; other traversal techniques (BFS, random walk, heursitics based on influence or the features of recovered points), have similar performance. A greedy omniscient attacker requires less than half the number of queries our current attack model does, leaving room for future improvement.

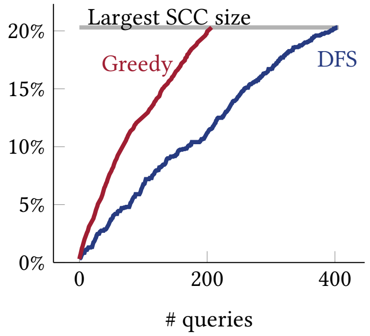


Figure 14: Recovering the largest SCC of the influence graph via an adaptive attack.

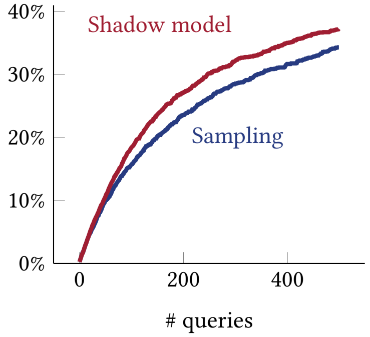


Figure 15: A shadow model-based dataset recovery attack, compared to randomly sampling points.

Shadow Models. An attacker with access to points from the same distribution as the training set might also have knowledge of the target model. This knowledge can be used to train a *shadow model* [27]: the shadow model has the same architecture as the target model, and is trained on similar data. After training a shadow model the attacker can construct a shadow influence graph i.e. the influence graph of the shadow model, and base its query strategy on it. Shadow models offer a marginally better attack model than randomly sampling points used to train the shadow model (Figure 15).

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, 8, 10, 24, 31]. Datta et al. [10] 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. [19] 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.

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 [37], but also LIME [24] which trains a simpler model with high local fidelity and QII [10] 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 will focus on the former group leaving the latter for future study.

The attack scenario we adopt has been recently proposed by Shokri et al. [27]. Shokri et al. [27] use model predictions for data with known membership to train classifiers that predict training set membership with high accuracy. However, Shokri et al. [27] assume access to the full probabilistic prediction of the model over the datapoints, rather than the deterministic assigned label; we assume a more realistic scenario, where one has access to the datapoint labels, and a given transparency report. Further, our attack doesn't require the training of a neural network and requires only the 1-norm of the explanation as input.

Our analysis 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 [36].

There exists some work on the defense against privacy leakage. Nasr et al. [20] use adversarial regularization, while Papernot et al. [22] and [23] 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 both record and feature-based explanations can be successfully exploited by an attacker to infer membership of the training set. For record-based explanations we were able to 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; what's worse, minority populations face a far greater risk of being exposed by transparency-based membership inference. 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.

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: small SCCs in the influence graph seem to be relatively safe, but it is not entirely clear whether this is a sufficient condition for safety from membership inference. It is possible to train 'safe' models with small SCCs in the influence graph; however, this will result in a drop in model accuracy that needs to be 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.

REFERENCES

[1] 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.*

[2] 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

[3] 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

[4] 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

[5] 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

[6] 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

[7] R D Cook and S Weisberg. 1982. *Residuals and influence in regression.* New York: Chapman and Hall.

[8] 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).*

[9] 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).*

[10] 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).*

[11] 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.

[12] B Goodman and S Flaxman. 2017. European Union regulations on algorithmic decision-making and a âĂIJright to explanationâĂİ. *AI Magazine* 38, 3 (2017), 50–57.

[13] D C Hoaglin. 2003. John W. Tukey and data analysis. *Statist. Sci.* (2003).

[14] 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.

[15] 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

[16] 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).*

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

[18] 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.

[19] 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

[20] 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

[21] C O'Neil. 2017. *Weapons of math destruction: How big data increases inequality and threatens democracy.* Broadway Books.

[22] 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

[23] 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

[24] 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

[25] 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

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

[27] 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

[28] 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

[29] 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

[30] 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

[31] 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

[32] R. Solis-Oba, P. Bonsma, and S. Lowski. 2017. A 2-Approximation Algorithm for Finding a Spanning Tree with Maximum Number of Leaves. *Algorithmica* 77, 2 (2017), 374–388. https://doi.org/10.1007/s00453-015-0080-0

[33] 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

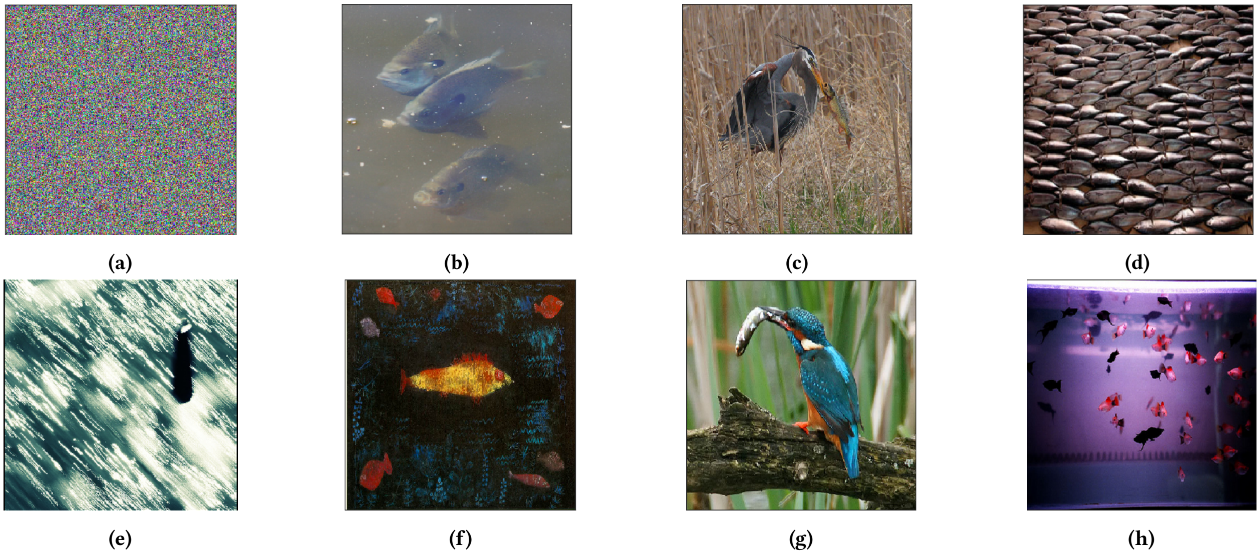
[34] M Sundararajan, A Taly, and Q Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML).*

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* https://doi.org/10.1109/CVPR.2016.308 arXiv:1512.00567

[36] 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

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

A IMAGES RECONSTRUCTED BY UNIFORM SAMPLING



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 16: (a) A uniform random input (b) The most influential point on any (uniform) input for RBF (c)-(h) Points that are at least once the most influential point for the Inception for uniform random inputs.

B INFLUENCE GRAPHS

In what follows we show two pictures of influence graphs. Blue edges are between pictures labeled as fish, red between dog pictures and purple in between classes. The size of a picture corresponds to the relative number of in-going edges. Nodes without in-going edges are not displayed. Due to file size limits smaller images are displayed as circles.

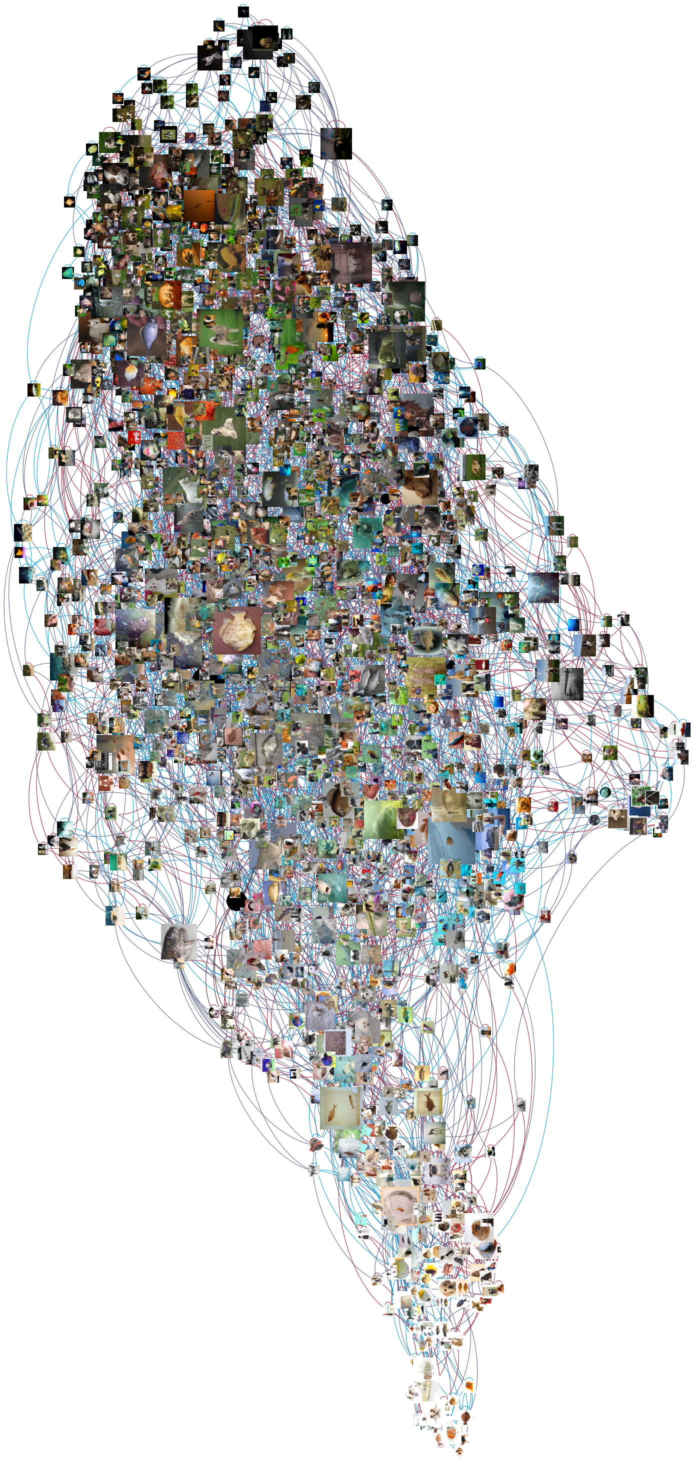


Figure 17: The influence graph for the RBF SVM for $k = 5$. The two classes strongly intertwined. The prominent color of each picture impacts the location in the graph from dark pictures on the top to white pictures on the bottom.

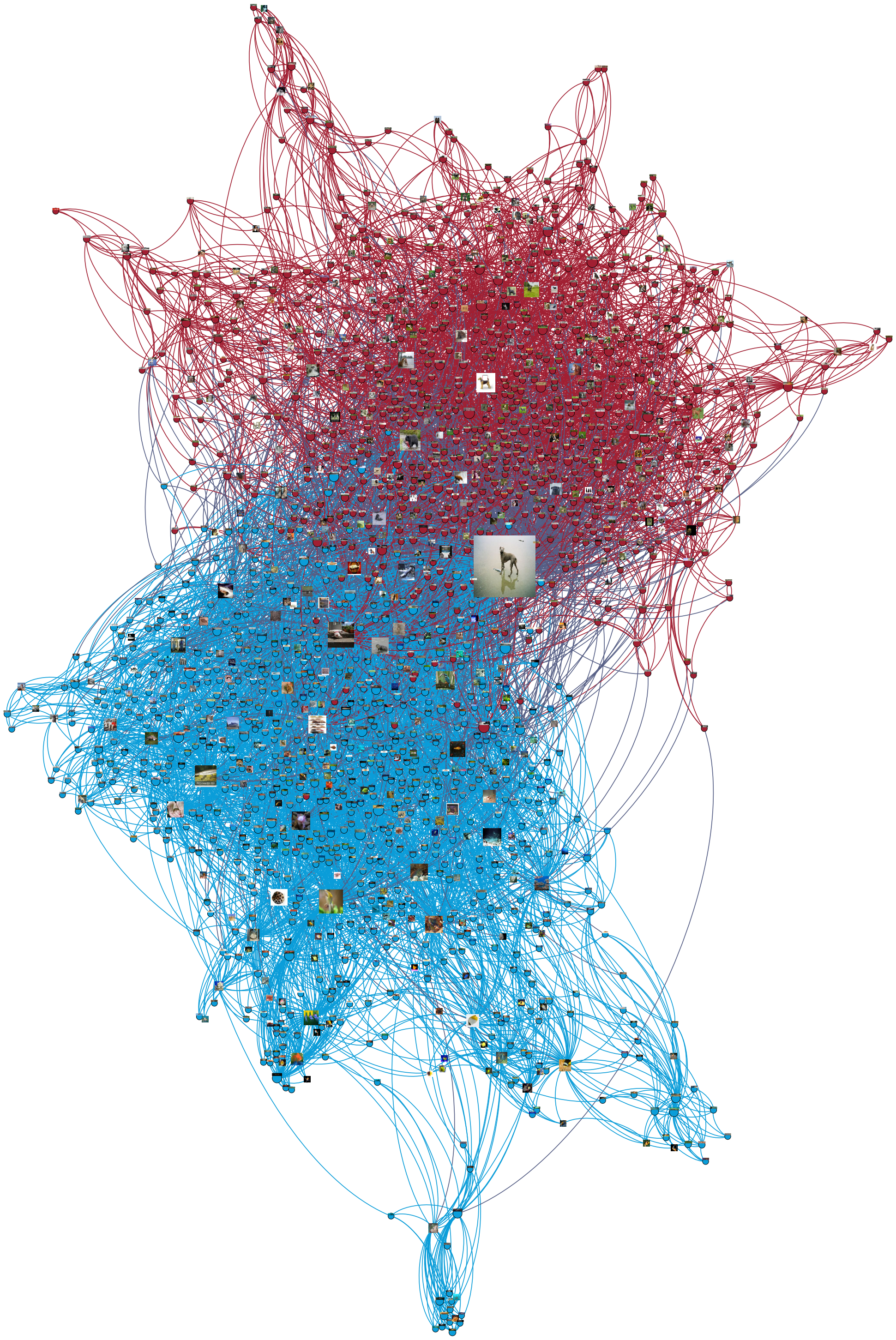


Figure 18: The influence graph for the inception-v3 model for $k = 5$. The two classes are clearly separated.