Auditing Black-box Models for Indirect Influence

Philip Adler*, Casey Falk*, Sorelle A. Friedler*, Gabriel Rybeck*,
Carlos Scheidegger[†], Brandon Smith*, and Suresh Venkatasubramanian[‡]
* Dept. of Computer Science, Haverford College, Haverford, PA, USA
padler1@haverford.edu, caseyfalk94@gmail.com, sorelle@cs.haverford.edu, grybeck@gmail.com, bsmith8108@gmail.com
[†] Dept. of Computer Science, University of Arizona, Tucson, AZ, USA
cscheid@cscheid.net
[‡] Dept. of Computer Science, University of Utah, Salt Lake City, UT, USA
suresh@cs.utah.edu

***Abstract*—Data-trained predictive models see widespread use, but for the most part they are used as *black boxes* which output a prediction or score. It is therefore hard to acquire a deeper understanding of model behavior, and in particular how different features influence the model prediction. This is important when interpreting the behavior of complex models, or asserting that certain problematic attributes (like race or gender) are *not* unduly influencing decisions.**

In this paper, we present a technique for *auditing* black-box models, which lets us study the extent to which existing models take advantage of particular features in the dataset, without knowing how the models work. Our work focuses on the problem of *indirect influence*: how some features might indirectly influence outcomes via other, related features. As a result, we can find attribute influences even in cases where, upon further direct examination of the model, *the attribute is not referred to by the model at all.*

Our approach does not require the black-box model to be retrained. This is important if (for example) the model is only accessible via an API, and contrasts our work with other methods that investigate feature influence like feature selection. We present experimental evidence for the effectiveness of our procedure using a variety of publicly available datasets and models. We also validate our procedure using techniques from interpretable learning and feature selection, as well as against other black-box auditing procedures.

I. Introduction

Machine-learning models now determine and control an increasing number of real-world decisions, from sentencing guidelines and parole hearings [2] to predicting the outcome of chemical experiments [20]. These models, powerful as they are, tend to also be opaque. This presents a challenge. How can we *audit* such models to understand why they make certain decisions? As conscientious model creators, we should want to know the extent to which a specific feature contributes to the accuracy of a model. As outside auditors, trying to understand a system can give us an understanding of the model's priorities and how it is influenced by certain features. This may even have legal ramifications: by law for example, decisions about hiring cannot be influenced by factors like race, gender or age.

As model creators, we could build interpretable models, either by explicitly using interpretable structures like decision trees, or by building shadow models that match model outputs in an interpretable way. In this work, we are interested in auditing a *black box* model from the outside (because the model is proprietary, accessible only through an API, or cannot be modified).

A. Direct and Indirect Influence

Much of the modern literature on black-box auditing (See Section II for details) focuses on what we call *direct* influence: how does a feature (or a group of features) directly affect the outcome? This is quantified by replacing the feature (or group) by random noise and testing how model accuracy deteriorates. In this paper, we focus on the different and more subtle issue of *indirect* influence.

Consider trying to verify that racial considerations did *not* affect an automated decision to grant a housing loan. We could use a standard auditing procedure that declares that the `race` attribute does not have an undue influence over the results returned by the algorithm. Yet this may be insufficient. In the classic case of *redlining* [17], the decision-making process explicitly excluded race, but it used `zipcode`, which in a segregated environment is strongly linked to race. Here, race had an *indirect* influence on the outcome via the zipcode, which acts as a *proxy*.

Note that in this setting, race would not be seen as having a direct influence (because removing it doesn't remove the signal that it provides). Removing both race and zipcode jointly (as some methods propose to do) reveals their combined influence, but also eliminates other task-specific value that the zipcode might signal independent of race, as well as leaving unanswered the problem of *other* features that race might exert indirect (and partial) influence through.

B. Our Work

In this paper, we study the problem of auditing black box models for indirect influence. In order to do this, we must find a way to capture information flow from one feature to another. We take a learning-theoretic perspective on this problem, which can be summarized via the principle, first enunciated in [10] in the context of certifying and removing bias in classifiers:

> *the information content of a feature can be estimated by trying to* predict *it from the remaining features.*

This research was funded in part by the NSF under grants IIS-1251049, CNS-1302688, IIS-1513651, DMR-1307801, IIS-1633724, and IIS-1633387.

How does this allow us to correctly quantify the influence of a feature in the presence of proxies? Let us minimally modify the data so that the feature can no longer be predicted from the remaining data. The above principle then argues that we have fully eliminated the influence of this feature, both directly and in any proxy variables that might exist. If we now test our model with this *obscured* data set, any resulting drop in prediction accuracy can be attributed directly to information from the eliminated feature.

Our main contributions in this work include

- A technique to *obscure* (fully and partially) the influence of a feature on an outcome, and a theoretical justification of this approach.
- A method for quantifying indirect influence based on a differential analysis of feature influence before and after obscuring.
- An experimental validation of our approach on a number of public data sets, using a variety of models.

II. Conceptual Context

Work on black-box auditing intersects with a number of related areas in machine learning (and computer science at large), including directions coming from privacy preservation, security, interpretability, and feature selection. We tease out these connections in more detail in Section VII.

Here we outline how we see our work in the specific context of the prior literature on auditing black box machine learning models. Modern developments in this area can be traced back to Breiman's work on random forests [4], and we highlight two specific recent related works. Henelius et al. [13] propose looking at variable sets and their influence by studying the consistency of a black-box predictor when compared to a random permutation of a set. Datta et al. [8] provide a generalization of this idea, linking it to game-theoretic notions of influence and showing that different choices of probability spaces and random variables yield a number of different interesting auditing measures. These two papers fundamentally hinge on the notion of associating each input value with an *intervention distribution*. These intervention distributions can be easily shown (in distribution) to obscure attributes. Our work, on the other hand, will audit black boxes by providing, for any given input point, an intervention that is *deterministic*, while guaranteeing (in some settings, see Section IV-C) that the attributes are still obscured over the entire distribution. Our intervention preserves more of the signal in the dataset, and – crucially in some settings – naturally preserves indirect influences of proxy variables. As we will show in the experiments (see Section V-D), the technique of Henelius et al. cannot detect proxy variables, and although Datta et al. can use some of their measures to detect proxy variables, their attribute rankings generally do not reflect the proxy relationships.

Our methods draw heavily on ideas from the area of *algorithmic fairness*. The process by which we eliminate the influence of a feature uses ideas from earlier work on testing for disparate impact [10]. Again, a key difference is that we no longer have the ability to retrain the model, and we *quantify* the influence of a feature rather than merely eliminating its influence.

III. Indirect Influence

Let $f: \mathbb{X} \rightarrow \mathbb{Y}$ be a black-box classification function, where $\mathbb{X} \subset \mathbb{X}^{(1)} \times \mathbb{X}^{(2)} \ldots \mathbb{X}^{(d)}$ is a d-dimensional feature space with the i^{th} coordinate drawn from the domain $\mathbb{X}^{(i)}$ and $\mathbb{Y}$ is the domain of outcomes. For example, $\mathbb{X} = \mathbb{R}^d$ and $\mathbb{Y} = \{-1, 1\}$ for binary classification on Euclidean vectors. Fix a data set $(X, Y) = \{(X_i, y_i)\} \subset \mathbb{X} \times \mathbb{Y}, 1 \leq i \leq n$ and let $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, where $x_{ij} \in \mathbb{X}_j$ for all i. We denote the accuracy of prediction as $\texttt{acc}(X, Y, f)$. For example, $\texttt{acc}(X, Y, f) = \frac{1}{n} \sum \mathbf{1}_{y_i \neq f(X_i)}$ is the standard misclassification error.

We wish to quantify the indirect influence of a feature j on the outcome of classification. The typical approach to doing this in the auditing literature is to perturb the j^{th} feature x_{ij} of each X_i in some way (usually by random perturbation), obtaining a modified data set X_{-j}. Then the influence of j can be quantified by measuring the difference between $\texttt{acc}(X, Y, f)$ and $\texttt{acc}(X_{-j}, Y, f)$ (note that f is not retrained on X_{-j}).

Unfortunately, randomly perturbing features can disrupt indirect influence in a number of ways. Firstly, random perturbations could also remove useful task-related information in proxy features that would degrade the quality of classification. Secondly, this prevents us from cleanly quantifying the *relative* effect of the feature being perturbed on related proxy variables.

We propose a different approach. We will still perturb a data set X to eliminate the direct and indirect influence of feature j, and measure the change in accuracy of prediction as before. However, we will do this perturbation in a directed and deterministic manner, organized around the question: "*can we predict the value of feature j from the remaining features*?" Intuitively, if we cannot, then we know that we have correctly eliminated the influence of j. Moreover, if we do this perturbation "minimally," then we have changed the data as little as possible in the process. We will say in this case that we have *removed* feature j from the data set and have *obscured* its influence on X.

A. Obscuring data with respect to a feature

We start by defining the error measure we will use to test predictability. Rather than the standard misclassification rate, we will use the well-known *balanced error rate* measure that is more sensitive to class imbalance. This is important if a feature has significant skew in the values it takes. Let $\text{supp}(Y) = \{y \in \mathbb{Y} | y \in Y\}$ be the set of elements of $\mathbb{Y}$ that appear in the data.

Definition III.1 (BER). *Let $f : \mathbb{X} \rightarrow \mathbb{Y}$ be a classifier, and let $(X, Y) = \{(X_i, y_i)\}$ be a set of examples. The* balanced error rate BER *of f on (X, Y) is the (unweighted) average class-conditioned error of f:*

$$\text{BER}(X, Y, f) = \frac{1}{|supp(Y)|} \left(\sum_{j \in supp(Y)} \frac{\sum_{y_i = j} \mathbf{1}_{f(X_i) \neq j}}{|\{i \mid y_i = j\}|} \right)$$

A feature i has been removed from a data set if we can no longer predict that feature from the remaining data. This motivates the following definition. Let $X^{(i)} = (x_{1i}, x_{2i}, \ldots, x_{ni})$ denote the column corresponding to the i^{th} feature.

Definition III.2 (ϵ-obscure)**.** *We define $X \setminus_\epsilon \mathbb{X}_i$ as the* ϵ-obscure *version of X with respect to feature $\mathbb{X}_i$ if $X^{(i)}$ cannot be predicted from $X \setminus_\epsilon X_i$. I.e., if, for all functions $f : \mathbb{X} \setminus \mathbb{X}_i \to \mathbb{X}_i$,*

$$\text{BER}(X \setminus_\epsilon \mathbb{X}_i, X^{(i)}, f) > \epsilon$$

We can now define a measure of influence for a feature.

Definition III.3 ((indirect) influence)**.** *The* indirect influence *II(i) of a feature i on a classifier f applied to data (X, Y) is the difference in accuracy when f is run on X versus when it is run on $X \setminus_\epsilon \mathbb{X}_i$:*

$$II(i) = acc(X, Y, f) - acc(X \setminus_\epsilon \mathbb{X}_i, Y, f)$$

Notes The definition of obscurity we use here is adapted from [10], but applied to any feature, rather than just "protected" ones. In what follows, we will typically treat ϵ as large (say above 0.5 for binary classification).

IV. Computing Influence

In this section, we will introduce a method we call *gradient feature auditing* (GFA) to estimate indirect influence. Using it, we compute the influence of each feature of the data and order the features based on their influence. This GFA algorithm works feature-by-feature: in order to compute $X \setminus_\epsilon \mathbb{X}_i$, we apply an *obscuring procedure* to each feature $j \neq i$ in $\mathbb{X}$. Because we operate one feature at a time, we cannot guarantee that all influence can be removed (and therefore estimated). However, we will show that the feature-level procedure is theoretically sound and in fact generalizes the standard ANOVA test for null hypothesis testing.

Let us start with a simple case: when the feature $W = \mathbb{X}_j$ to be obscured is numerical, and the feature $O = \mathbb{X}_i$ we are removing is categorical. Let $W_x = \Pr(W \mid O = x)$ denote the marginal distribution on W conditioned on $O = x$ and let the cumulative distribution be $F_x(w) = \Pr(W \geq w \mid O = x)$.

Define the *median* distribution A such that its cumulative distribution F_A is given by $F_A^{-1}(u) = \text{median}_{x \in O} F_x^{-1}(u)$. In [10] it was shown that if we modify the distribution of W to match A by "moving" values of W so as to mimic the distribution given by A, then O is maximally obscured, but W also minimally changes, in that A also minimizes the function $\sum_{x \in O} d(W_x, A)$ where $d(\cdot, \cdot)$ was the earthmover distance [23] between the distributions using ℓ_2 as the base metric. We call this procedure `ObscureNumerical`.

This procedure does not work if features to be obscured and removed are not numerical and categorical respectively. We now describe procedures to address this issue.

A. Removing numerical features

In order to remove a numerical feature, we must first determine what aspects of the number itself should be removed. In an optimal setting, we might remove the entirety of the number by considering its binary expansion and ensuring that no bit was recoverable. However, when working with most numerical data, we can safely assume that only the higher order bits of a number should be removed. For example, when considering measurements of scientific phenomena, the lower order bits are often measurement error.

Thus, we bin the numerical feature and use the bins as categorical labels in the previously described obscuring procedure. Bins are chosen using the Freedman-Diaconis rule for choosing histogram bin sizes [11].

B. Obscuring categorical features

Our procedure relies on being able to compute cumulative density functions for the feature W being obscured. If it is categorical, we no longer have an ordered domain on which to define the cumulative distributions F_w. However, we do have a base metric: the exact metric $\mathbf{1}$ where $\mathbf{1}(x, w) = 1 \iff x = w$. We can therefore define A as before, as the distribution minimizing the function $\sum_{x \in O} d(W_x, A)$. We observe that the earthmover distance between any two distributions over the exact metric has a particularly simple form. Let $p(w), q(w), w \in W, \sum p(w) = \sum q(w) = 1$ be two distributions over W. Then the earthmover distance between p and q with respect to the exact metric $\mathbf{1}$ is given by $d(p, q) = \|p - q\|_1$ Therefore the desired minimizer A can be found by taking a component-wise median for each value w. In other words, A is the distribution such that $p_A(w) = \text{median}_w W_x(w)$. Once such an A has been computed, we can find the exact repair by computing the earthmover distance (via min-cost flows)[1] between each W_x and A. This results in fewer changes than merely changing values arbitrarily.

We must create the obscured version of W, denoted $\hat{W}$. Let $\hat{W}_{w,x}$ be the partition of $\hat{W}$ where the value of the obscured feature is w and the value of the removed feature is x. We must create $\hat{W}$ so as to ensure that $|\{\hat{W} | O = x\}| \in \mathbb{Z}$ for all values of $x \in O$. We set $|\hat{W}_{w,x}| = \lfloor p_A(w) \cdot |\{W | O = x\}| \rfloor$. Letting $d(w) = |\hat{W}_{w,x}|$ for all $w \in W$ gives the node demands for the circulation problem between W_x and A, where supplies are set to the original counts $d(w) = -|W_{w,x}|$. Since $|\hat{W}_{w,x}| \leq |W_{w,x}|$, an additional *lost observations node* with demand $|W_{w,x}| - |\hat{W}_{w,x}|$ is also added. The flow solution describes how to distribute observations at a per category, per obscured feature value level. Individual observations within these per category, per feature buckets can be distributed arbitrarily. The observations that flow to the lost observations node are distributed randomly according to distribution A. We call this procedure `ObscureCategorical`.

Using the categorical or numerical obscuring procedure appropriately depending on the data yields our procedure for computing $X \setminus_\epsilon \mathbb{X}_i$.

Notes. This description assumes that we want to remove *all* effects of a variable in order to measure its influence. However, how the influence changes as we remove its effect is also

[1] This is a straightforward application of the standard min-cost flow problem; we defer a detailed description to the full version of this paper.

interesting. Indeed, this is why we we refer to the overall process as a *gradient* feature audit. To that end, all of the algorithms above can be adapted to allow for a *partial* removal of a variable. On a scale of 0 to 1 where 0 represents the original data, and 1 represents a full removal, we can remove a *fractional* part of the influence by allowing the individual conditional distributions to move *partly* towards each other.

While the process above produces a score, the induced ranking is also useful, especially when we compare our results to those produced by other auditing methods, where the score itself might not be directly meaningful. We illustrate this further in Section V.

C. Obscuring, ANOVA, and the F-test

We now provide a theoretical justification for our obscuring procedure. Specifically, we show that if the feature W being obscured has Gaussian conditionals, then our obscuring procedure will create a data set on which the F-test [6] will fail, which means that the null hypothesis (that the conditionals are now identical) will not be invalidated. Thus, our procedure can be viewed as a generalization of ANOVA.

Consider a data set D consisting of samples (x, y), where y is a class label that takes the values $-1, 1$ and the x are drawn from univariate Gaussians with different means and a shared variance. Specifically, $\Pr(x|y = i) = \mathcal{N}(v_i, \sigma^2)$. We assume the classes are balanced: $\Pr(y = -1) = \Pr(y = 1)$.

Let $\tilde{x}$ be the obscured version of x. It is easy to show that[2], in this case, the obscuring procedure will produce values following the distribution $p(\tilde{x}|y = i) = \mathcal{N}(1/2(v_{-1} + v_1), \sigma^2)$.

We apply the F-test to see if we can tell apart the two conditional distributions for the two different values of y. Let $S_y = \{x \mid (x, y) \in D\}$. The test statistic F is the ratio of the between-group sample variance and the in-group sample variance. Set $\mu = (v_{-1}+v_1)/2$ and $\delta = (\mu - v_{-1})^2 = (\mu - v_1)^2$. Thus

$$F = \frac{n\delta}{(1/2)\sum_{x\in S_{-1}}(x - v_{-1})^2 + \sum_{x\in S_1}(x - v_1)^2}$$

We note that $\sum_{x\in S_1}(x - v_1)^2$ has expectation $n/2\sigma^2$, (and so does the corresponding expression related to S_{-1}). Using the plug-in principle, we arrive at an estimator for the F statistic: $F = \frac{\delta}{\sigma^2}$ This is the traditional expression for a two-variable, one-way ANOVA: δ is a measure of the variance that is explained by the group, and σ^2 is the unexplained variance.

We apply this to the obscured distribution. From the remarks above, we know that the conditional distributions for $y = 0, 1$ are identical Gaussians $N(\mu, \sigma)$. We now show that the positive parameter δ is concentrated near zero as the number of samples increases.

Let $x_1, \ldots, x_n$ be samples drawn from the conditional distribution for $y = 0$, and similarly let $y_1, \ldots, y_n$ be drawn from the distribution conditioned on $y = 1$. Set $X = \frac{1}{n}\sum x_i$ and $Y = \frac{1}{n}\sum y_i$. Note that $E[X] = E[Y] = \mu$, and so $E[X - Y] = 0$.

Let $\hat{\delta} = (X - Y)^2$. We first observe that

$$\begin{aligned}|X - Y| &\le |X - E[X]| + |E[X] - E[Y]| + |Y - E[Y]| \\ &= 2|X - E[X]|\end{aligned}$$

because X and Y are identically distributed. Therefore,

$$\begin{aligned}\Pr(\hat{\delta} \ge \epsilon^2) &\le 4\Pr(|X - E[X]|^2 \ge \epsilon^2) \\ &\le 4\exp(-n\epsilon^2/2\sigma^2)\end{aligned}$$

by standard Hoeffding tail bounds [18]. Therefore, with $\log n/\epsilon^2$ samples, the F-test statistic is with high probability less than ϵ^2, and thus the null hypothesis (that the distributions are identical) will not be invalidated (which is equivalent to saying that the test cannot distinguish the two conditional distributions).

V. Experiments

In order to evaluate the introduced gradient feature auditing (GFA) algorithm, we consider experiments on five data sets, chosen to balance easy replicability with demonstration on domains where these techniques are of practical interest. Datasets and GFA code are available online.[3]

Synthetic data. We generated $6,000$ items with $3,000$ assigned to each of two classes. Items have five features. Three features directly encode the row number i: feature A is i, B is $2i$, and C is $-i$. There is also a random feature and a constant feature. We use a random $\frac{2}{3} : \frac{1}{3}$ training-test split.

Adult Income and German Credit data. We consider two commonly used data sets from the UC Irvine machine learning repository[4]. The first is the Adult Income data set consisting of 48,842 people, each with 14 descriptive attributes from the US census and a classification of that person as making more or less than $50,000 per year. We use the training / test split given in the original data. The second is the German Credit data set consisting of 1000 people, each with 20 descriptive attributes and a classification as having good or bad credit. We use a random $\frac{2}{3} : \frac{1}{3}$ training-test split on this data and the two data sets described below.

Recidivism data. The Recidivism Prediction data set is taken from the National Archive of Criminal Justice Data[5] and contains information on 38,624 prisoners sampled from those released from 15 states in 1994 and tracked for three years. The full data includes attributes describing the entirety of the prisoners' criminal histories before and after release as well as demographic data. For this work, we processed the data additionally following the description of the processing performed in a previous study on interpretable models [29]. This processing resulted in 10 categorical attributes. The prediction problem considered is that of determining whether a released prisoner will be rearrested within three years.

[2] This follows from the fact that the earthmover distance between two distributions on the line is the ℓ_1 difference between their cumulative density functions. In this case it means that the earthmover distance is precisely the distance between the means.

[3] https://github.com/cfalk/BlackBoxAuditing

[4] https://archive.ics.uci.edu/ml/datasets.html

[5] http://doi.org/10.3886/ICPSR03355.v8

Dark Reactions data. The Dark Reactions data [20] is a set of $3,955$ historical hydrothermal synthesis experiments aiming to produce inorganic-organic hybrid materials. 273 attributes indicate aggregates of atomic and ionic properties of the inorganic components and semi-empirical properties of the organic components. The classification variable indicates whether the reaction produced an ionic crystal.

Models. We built two models that are notoriously opaque to simple examination; SVMs[6] [12] and feedforward neural networks (FNNs).[7] We also include C4.5 decision trees[8] [19] so that the audited results can be examined directly in comparison to the models themselves.

The FNN on the synthetic data was trained using a `softmax` input-layer for 100 epochs with a batch-size of 500 and a learning-rate of 0.01; no hidden layer was involved. The Adult Income FNN model was trained using a single `softmax` input-layer for 1000 epochs using a batch-size of 300 and a learning-rate of 0.01; no hidden layer was involved. The German Credit data FNN model was trained similarly to the Adult model, except using a batch-size of 700. The Dark Reaction data FNN model was trained using `tanh` activations for the input layer and `softmax` activations in a fully connected hidden layer of 50 nodes; it was trained using a batch-size of 300 for 1500 epochs with a modified learning rate of 0.001. The Recidivism data set FNN was trained using `softmax` activations on a fully connected hidden layer of 100 nodes. The model was trained using a batch size of 500 and 100 epochs.

Auditing using test data. Our claim in this work is that we can obscure a data set with respect to a feature in order to test its influence. To test this, we could also retrain our classifier on obscured data and compare the resulting outcomes on test data. We have run this experiment on the synthetic data and found the resulting scores to be very similar, demonstrating that even though our obscuring process applies after training, it is still effective at removing the influence of a feature. We defer detailed experiments to an extended version of this work.

A. Black-box feature auditing

We now assess the performance of our GFA method. We trained each model on each of the five data sets. We then ran GFA using the test data for each data set. As we noted in Section IV, we progressively increase the degree to which we obscure a data set by removing a variable. Specifically, we used partial obscuring values at 0.1 intervals between 0 (no removal) and 1 (full removal) giving us 11 total partially obscured data sets to consider the accuracy change for. Figure 1 shows the resulting GFA plots.

Synthetic data. Beginning with the synthetic data under any of the models, we see that removing any one of the three main features (A, B, and C) that encode the outcome class causes the model to degrade to 50% accuracy as our approach would predict. Removing the constant feature has no effect on the model's accuracy, also as expected. The removal of the random feature causing the model to lose a small amount of accuracy may initially seem surprising, however this is also as expected since the random feature also individually identifies each row and so could be used to accurately train an overfitted model.

Adult income data. On the Adult income data set, we see that the ranking changes depending on the model. Recall that the more "important," highly ranked features are those that cause the accuracy to drop the most, i.e. are towards the bottom of the charts. While `race` is found to have only a small influence on all the models, the removal of `age` has a large impact on the SVM and FNN models, but is much less important to the decision tree. On the FNN model gradient auditing plot we also see a set of features which when partially removed actually *increased* the accuracy of the model. In a model that was not optimal to begin with, partially obscuring a feature may in effect reduce the noise of the feature and allow the model to perform better.

German credit data. The results on the German credit data exhibit an arbitrary or noisy ordering. We hypothesize that poor models are likely to produce such auditing results. There are two interrelated reasons why this may be the case. First, since the audit assesses the importance of a feature using the accuracy of the model, if the model is poor, the resolution of the audit is degraded. Second, if the model is poor partially due to overfitting, obscuring features could cause spurious and arbitrary responses. In these contexts, it makes more sense to consider the change in accuracy under a consistency measure. We explore this further in Section V-B.

Recidivism data. On the Recidivism data we see incredible consistency between the rankings of the different models. The top three most important features under all three models are `PRIRCAT`, a categorical representation of the number of prior arrests of the prisoner, `RLAGE`, the age at release, and `TMSRVC`, the time served before the release in 1994. The four least important features are also common to all three models: the sex of the prisoner, whether the prisoner is an alcohol or drug abuser, and the number of infractions the prisoner was disciplined for while incarcerated.

Dark Reactions data. The Dark Reactions data shows different top ranked features for the three models, though all three rankings include the minimum Pauling electronegativity and the maximum Pearson electronegativity in the top ranked cluster of features. These values are calculated for the inorganic components of the reaction and have been shown to be important for distinguishing between chemical systems in this data set [20], which this audit confirms. Features indicating the presence of elements and amounts of metal elements with specific valence counts similarly allow the models to classify chemical systems. The SVM and FNN top features include atomic properties of the inorganics that are related to the electronegativity of an element, so these proxies are correctly also highly ranked. The top ranked decision tree features additionally include the average molecular polarizability for the organic components, which was previously hypothesized

[6] Implemented using Weka's version 3.6.13 SMO: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html

[7] Implemented using TensorFlow version 0.6.0: https://www.tensorflow.org/

[8] Implemented using Weka's version 3.6.13 J48: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html

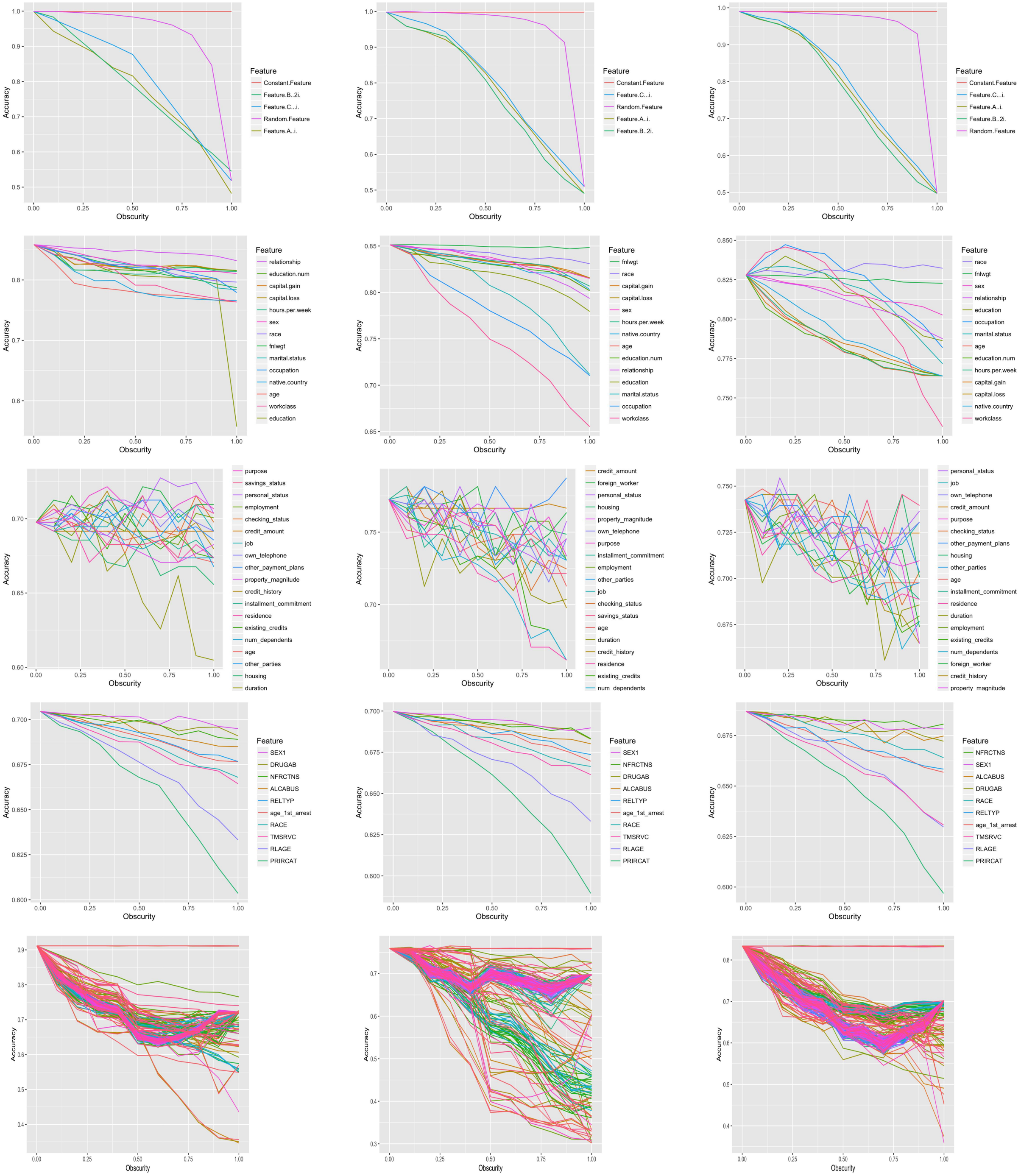

Fig. 1. Obscurity vs. accuracy plots for each model and each data set considered. First column: C4.5 decision trees. Second column: SVMs. Third column: FNNs. First row: Synthetic data. Second row: Adult income data set. Third row: German credit data. Fourth row: Recidivism data. Final row: Dark Reaction data, shown without a feature legend due to the large number of features.

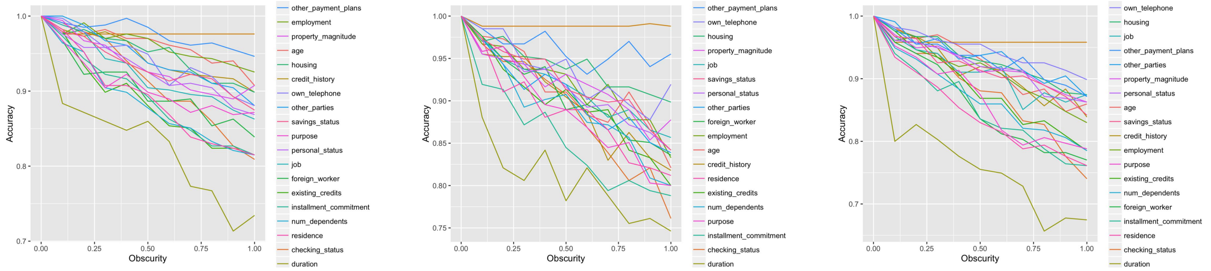

Fig. 2. Obscurity vs. consistency plots for the German Credit data. First column: decision tree model. Second column: SVM. Third column: FNN.

as important to synthesis of templated vanadium selenites explored via this data set [20]. For all three models, the lowest ranked descriptors are constants scored correctly as having no influence to the model.

Running time. Running times for these experiments, including time to train the model, time to do all partially obscured audits, and time to write the partially obscured data sets to disk, varied from 13 seconds on the synthetic data set with the C4.5 decision tree model to just over 3 hours for the Dark Reaction data set with the FNN. Since one-time audits are not highly time-sensitive tasks, and the code was unoptimized, we present these times largely as evidence of the feasibility of this method.

B. Auditing for consistency

The results in Figure 1 allow us to both create a ranking of influence as well as evaluate absolute accuracy of the model on obscured data. However, the German Credit data set yields very noisy results. To address this, we propose using *model consistency*: we replace the original labels with those predicted by the model on unobscured data, and then calculate accuracy as we progressively obscure data with respect to these labels. The unobscured data will thus always have a consistency of 100%. The new gradient measured will be the difference between the 100% consistency at obscurity of 0 and the degraded consistency when fully obscured.

As can be seen in Figure 2, under the consistency measure the accuracy of the German Credit model degrades smoothly so that a ranking can be extracted. The slight noise remaining is likely due to the final step of the categorical obscuring algorithm that redistributes "lost observations" randomly. The resulting ranking is fairly consistent across models, with `credit amount`, `checking status`, and `existing credits` ranked as the top three features in all models.

Similar to the German Credit data, the FNN model on the Adult Income data set was not an optimal model. This means that in the accuracy-based plots in Figure 1 obscuring the features at first leads, counterintuitively, to an increase in accuracy. In the consistency graph for the FNN model on the Adult Income data (see Figure 3) we see that while the ranking derived from accuracy closely resembles the consistency ranking, a cluster of features (`native.country`, `capital.loss`, `capital.gain`, and `hours.per.week`) had been ranked above `occupation` and `marital.status` and the consistency ranking moves that cluster down in rank.

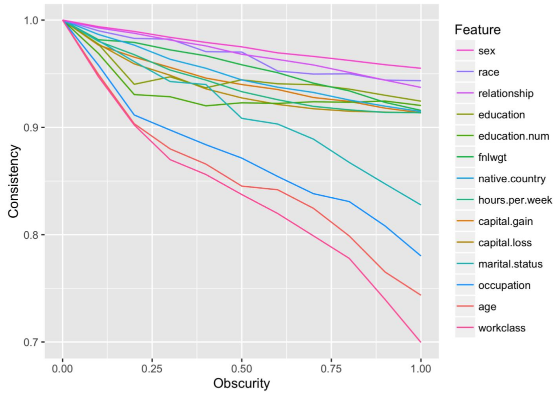

Fig. 3. Obscurity vs. consistency for Adult Income data modeled by an FNN.

C. Evaluating with respect to a direct influence audit

In order to determine if GFA is correctly determining the indirect influence of a model, we will first develop a simple method of detecting *direct* influence, and then examine the outlying attributes for which our indirect audit differs.

Direct influence audit. The first step of the direct influence audit method we will use is the creation of an interpretable model of the model. By a "model of a model" we mean that we should 1) train the model f on training data (X, Y), 2) determine new labels $\hat{Y}$ from the predicted outcomes $f(X)$, and 3) overfit an interpretable model $I(f)$ to these predicted labels (as done in [3]). (This idea is similar to model compression, but without the need to find new test data [5].) Assuming that the model resulting from this procedure has high accuracy on $\hat{Y}$, we now have an interpretable model of our model.

For the SVM and decision tree models trained on the each of the five data sets, we trained an unpruned C4.5 decision tree model of the model. With these interpretable models of a model, unfortunately a manual comparison to the feature ranking is still impossible due to the size of the resulting trees. We create feature importance scores by calculating the probability that each feature appeared on the path from the root to the leaf node containing an item from the training set. This ranking is weighted by the number of items at each leaf node. Any feature appearing at the root node, for example, will have a probability of 1 and come first in the derived ranking. This gives us the direct influence audit results we will compare to.

Synthetic data. Beginning with the simple decision tree model for the synthetic data, looking at the decision tree reveals that only feature A is used explicitly - the decision tree has a single split node. When we create a decision tree model of this model, to confirm the model of a model technique, the result is exactly the same decision tree. Creating a decision tree model of the SVM model, we find again that there is a single node splitting on feature A. Both models of models have 100% accuracy on the training set (that they were purposefully over-fit to). The probability ranking of all of these models for the synthetic data contains feature A first with a probability of 1 and all remaining features tied after it with a probability of 0 (since only feature A appears in the decision tree). Since the probability ranking is a direct audit, it is not surprising that it does not score proxy variables B and C highly.

Comparison to a direct influence audit. To evaluate the model of a model probability ranking in comparison to the GFA accuracy ranking, we compare the feature rankings on three datasets (Adult, Recidivism and German). Specifically, we collect the three generated rankings into one vector for each of the feature ranking procedures and run a Spearman rank-correlation statistical test. We find a combined sample rank correlation of 0.413 (a 95% bootstrap confidence interval of $[0.115, 0.658]$), and find also that we can reject the null hypothesis (of no correlation), with $p < 0.002$. When we compare the GFA ranking based on model *consistency* (cf. Section V-B), the results are similar to the ones based on model accuracy. This provides evidence that our feature auditing procedure closely matches a direct influence audit overall.

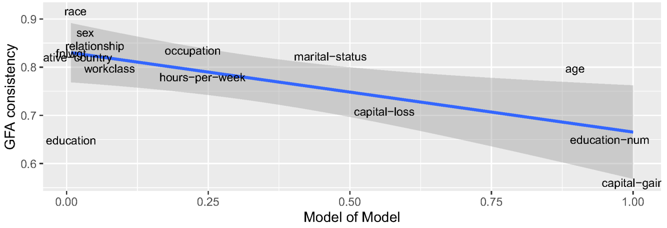

Fig. 4. Model of a model decision tree probability rankings vs. GFA consistency scores shown with a linear regression and 95% confidence interval. The outlying features contain proxy information in the data set.

To consider the cases where these rankings don't match, we look at Adult Income under the C4.5 decision tree model. As shown in Figure 4, linear regression confirms that most features have similar scores, but there are a few important outliers: `marital-status`, `education`, `race`, `age`, and `capital-gain`. We hypothesize that the information in these features can be reconstructed from the remaining attributes, and so they are scored differently under an indirect influence audit. We explore this hypothesis next.

D. Comparison to previous work

Henelius et al. [13] provide a different technique to solve the Black-box Feature Auditing problem. They focus on determining not only the influence scores associated with each feature, but also groupings of features that are more influential as a group than they are individually (i.e., have mutual influence on a model's outcomes) and use the consistency measure as the score. The results of running their algorithm[9] on the synthetic data we considered here is shown in Table I.

	C4.5 Decision Tree		SVM	
Feature	Henelius et al.	GFA	Henelius et al.	GFA
A	0.50	0.50	0.87	0.50
B	1.0	0.53	0.87	0.50
C	1.0	0.53	0.88	0.50
Random	1.0	0.96	0.99	0.97
Constant	1.0	1.0	1.0	1.0

TABLE I
Synthetic data comparison between Henelius et al. and GFA. All models achieve 100% accuracy, so GFA consistency and accuracy are the same.

These scores on the synthetic data set illuminate the key difference between the Henelius et al. work and GFA: while Henelius et al. focus on auditing for *direct influence* on a model, GFA includes both direct and *indirect influence* through proxy variables. For example, the C4.5 decision tree created on the synthetic data is a very simple tree containing one node that splits on the value of feature A. Henelius et al. correctly show that A is the only feature directly used by the model. However, GFA additionally shows that features B and C are proxies for A (recall that B is defined as two times feature A and C is defined as negative one times feature A).

	C4.5 Audit Scores			Feature predictability	
Feature	Henelius	GFA	GFA	REPTree	C4.5 or
	et al.	cons.	acc.		M5
capital-gain	0.94	0.56	0.55	0.16	0.21
education	0.98	0.65	0.65	1.0	1.0
education-num	0.92	0.65	0.65	1.0	1.0
capital-loss	0.98	0.71	0.68	0.09	0.16
hrs-per-week	0.96	0.78	0.74	0.44	0.49
age	0.94	0.80	0.76	0.65	0.68
workclass	0.98	0.80	0.76	0.11	0.16
fnlwgt	0.99	0.83	0.77	0.15	0.22
marital-status	0.87	0.82	0.77	0.74	0.76
native-country	1.0	0.82	0.78	0.22	0.28
occupation	0.90	0.84	0.79	0.21	0.17
relationship	0.99	0.84	0.81	0.69	0.70
sex	1.0	0.87	0.82	0.62	0.62
race	1.0	0.92	0.83	0.25	0.23

TABLE II
Adult Income data comparison between Henelius et al and GFA consistency and accuracy scores for a C4.5 decision tree model. Feature predictability scores are correlation coefficient or Kappa statistic (for numerical or categorical features, respectively) when predicting that feature from the remaining features using two tree-based models.

For a real-world comparison to Henelius et al., we consider the Adult data set under a C4.5 decision tree model. The scores and rankings generated by Henelius et al. do not match those generated by GFA. Figure 5 shows that features `marital-status`, `fnlwgt`, `sex`, `education`, `education-num`, and `capital-gain` are outliers. In order to determine if this is due to the presence of proxy

[9] Available at: https://bitbucket.org/aheneliu/goldeneye/

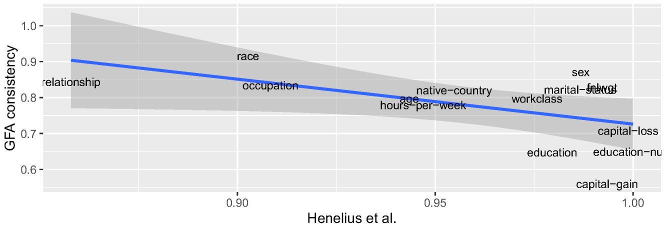

Fig. 5. Henelius et al. influence scores vs. GFA consistency scores shown with a linear regression and 95% confidence interval. The outlying features contain proxy information in the data set.

variables, or variables that are more complexly encoded in the remaining attributes by the decision tree, we then used two tree-based models to predict each feature from the remaining features (see Table II).[10] Models were built on the test set (using a $\frac{2}{3} : \frac{1}{3}$ training-test split) in order to replicate the data used for the audit. Reported predictability scores are the correlation coefficient for numerical features and the Kappa statistic for categorical features.

Looking at the resulting feature predictability scores, we see that `education` and `education-num` are both perfectly reconstructable from the remaining attributes. This is not surprising since `education-num` is a numerical representation of `education` and thus, a perfect proxy. The GFA consistency and accuracy measures both have `education` and `education-num` as tied for an importance score of 0.65, thus confirming that GFA handles indirect influence, while Henelius et al. has `education` with a score of 0.98 while `education-num` scores 0.92.

The features `marital-status` and `relationship` are both highly, but not exactly, predictable from the remaining attributes. This is likely because these are close, but not exact, proxies for each other. For example, the relationship status “Unmarried” may mean a marital status of “Divorced,” “Never-married,” or “Widowed.” The GFA scores for `marital-status` and `relationship` show they are of similar importance to the model, with consistency scores of 0.82 and 0.84 respectively (ranked 8.5 and 11.5 in importance), and accuracy scores of 0.77 and 0.81 (ranked 8.5 and 12). The Henelius et al. scores are less similar at 0.87 (ranked most important) and 0.99 (ranked 10.5). The GFA closely matching scores shows the procedure accounts for indirect influence of a feature, while Henelius et al. does not.

Recent work by Datta et al. [8] (discussed in more depth in Section II) present a similar approach, focusing on direct influence, that can additionally identify proxy variables. Identifying the influence of proxy variables using this method proceeds in two steps: first, the proxy variables are identified, then their individual direct influence in the ranked list is found. Considering the Adult data set, but under a different model, they find that `marital-status` and `relationship` are both proxies for `sex`. Their ranking finds that `marital-status` is ranked first for influence, `relationship` third, and `sex` ninth. Additionally, their influence score for `relationship` is under half of that for `marital-status`, and the influence score for `sex` is under half of that for `relationship`. Thus, similar to Henelius et al., the Datta et al. two step procedure does not account for the shared indirect influence of `marital-status` and `relationship` on the outcome.

E. Comparison to feature selection

Finally, we examine the relation between our auditing procedure and feature selection methods. While GFA is fundamentally different from feature selection sincea) features may not be removed from the model, and b) the model may not be retrained, due to their similar ranked feature results, we compare the resulting GFA ranking to a feature selection generated ranking. Feature selection was applied to the synthetic, Adult Income, and German Credit data sets. It was performed using a wrapper method (around both C4.5 decision trees and SVMs) and a greedy forward search through attribute subsets to generate a ranking.[11] For both the Adult Income and German Credit data sets feature selection created identical rankings for both decision trees and SVMs.

Spearman’s rank correlation is a nonparametric test of the relationship between two rankings, in this case the rank orderings generated by feature selection and by GFA. The synthetic data had a strong correlation between these rankings, with feature A ranked first by all methods, though given the small feature size this is not statistically significant. The Adult Income and German Credit data set ranking comparison correlations were both weak. This was not improved when using the consistency feature ranking instead, and in the case of the C4.5 decision tree was an order of magnitude weaker.

The weak correlation is not surprising since the two methods ask philosophically different questions. We are interested in the importance of a feature to a specific instance of a model, while feature selection considers importance with respect to an as-yet uninstantiated model. In addition, feature selection gives an indication of the direct influence of a feature, while GFA is focused on indirect influence.

VI. Discussion

Feature influence is a function of the interaction between model and data. A feature may be informative but not used by the classifier, or conversely might be a proxy and still useful. Thus, influence computation must exploit the interaction between model and data carefully. If the obscured and unobscured datasets are similar, then the classifier can’t have found useful signal and the classifier’s outputs won’t change much under our audit. If there *were* significant differences and the classifier used these differences in its model, then gradient

[10] Weka’s REPTree, J48, and M5P models were used for this analysis with the default model-building parameters. J48 was used to predict categorical features and M5P was used for numerical features. REPTree can handle both categorical and numerical features.

[11] Feature selection was implemented in Weka version 3.6.13 using WrapperSubsetEval and Greedy StepWise on J48 and SMO models. Default options were used, save for the generation of a complete ranking for all features.

feature auditing will show a change in the classifier behavior, as desired. Finally, if there were differences between attribute subgroups but those differences are irrelevant for the classifier, then gradient feature auditing will not show a large change in classifier behavior. This "sensitivity to irrelevance" is an important feature of a good auditing procedure.

It remains a challenge to effectively compare different approaches for auditing models since, as we have seen, different approaches can have points of agreement and disagreement. Our obscuring procedure prefers to use a computational metaphor – predictive power – rather than a statistical metaphor such as hypothesis testing, but it seems likely that there are ways to relate these notions. Doing so would provide a combined mathematical and computational framework for evaluating black-box models and might help unify the different existing approaches to performing audits.

VII. Related Work

In addition to early work by Breiman [4] and the recent works by Henelius et al. [13] and Datta et al. [8], a number of other works have looked at black-box auditing, primarily for direct influence [9], [27], [25], [24]. There are potential connections to privacy-preserving data mining [1]: however, the trust model is different: in privacy-preserving data mining, we do not trust the user of the results of classification with sensitive information from the input. In our setting, the "sensitive" information must be hidden from the classifier itself. Another related framework is that of *leakage* in data mining [15], which investigates whether the methodology of the mining process is allowing information to leak from test data into the model. One might imagine our obscuring process as a way to prevent this: however, it might impair the efficacy of the classifier.

Another related topic is *feature selection*, as we discussed in Section V-E. From a technical standpoint (see [7]), the *wrapper* approach to feature selection (in which features are evaluated based on the quality of the resulting prediction) is most related to our work. One such method is stepwise linear regression, in which features are removed from input for a generalized linear model based on their degree of influence on the model to make accurate predictions, measured using a correlation coefficient.

Model *interpretability* focuses on creating models that are sparse, meaningful, and intuitive and thus human-understandable by experts in the field the data is derived from [26]. The classic example of such models are decision trees [19], while recently supersparse linear integer models (SLIMs) have been developed as an alternative interpretable model for classification [26]. New work by Ribeiro et al. [21] trains a shadow interpretable model to match the results of a given classifier. For neural networks, various approaches based on visualizing the behavior of neurons and inputs have also been studied [28], [14], [16].

Finally, we note that a compelling application for black-box audits of indirect influence includes considerations of algorithmic fairness [22]. Understanding the influence of variables can help with such a determination.

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.

[2] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica*, May 23, 2016.

[3] N. Barakat and J. Diederich. Learning-based rule-extraction from support vector machines. In *Proc. of the 14th International Conference on Computer Theory and Applications*, 2004.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] C. Bucilua, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006.

[6] G. Casella and R. L. Berger. *Statistical Inference*. Cengage Learning, 2nd edition, 2001.

[7] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40:16–28, 2014.

[8] A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Proceedings of 37th IEEE Symposium on Security and Privacy*, 2016.

[9] W. Duivesteijn and J. Thaele. Understanding where your classifier does (not) work - the SCaPE model class for EMM. In *International Conference on Data Mining (ICDM)*, pages 809–814, 2014.

[10] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. *Proc. 21st ACM KDD*, pages 259–268, 2015.

[11] D. Freedman and P. Diaconis. On the histogram as a density estimator: L 2 theory. *Probability theory and related fields*, 57(4):453–476, 1981.

[12] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.

[13] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou. A peek into the black box: exploring classifiers by randomization. *Data Min Knowl Disc*, 28:1503–1529, 2014.

[14] M. Kabra, A. Robie, and K. Branson. Understanding classifier errors by examining influential neighbors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2015.

[15] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):15, 2012.

[16] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. In *Proc. ICML*, 2011.

[17] D. S. Massey and N. Denton. *American Apartheid: Segregation and the making of the underclass*. Harvard University Press, 1993.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[19] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[20] P. Raccuglia, K. C. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533:73–76, May 5, 2016.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. ACM KDD*, 2016.

[22] A. Romei and S. Ruggieri. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, pages 1–57, April 2013.

[23] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[24] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):1, 2008.

[25] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1, 2007.

[26] B. Ustun, S. Traca, and C. Rudin. Supersparse linear integer models for interpretable classification. Technical Report 1306.6677, arXiv, 2014.

[27] O. P. Zacarias and H. Bostrom. Comparing support vector regression and random forests for predicting malaria incidence in Mozambique. In *Advances in ICT for Emerging Regions (ICTer), 2013 International Conference on*, pages 217–221. IEEE, 2013.

[28] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Computer Vision — ECCV 2014*, pages 818–833. Springer, 2014.

[29] J. Zeng, B. Ustun, and C. Rudin. Interpretable classification models for recidivism prediction. Technical Report 1503.07810, arXiv, 2015.