2016 IEEE 2nd International Conference on Collaboration and Internet Computing

Detecting Discrimination in a Black-box Classifier

Yasmeen Alufaisan, Murat Kantarcioglu, Yan Zhou
The University of Texas at Dallas
Richardson, Texas USA
{yxa130630,muratk,yan.zhou2}@utdallas.edu

***Abstract*—Data mining techniques are playing an increasing role in making crucial decisions in our daily lives ranging from credit card approvals to employment decisions. Typically algorithms used to build decision models remain as a black-box to the end user. Therefore the process of decision making appears to be opaque. At the same time, increasing the transparency of a black-box decision making model allows us to discover hidden discrimination, and hold entities accountable. Although algorithmic transparency with respect to black box classifiers requires addressing many challenges, our main objective in this paper is to investigate whether a black-box classification model is biased against certain subgroups. Specifically, we study the indirect discrimination of hidden protected features. Protected features, such as race, gender, and religious beliefs, are those that are prohibited to be legally used for making decisions. Simply removing the protected features is not enough to eliminate discrimination because there could be strong correlations between protected features and non-protected features, such as race versus zip code. In this paper, we present two techniques to measure discrimination of a black-box model as a result of data bias or algorithmic weakness. Data bias is investigated further by introducing artificial bias to the dataset under consideration. Our experimental results demonstrate the effectiveness of our bias measures where bias comes from different sources.**

***Index Terms*—Data Mining, Discrimination, Machine Learning**

I. Introduction

Algorithmic transparency becomes an emerging issue as machine learning and data mining techniques are increasingly used in many real applications. More and more companies rely on machine learning based software products to make substantial decisions, for example insurance premiums and loan approval decisions, based on the input about a user. However, how the input data is manipulated and interpreted by the data mining algorithm is unclear. The entire process is running more or less in a black box where systematic bias and discrimination could have gone unnoticed because of its opaqueness. In this paper, we focus on detecting algorithmic discrimination caused by biases in data and weaknesses in the algorithms used to build the classifiers.

The discrimination problem in black box models could have different causes. First of all, the internal learning algorithm itself has its own weaknesses for some input data. For example, some algorithms may be biased towards linearly separable data where the separation is based on the features correlated with the protected features. Protected features are those that are prohibited to be used in decision making because of its tendency towards discrimination. Secondly, the data presented to the internal learning algorithm may induce bias by encoding strong correlations between protected features and the decision outcome. For example, some census data may show strong links between $(age, gender)$ and car accident rates. In addition, some non-discriminative (non-protected) features may be indirectly linked to the discriminative features. When they are presented to the internal learning algorithm for model building, the outcome may still be biased because of the inherent linkage to the discriminative features. Therefore, simply removing the protected features is not enough to eliminate discrimination. It is practically impossible for the end user to detect such systematic biases. Transparency allows us to discover hidden discrimination, and hold entities accountable.

Although there are some important work done related to algorithmic transparency (see Section II), existing research has the following limitations:

- The solutions may demand full access to the underlying classifier;
- The analysis may include protected features as part of the input data set;

To address the above mentioned limitations, in this paper, we present a bias detection model to 1.) detect potential correlation between a subset of features that are strongly correlated with the protected features, and 2.) measure discrimination as a result of algorithmic weakness or data set bias. We present a bias measure that clearly indicates whether a black-box classifier is biased against a group of samples. We explore the biases of a black-box classifier with a limited and small number of queries. We also demonstrate the pitfalls of making decisions using a biased classifier by injecting biases into the data. This involves manipulating a feature that is highly correlated with a protected feature that has a strong bias on the class value, or modifying data to inject artificial biases.

The rest of the paper is organized as follows. Section II presents the related work in the area of discriminative data mining. Section III formally defines the problem. Section IV presents our model. Section V presents experimental results on both artificial and real datasets. Section VI concludes our work and discusses future directions.

II. Related Work

The problem of discrimination has been studied in the data mining community before. The prior work in discriminative data mining has focused on two directions: discrimination discovery and discrimination prevention. Pedreschi et al. were the first to discuss discrimination discovery in data mining

978-1-5090-4607-2/16 $31.00 © 2016 IEEE
DOI 10.1109/CIC.2016.49

models [1, 2]. In their work, they proposed *elift* measure to discover discrimination in classification rules [1]. They consider direct discrimination due to the existence of protected features and indirect discrimination based only background knowledge only. Mancuhan and Clifton proposed discrimination discovery model that uses Bayesian network to estimate probability distribution of a class [3]. Their *Belift* measure includes the protected features when measuring discrimination and requires an initial Bayesian network structure with naïve assumption. They also proposed a discrimination prevention model that corrects the class labels for discriminated instances without using the protected features in the decision process. Luong et al. introduced k-NN approach for discrimination discovery and discrimination prevention [4]. Feldman et al. introduced the balanced error rate (BER) as a measure for discrimination [5]. They determine the existence of discrimination if the protected feature can be predicted using the non protected features (i.e the protected feature is leaked into the data). They measure predictability using the BER.

In [6], Mancuhan and Clifton proposed a discrimination prevention technique that corrects labels in a training set. Their work targets decision policies. A decision policy is a classification rule that makes decisions by considering all the features independently. Scoring function correction for naïve Bayes and Decision Tree classifiers were used as a technique to prevent discrimination in [7] and [8].

Sweeney investigated the issue of discrimination in online ad delivery [9]. The findings in [9] show that online ads delivery have a strong correlation with racially associated names. For an instance, ads with "arrest" in the ad text have higher percentage of appearing for black identifying first names compared to white identifying first names. Algorithmic transparency in black-box models received great attention recently. Datta et al. introduced Quantitative Input Influence (QII) that measures the influence of the input on a system's output [10]. They also use QII to present transparency reports that explains why a certain decision was made. A black-box auditing technique was developed by Adler et al. to rank the features used by a black-box based on their influence on the prediction outcome [11]. Both [10] and [11] assume access to the black-box input. Ribeiro et al. introduce LIME as a technique to identify an interpretable model that explains the predictions of any classifier [12].

We summarize our contributions and distinguish our work from the previous research as follows:

- In our work we focus on the case of black-box classifier where we have no access to the data used to build the model nor the statistical reasoning behind the model. Furthermore, the protected feature is not observed in any part of the discrimination discovery process. We show with empirical results that we can detect discrimination of a black-box classifier by probing the black-box classifier for class labels of a small number of instances.
- We propose a new measure for discrimination discovery named *Feature-based Targeted Sampling (FTS)*.
- We enhance the traditional *elift* measure to discover discrimination. We call our enhanced model ML*lift* due to the use of machine learning model for probability estimation. Unlike the work in [1] and [3], our ML*lift* measure is not limited to association rules and does not assume the knowledge of an initial Bayesian network.
- We develop a new technique to detect the correlation between a subset of non-protected features with hidden protected features.

III. Problem Definition

Consider the sets of features $A = \{a_1, a_2, ..., a_m\}$ and $B = \{b_1, b_2, ..., b_v\}$ where A contains the protected features and B contains the non-protected features. A protected feature in A can take multiple categorical values. We assume that one of these values represents minority subgroup instances $\mathcal{W}$ (e.g. senior citizens given "age" as the feature). Assume that B consists of two subsets: $B^c = \{b_1^c, b_2^c, ..., b_w^c\}$ and $B^{nc} = \{b_1^{nc}, b_2^{nc}, ..., b_u^{nc}\}$ where B^c has features that are highly correlated with A and B^{nc} has features that are not correlated with A. We assume that we know A and B but we have no knowledge whether a feature $b_j \in B^c$ or $b_j \in B^{nc}$. Each instance x_i has a class label $y \in \{1, -1\}$ where 1 represents the good class (e.g., credit approval) and -1 represents the bad class (e.g., credit rejection). A set of instances is represented as a matrix M where M_{ij} contains the i^{th} instance and the j^{th} feature.

We examine the bias of a black-box classifier $\mathcal{H}$ built on a *training dataset D that consists of features in B only; features in A are hidden and not used by $\mathcal{H}$*. With the black-box classifier we have neither the knowledge of the training data nor the statistical reasoning behind $\mathcal{H}$'s decision making. We only have the knowledge about the input space (the set of features used to learn $\mathcal{H}$). We use $\mathcal{H}$ as an oracle to query the labels of a limited number of data instances.

We assume access to data that is disjoint from the black-box input D but follows the same distribution as D. We will refer to this data as the test data T. We further divide T into two sets: queried test data Q and non-queried test data R. The queried test data Q receives labels for its instances by querying the black-box classifier, therefore the accuracy of the labels depends on the accuracy of the black-box. We limit the number of queries for labels from the black-box classifier to control the cost and avoid detection by the black-box classifier. Sending unlimited queries to a black-box classifier is usually cost-prohibitive. Take as an example a black-box loan-approval classifier. Real people with information for identity verification (e.g. social security numbers) would be required as input to the loan-approval classifier; however, collecting person-specific information in many cases is expensive. As a result, unlimited probing is impractical. In addition, unlimited probing may also raise security concerns.

IV. Discrimination Detection Model

In this section, we describe our discrimination detection technique that can be used to unveil biases in a black-box

model. The technique consists of two steps. The first step is *correlation discovery* which identifies the features in B^c that are highly correlated with the minority subgroup $\mathcal{W}$. The second step is *discrimination discovery* where we present several models to measure the bias/discrimination in a black-box classifier.

A. Correlation Discovery

In the correlation discovery model we aim to find the set of features B^c that are highly correlated with a minority subgroup $\mathcal{W}$. If our protected feature A is age, we would set $\mathcal{W}$ to be the minority subgroup of senior citizens. A special character that we also seek in B^c is a relationship with discriminated instances. The discriminated instances are the ones that receive different classification on the grounds of their protected features. Therefore, our correlation discovery goal is to search for a partition of the instances and a subset of the features where the projection of these dimensions is closer to each other than any other members in a different dimension.

The correlation discovery model will take the test data T as its input since it is the only data we have access to. The existence of the protected feature A in T does not contradicts our main claim that the protected feature is not observed by the black-box model. That is because we only assume we have knowledge of the protected feature in the unlabeled test data T which can have access to the protected feature using background knowledge. For example, we can link the non-protected feature zip code to the protected feature race using an external data such as the census data [13].

We start the correlation discovery process by dividing our test data matrix T into two partitions: the instances will be placed in one partition and the set of features in the other partition. Since we aim to find the set of features correlated with a subgroup $\mathcal{W}$ that has a discriminative value, we convert the feature space into a binary input space. For each categorical feature we create k binary features where k is the number of all possible values the feature can take. With continuous features, we convert each feature value to "one" if it is above the mean or to "zero" otherwise. To find the minimum set of features that is correlated with the set of instances in $\mathcal{W}$, we use spectral bi-clustering model [14] as follows:

- Create the diagonal matrices D_1 and D_2 where $D_1(i,i) = \sum_j T_{i,j}$ and $D_2(j,j) = \sum_i T_{i,j}$.
- Form the matrix $T_n = D_1^{-1/2} T D_2^{-1/2}$.
- Compute the singular vectors U and V of T_n using Singular Values Decomposition (SVD) [15].
- From the matrix $Z = \begin{bmatrix} D_1^{-1/2}U \\ D_2^{-1/2}V \end{bmatrix}$
- Run k-means clustering algorithm on Z to obtain the multi-partitioning.

Finding correlation using spectral bi-clustering model has an advantage over the statistical methods (e.g independence tests) for discrimination detection for two main reasons. First, in order to use the statistical methods to identify the correlation between A and B^c, we need to have know the exact features in B^c. Since our initial goal is to identify the subset of features in B^c, we need to compute the correlation between every possible subset of the features in B and A with the statistical methods which is computational infeasible. The second and most important reason is that bi-clustering has the ability to uncover similar patterns of behaviors between a subset of rows across a subset of columns [14]. This serves our purpose of finding discriminated instances that inter-correlate with features in A and B^c.

B. Discrimination Discovery

We now introduce our discrimination measures. We define two different measures: *Machine Learning extended lift* (ML*lift*) and *Feature-based Targeted Sampling* (FTS). Using both measures we can uncover a black-box classifier bias caused by either algorithmic weakness or data-oriented bias. Because any measure can be biased by itself, using two different measures can increase the validity of our conclusion about whether a black-box classifier is discriminative or fair.

1) MLlift: Machine Learning extended lift (ML*lift*) is a measure for discrimination that can use any machine learning model such as Naïve Bayes and Logistic Regression to estimate probabilities. The idea behind ML*lift* comes from the known discrimination measure *elift* introduced by Pedreschi et al. in [1]. Given itemsets A and B and a class C, *elift* measures discrimination in a classification rule as follows:

$$\frac{conf(A,B \rightarrow C)}{conf(B \rightarrow C)}$$

where A is a potentially discriminative itemset and B is a potentially non-discriminative itemset. *Elift*'s goal is to measure the increase in confidence for a class value C when adding an itemset A. One disadvantage of *elift* is its limited application to classification rules where the itemsets A and B are assumed to be independent of each other.

Our ML*lift* is an extension to *elift*. We can apply ML*lift* to measure discrimination in any instance with any number of features. Additionally, when measuring discrimination with ML*lift* we do not take into account the protected features in A. We consider A to be hidden and we measure the extended influence of A through its correlated features in B^c. ML*lift* for an instance x_i is defined as follows:

$$\text{ML}lift(x_i) = \frac{P(C|b^c_{i1}, b^c_{i2}, ..b^c_{iw}, b^{nc}_{i1}, b^{nc}_{i2}, ..., b^{nc}_{iu})}{P(C|b^{nc}_{i1}, b^{nc}_{i2}, ..., b^{nc}_{iu})} \quad (1)$$

where b^c_{ij}, and b^{nc}_{ij} represent features $b^c_j \in B^c$ and $b^{nc}_j \in B^{nc}$ for the instance x_i. An instance x_i is considered discriminated against if ML$lift(x_i) > \epsilon$.

Since our goal is to measure discrimination in a minority set of instances $\mathcal{W}$, we only consider the increased confidence of x_i being placed in the bad class, that is $C = -1$, given that x_i is in $\mathcal{W}$. Accordingly, ML*lift* of a black-box model is

computed as the percentage of discriminated instances in the minority subgroup:

$$\text{ML}\textit{lift}(\mathcal{W}) = \frac{\sum_{i=1}^{n} I(x_i)}{|\mathcal{W}|}, \text{where:}$$
$$I(x_i) = \begin{cases} 1, & \text{if ML}\textit{lift}(x_i) > \epsilon \text{ and } x_i \in \mathcal{W} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

We can estimate ML*lift* probabilities using any machine learning model that outputs class membership probabilities. However, since each ML model brings its own bias and variance to the final results, we will compare the results of ML*lift* using different ML models to determine the best model to use. We will consider Logistic Regression (LR*lift*), Random Forest (RF*lift*), and SVM with rbf kernel (SVM*lift*). We also consider a model that combines the probabilities of LR*lift*, RF*lift*, and SVM*lift* which we will refer to as CB*lift*. We define the combined lift as follows:

$$\text{CB}\textit{lift}=\frac{P_{\text{RF}}(C|B^c \cup B^{nc}) + P_{\text{SVM}}(C|B^c \cup B^{nc}) + P_{\text{LR}}(C|B^c \cup B^{nc})}{P_{\text{RF}}(C|B^{nc}) + P_{\text{SVM}}(C|B^{nc}) + P_{\text{LR}}(C|B^{nc})}$$

To avoid zero probability in computing ratios, we add δ-smoothing to all ML*lift* computation denominator and nominator where δ=0.01.

ML*lift* uses the queried test data Q, where Q labels are obtained from the black-box classifier, to estimate the probabilities and then measure the discrimination for the instances in the remaining test data R.

2) Feature-based Targeted Sampling (FTS): In this section we describe our Feature-based Targeted Sampling (FTS) to measure discrimination in a black-box model. FTS construct artificial data by sampling features' values from two groups. The choice of which group to sample from depends on whether a feature is correlated or not correlated with the protected feature. The first group includes instances with good class $C = 1$, we refer to this group as $\mathcal{G}$. The second group contains instances in the protected subgroup $\mathcal{W}$. Formally, a sample $x_s = \{x_{s1}, ..., x_{sm}\}$ is constructed as:

$$x_{sj} = \begin{cases} x_{ij}, x_i \in \mathcal{W}, & \text{if } \mathcal{X}_j \text{ in } B^c \\ x_{ij}, x_i \in \mathcal{G}, & \text{if } \mathcal{X}_j \text{ in } B^{nc} \end{cases} \quad (3)$$

where $\mathcal{X}_j$ is the j-th feature and x_i is an instance chosen at random. With FTS, discrimination in the minority subgroup is measured by defining a Random Forrest model (RF) that takes as an input the set of instances in a sample $\mathcal{S}$ created as in Eq. (3) as follows:

$$score_{mnr} = \frac{\sum f(x_s)}{|\mathcal{S}|}$$

where $x_s \in \mathcal{S}$, and $f(x_s) = \begin{cases} 1, & \text{if } RF(x_s) = -1 \\ 0, & \text{otherwise} \end{cases}$

Since most of the features are in B^{nc} and only a small set of correlated features is in B^c, our sample should have a small number of instances being classified as bad. If we observe a high percentage of the sample being classified as bad, we can conclude that the black-box model contains high bias against a minority subgroup $\mathcal{W}$ since sampling the features that are in B^c cause an instance to be classified as bad even though the other features are sampled from $\mathcal{G}$. To quantify the results of FTS, we will compute $score_{mjr}$ which is the result of FTS when sampling B^c features from the majority group (e.g male in the gender protected feature). We can then quantify FTS discrimination by computing: $score_{diff}$=$score_{mnr} - score_{mjr}$

FTS uses the queried test data Q for sampling because it is the only data available with labels.

V. Experiments

We demonstrate the effectiveness of the bias/discrimination measures introduced in the paper on both artificial and real datasets. We split the datasets into two partitions. One partition contains the training data D and the other partition contains the test data T. The test data is further divided into k folds to produce the queried test data Q and non-queried test data R. We let the size of Q equal to $\frac{k-1}{k}$ and the size of R equal to $\frac{1}{k}$. To allow each instance in the test data to be a part of each of the two sets: Q, and R, we run k-fold cross validation on Q and R. The final results are averaged over the results of all the folds. For the baseline results we set $k = 2$. This experimental setting is essential for the following reasons:

- Our goal is to simulate real life settings where the black-box model has a partition of the data for training purposes and we can only have access to a small set of samples with labels obtained from the black-box model. We can also have access to a set of unlabeled samples.
- Q size is very important because we use Q to 1.) estimate the probabilities in ML*lift*, and 2.) sample features' values in FTS. Therefore, we vary the size of Q and choose the best size that is large enough to provide good results but not not too large to avoid detection by the black-box when querying many labels.

We assume that there is only one protected feature and this protected feature is *known but hidden; it is not used by the black-box classifier*. The only usage of the protected feature is in the correlation discovery between the protected feature A and its highly correlated features B^c.

During the correlation discovery process, an important decision to make is determining the right number of clusters k. Our discrimination discovery model has high dependency on the strength of correlation between the protected feature A and its highly correlated features B^c. For this reason we vary k from 2 to max-k and pick k that produces correlated features B_k^c with high quality. For each k, we measure correlation quality by building a classification algorithm $f : B_k^c \rightarrow A$. We select k that yields the highest accuracy. We exclude any k that produces B_k^c if B_k^c contains more than 50% of the total features in B. We use Random Forest as our classification model f and set max-k to 20 because with all the datasets we reach consistent clusters around k=20.

As discussed in Section IV-B, to compute ML*lift* we need to choose the best ML model for probability estimation. When comparing the results of RF*lift*, SVM*lift*, LR*lift*, and CB*lift* we find that all these models give very close results to each other. We then partition the data into two sets and compute ML*lift*

using all the discusses models. SVM*lift* has the most consistent results among the sets and therefore we will be using it in all the experiments as our default ML*lift*.

For all the datasets, we measure the discrimination of different black-box models. These models are: Random Forest (RF), SVM with RBF kernel (SVM-rbf), SVM with linear kernel (SVM-L), naïve Bayes (NB), Logistic Regression (LR), and Decision Tree (DT). We use the black-box classifiers to assign labels to the queried test data Q. After that, Q is used to estimate probabilities for ML*lift*. The discrimination is measured using the unlabeled data R. We set the threshold for ML*lift* to $\epsilon = 1$ due to U.S legislation on equal treatments with respect to the protected features [16]. An instance with ML*lift*> 1 is considered discriminated by the black-box classifier. For FTS, we sample the features' values from Q since it is the only labeled data we have access to.

Due to the lack of ground truth about discrimination in the datasets, we need to have a baseline to validate the results of our measures. We need to be able to answer the following question: when can we consider a dataset or an algorithm biased against a subgroup? For this reason, we compare our results to the 80% rule recommended by the U.S Equal Employment Opportunity Commission (EEOC) to measure disparate impact [17]. In U.S law, disparate impact is the indirect or intentional discrimination for different groups based on their protected features. The disparate impact can be formally define as follows:

$$\text{DI:}\frac{P(C=1|\text{minority})}{P(C=1|\text{majority})} \leq 0.8$$

where C=1 represents the good class.
When reporting discrimination results, we will be validating our measures using the 80% rule. Detecting DI less than 0.8 indicates the existing of bias in the black-box model or in the data if no black-box is used. A decrease in DI indicates an increase in the discrimination.

All the experiments are implemented using MATLAB Statistics and Machine Learning Toolbox.

A. Experiments on Artificial Datasets

We generate artificial data with a multivariate Gaussian distribution. Our artificial data consists of m continuous features. Each feature $\mathcal{X}_i$ has a weight w_i. The class value y is a linear combination of the features' weights. For an instance x_i the class value y_i assigned using the following rule:

$$y_i = \begin{cases} -1, & \text{if } \sum_{j=1}^{m} w_{ij}(\mathcal{X}_{ij}) >= 0.5 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

where w_j is the weight of the feature $\mathcal{X}_j$ and $\sum_{j=1}^{m} w_j = 1$. In the following experiments we set $m = 7$ and the number of instances $n = 1000$. We randomly split the data into two equal partitions to obtain the training\test data.

Let $\mathcal{X}_1$ be our protected feature. The protected subgroup in this case will be the instances with $\mathcal{X}_1 > mean(\mathcal{X}_1)$ due to their high probability of being in the bad class given our class assignment in Eq. (4). We assign high correlation between the features $\mathcal{X}_1$ and $\mathcal{X}_2$ to measure the effect of having a high correlation with $\mathcal{X}_1$ when $\mathcal{X}_1$ is hidden.

During the correlation discovery phase, the bi-clustering model was able to catch the correlation between $\mathcal{X}_1$ and $\mathcal{X}_2$ by placing them in the same cluster while all the other features were allocated to different clusters. This suggests that we have an effective model for correlation discovery. The accuracy of $f : \mathcal{X}_2 \rightarrow \mathcal{X}_1$ is 0.998 which is expected since we assign high correlation between $\mathcal{X}_1$ and $\mathcal{X}_2$.

To investigate the influence of the hidden feature $\mathcal{X}_1$, we manipulate its correlated feature $\mathcal{X}_2$ and observe its effect on the outcome of decision making. We manipulate $\mathcal{X}_2$ as follows:

$$x_{i2} \rightarrow x_{i2} \pm x_\delta \text{ where } x_\delta \in [a, b] \text{ and}$$
$$\begin{cases} a = \min\{\mathcal{X}_2\} \text{ and } b = \mu\{\mathcal{X}_2\}, & \text{if } x_{i2} > \mu\{\mathcal{X}_2\} \\ a = \mu\{\mathcal{X}_2\}, \text{and } b = \max\{\mathcal{X}_2\} & \text{otherwise} \end{cases}$$

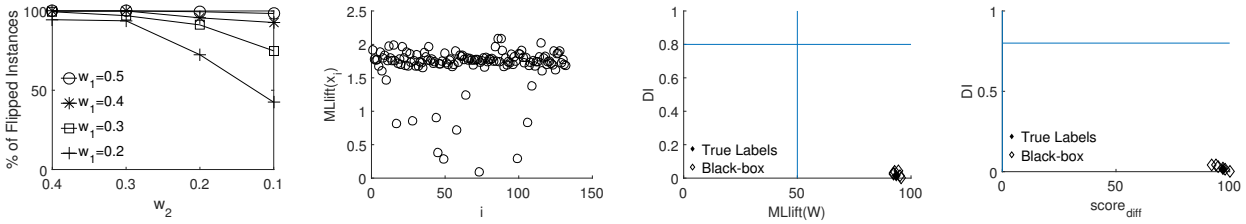
where $\mu\{\mathcal{X}_2\}$ is the mean value for the feature $\mathcal{X}_2$.

When performing this manipulation, we calculate how many instances change their prediction from bad class to good class when varying the weights of $\mathcal{X}_1$ and its correlated feature $\mathcal{X}_2$ using Random Forest model. Figure 1(a) presents the results when w_1=0.5, 0.4, 0.3, and 0.2 and w_2= 0.4, 0.3, 0.2 and 0.1. As the weights of $\mathcal{X}_1$ and $\mathcal{X}_2$ increase, the percentage of flipped instances increases. When $w_1 = 0.2$ and $w_2 = 0.1$, the percentage of flipped instances reaches 42%. In this case each of the non-correlated features has a weight of $(1 - (w_1 + w_2)/(n - 2) = 0.14$. Even when $\mathcal{X}_2$ has a weight value of 0.1 that is less than the non-correlated features weights, manipulating $\mathcal{X}_2$ causes many instances to flip their class due to the influence of $\mathcal{X}_1$ on $\mathcal{X}_2$. This proves that removing the protected features alone is not enough to protect from prohibited discrimination against a protected group when features correlated with a protected feature are present in the dataset.

Discrimination Discovery Results For all the following experiments, the class label y is obtained with $w_1 = 0.5, w_2 = 0.4$ and $\sum w_j = 0.1$ for $j > 2$. We assign high weights for $\mathcal{X}_1$ and $\mathcal{X}_2$ to evaluate our discrimination models when the class is highly biased toward the protected hidden feature and its correlated features.

Figure 1(b) shows the real ML*lift*(x_i) (i.e. when no black-box is used) for each instance x_i in the minority subgroup in the test data R. A significant number of instances is grouped between the values 1.5 to 2 which means that they are about twice as likely to be given a bad class when considering the correlated feature $\mathcal{X}_2$. This data revealed 93% discrimination with ML*lift*$(\mathcal{W})$ which is expected since we intentionally generated biased data. ML*lift*(x_i) represents the discrimination at the instance level, as defined in Eq. (1), where any instance with ML*lift*$(x_i) > 1$ is a considered discriminated instance. On the other hand, ML*lift*$(\mathcal{W})$ measures the discrimination for all the instances in the minority subgroup $\mathcal{W}$; it is the percentage of discrimination against $\mathcal{W}$ as defined in Eq. (2).

When introducing black-box models, the accuracy represented by the true positive rate (TPR) for all the black-box models reaches 96% and above. Given their high accuracy,



(a) Data Manipulation B^c (b) ML*lift*(x_i)-True Labels (c) ML*Lift*($\mathcal{W}$) (d) FTS

Fig. 1. Artificial Data

we would expect the discrimination measures to be close to the discrimination when measured with the true labels with no black-box used.

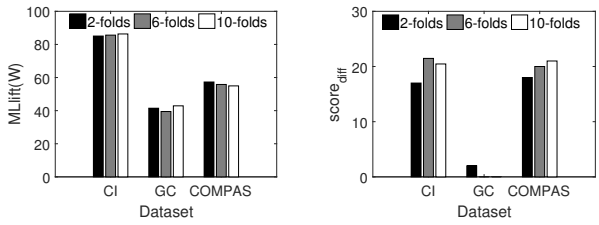
In Figures 1(c) and 1(d) we present the discrimination results where the filled marker represent the true labels and each non-filled marker represent a different black-box classifier. As can be seen from Figure 1(c), all the black-box models have ML*lift*($\mathcal{W}$) very close to the true labels where the minimum ML*lift*($\mathcal{W}$) is 92% and the maximum is 96%. The y-axis represents the disparate impact measured by the 80% rule. Similar to ML*lift*($\mathcal{W}$) all the black-box models have close DI where the maximum difference between them is 0.042. Figure 1(d) shows the relationship between FTS $score_{diff}$ and DI. FTS $score_{diff}$ represents the increase in the number of instances in the sample that are given bad class when sampling B^c values from the minority subgroup compared to the majority. We notice that as the bias measured by FTS increases, the bias measured by DI increases as well.

B. Experiments on Real Datasets

In this section, we demonstrate the success of our discrimination detection model using three real datasets: Census Income and German Credit taken from UCI data repository [18] and COMPAS data collected by ProPublica [19].

- Census income (CI) has 48,842 instances and 14 features. These features are: age, work class, final weight, education, number of years in education, martial status, occupation, relationship, race, sex, capital-gain, capital-loss, hours per week, and native country. Similar to [3], the final weight feature is ignored because it should be part of the learning process. The class value is low income (less or equal to 50K) or high income (greater than 50K). We will refer to low income as the bad class and the high income as the good class. The class distribution is 24.78% for the good class and 75.22% for the bad class. We consider "female" as our protected subgroup. We use the original data split for training\test in our experiments.
- German credit (GC) represents good\bad credit risks for 1000 instances. The dataset contains 20 features. These features are: status of existing checking account, duration in month, credit history, purpose, credit amount, savings account/bonds, present employment since, installment rate in percentage of disposable income, personal status and sex, other debtors / guarantors, present residence since, property, age, other installment plans, housing, number of existing credits at this bank, number of people being liable to provide maintenance for, telephone, and foreign worker. The data has 70% instances with good class and 30% instances with bad class. We set the protected subgroup value for this data to "female and not single". We randomly split the data into two equal partitions to obtain the training\test data.
- COMPAS stands for Correctional Offender Management Profiling for Alternative Sanctions. It is a scoring system used to assign risk scores to criminal defendants to determine their likelihood of becoming a recidivist. ProPublica collected records for more than 10,000 criminal defendants in Broward County, Florid. COMPAS data contains the prediction of two scores: *risk of recidivism* and *risk of violent recidivism*. We only consider the risk of recidivism scores because risk of violent recidivism has only 20% accuracy when compared to the actual recidivism [19]. ProPublica reported that the risk of recidivism has an accuracy of 61%. We use the same features selected by ProPublica and these features are: sex, age, race, priors count, charge degree, and risk of recidivism score. Scores can be either high, medium, or low. Similar to ProPublica, we combine medium and high scores to constitute a high score. Therefore our class values are: high and low scores. We selected instances with race values equal to black and white only. The selected sample has 6,150 instances. We consider race as our protected feature with "black" as the protected subgroup value. We randomly split the data into two equal partitions to obtain the training\test data.

An important factor that we should take into account is that census income has an inherit discrimination due to the fact that females on average receive lower income than males in the United States. For an instance, the female-to-male earnings ratio was 0.79 in the year of 2014 [20]. COMPAS data is also shown to contain discrimination against black race [19]. ProPublica demonstrated in their work that black defendants are twice as likely to receive high scores compared to white defendants. With German credit data, the discrimination should be lower since it has data for people who are already approved for credit. Therefore we would expect a good discrimination measure to show higher discrimination for census income and COMPAS data compared to German credit. We will use



(a) ML*lift*($\mathcal{W}$) (b) FTS

Fig. 2. The results of varying the number of folds between Q and R test data

TABLE 1
CORRELATION DISCOVERY RESULTS

Data set	Correlated Features	TPR
Census Income	marital status, occupation, relationship, race	83%
German Credit	status, history, purpose, savings, employee since, housing	68%
COMPAS	age, priors count	65%

this background knowledge for additional evaluating of the different discrimination measures.

Before considering the black-box models, we conduct some tests on the original data to justify choices made in regards for some parameters. For both ML*lift*($\mathcal{W}$) and FTS discrimination measures, we test the effect of varying the queried test data (Q) size on the results of each measure. The size of Q is especially important to ML*lift* measure that depends on probability estimation. We need enough samples in Q to have good probability estimation but we would also like to keep Q size as small as possible due to the restriction in querying the black-box model. We vary Q size by giving it $\frac{1}{2}$ (2-folds), $\frac{5}{6}$ (6-folds), and $\frac{9}{10}$ (10-folds) of the test data. Figure 2(a) presents the effect of varying Q size on the results of ML*lift*($\mathcal{W}$). We notice that changing Q size has a very small influence in all the datasets. The maximum difference is 1%, 3%, and 2% in census income, German credit, and COMPAS respectively. Figure 2(b) shows the results of FTS when using different sizes of Q. Similar to ML*lift*($\mathcal{W}$), there is no significant change in FTS $score_{diff}$ when varying the number of folds. The highest difference among all the datasets is 4% with census income data. Since our goal is to query the black-box classifiers for labels of a small number of samples and in all cases the size did not have a great impact on the results, we use only half of the test data to query the black-box for Q labels in all further experiments. Using 2-folds does not result in querying the black-box classifier for many labels and at the same time gives us realistic probability estimations compared to the results of 10-folds.

In Table 1 we present the correlated features B^c identified during the correlation discovery phase along with the accuracy measured by the true positive rate (TPR) for $f : B^c \rightarrow A$. For all the datasets, the chosen correlated features are the ones generated from a cluster that results in the highest TPR with f. Census income data has the highest TPR for

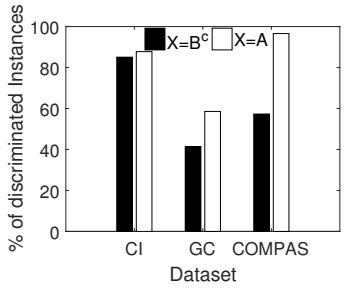


Fig. 3. MLlift: The percentage of discriminated instances discovered with $\frac{P(-1|X \cup B^{nc})}{P(-1|B^{nc})}$

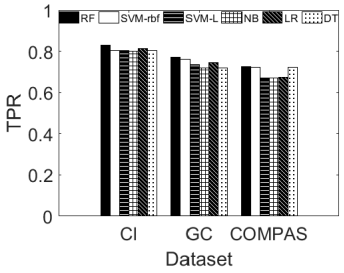
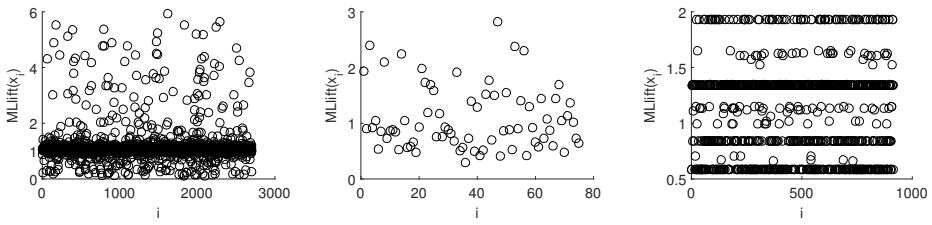


Fig. 4. Black-box models accuracy

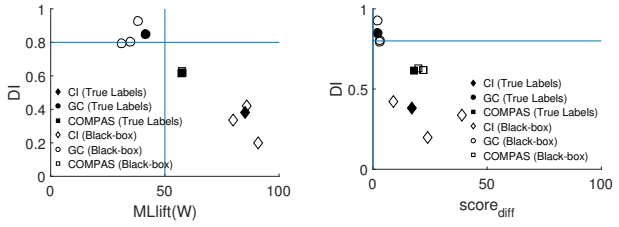
f which indicates strong correlation between A and B^c. German credit and COMPAS revealed lower TPR with f so we would expect weaker correlation in these two datasets. To further validate the correlation strength between A and B^c, in Figure 3 we compare the percentage of discrimination in the minority subset as a result of $\frac{P(-1|B^c \cup B^{nc})}{P(-1|B^{nc})}$ (i.e. ML*lift*) and $\frac{P(-1|A \cup B^{nc})}{P(-1|B^{nc})}$. Census income has less than 3% difference when replacing B^c with A but the other datasets show greater difference. These results confirms the fact that with strong correlation between A and B^c, removing A from the data is not enough to remove potential discrimination when A is leaked to the data through B^c. When using the black-box models, we can see that each model has different accuracy for each dataset in Figure 4. Census income has better accuracy than German credit and COMMPAS. Census Income has 80% accuracy on average among all the black-box modes. German credit accuracy ranges from 71% to 77%. COMPAS has an accuracy of 72% with RF, SVM-rbf, and DT. With weaker classifiers such as SVM-L, NB, and LR, COMPAS accuracy becomes lower than 70%. We will exclude SVM-L, NB, and LR from any further experiments due to their low accuracy with COMPAS data.

Discrimination Discovery Results We start the discrimination discovery results by showing discrimination at the instance level with ML*lift*(x_i). In Figure 5 we show ML*lift*(x_i) for the instances in each dataset using the true labels. Figure 5(a) shows the results for ML*lift*(x_i) with census income. A significant number instances have their ML*lift*(x_i) value larger than 1 which indicates the existence of high discrimina-



(a) Census income (b) German credit (c) COMPAS

Fig. 5. MLlift values for instances with their true labels



(a) ML*lift*($\mathcal{W}$) (b) FTS

Fig. 6. Discrimination in the black-box models

tion in the data. Census income data has 85% discrimination with ML*lift*($\mathcal{W}$). German credit has about 40% discrimination among the minority subgroup when measured with ML*lift*($\mathcal{W}$) as shown in Figure 5(b). COMPAS data has only five features including the protected feature race that we consider hidden. Without considering race, our set of features is: sex, age, prior count, and charge degree. Due to the small number of features, we expect many instances to have non unique records. This explains the patterns in ML*lift*(x_i) results shown in Figure 5(c). ML*lift*($\mathcal{W}$) for COMPAS data is 57%.

The black-box classifiers discrimination results and their relationship with the disparate impact DI are shown in Figure 6. Each maker type represents the results of a different dataset. The diamonds, circles, and squares represents census income, German credit, and COMPAS results respectively. A filled marker shows the results for the true labels where no black-box classifier is used. For example, a filled circle represent German credit results with the true labels. Each non-filled marker is the result of different black-box model. The DI baseline is 0.8 where any value for DI less than 0.8 indicates discrimination.

The results of ML*lift*($\mathcal{W}$) with the different black-box models and their relationship to the disparate impact (DI) is shown in Figure 6(a). We set the baseline for ML*lift*($\mathcal{W}$) to 50%. In other words, any data with more than 50% of its minority instances with ML*lift*>1 will be considered to have high discrimination against minorities. With all the datasets we notice a consistency between the relationship of ML*lift*($\mathcal{W}$) and DI where all the black-box models agree on discrimination using both measures. With census income, the highest discrimination is with Random Forest as the black-box with both ML*lift*($\mathcal{W}$) and DI. With German credit there are two cases of false positives. However, these two cases are on the borderline of being in the discrimination zone using DI but indicated as fair with ML*lift*($\mathcal{W}$). COMPAS data have identical results for both measures with all the black-box models.

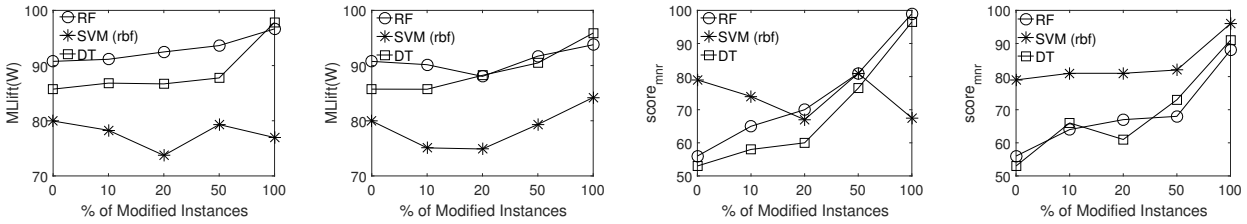
FTS relationship with disparate impact DI is shown in Figure 6(b). Having FTS $score_{diff}$ value equal to zero means that we have equal FTS $score_{mnr}$ and $score_{maj}$ values. German credit is the only data that has false positives. However, these false positives have FTS $score_{diff}$ equal to 3% at most. Census income results are consistent with DI. Using both measures, the results of true labels and black-box models indicated discrimination. COMPAS yields similar results for the black-box models. These results are also close to the values of true labels where no black-box is used.

Artificial Discrimination Injection

In this section we analyze what happens to the data when we inject artificial discrimination to the minority subgroup. Artificial discrimination is added to the data by altering the class labels as follows: $(x_i, y_i) \rightarrow (x_i, y_i')$ where x_i is in the minority subgroup $\mathcal{W}$, $y_i = 1$ and $y_i' = -1$.

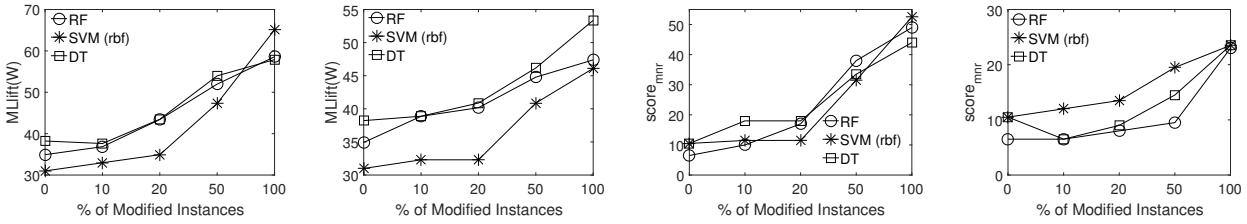
By flipping the labels for users in the minority subgroup $\mathcal{W}$ from good class to bad class we increase the percentage of users that are discriminated against. This artificial discrimination injection would allow us to further investigate the influence of the protected hidden feature on the class value. For example, consider the features "college student" and "age". Being an undergraduate college student is highly correlated with the age 18 to 24. Therefore, we can simulate changing the labels for instances with age 18 to 24 by changing the labels for instances that are undergraduate college students when the age is hidden.

The artificial discrimination is added to the black-box input D in one experiment and to the black-box output in separate experiment Q. We examine the effect of such injection on ML*lift*($\mathcal{W}$) and FTS. Figures 7, 8, and 9 show the results of injecting artificial discrimination to census income, German credit, and COMPAS data accordingly. The x-axis represents the percentage of modified instances in the minority subgroup with good class. We will refer to these instances as the target group. We notice in all the results that injecting discrimination to the training data has more effect on the discrimination measures compared to the same injection to the test data. That is because training data size is larger than the test data size Q. It is expected that manipulating larger size of instances would have larger impact.



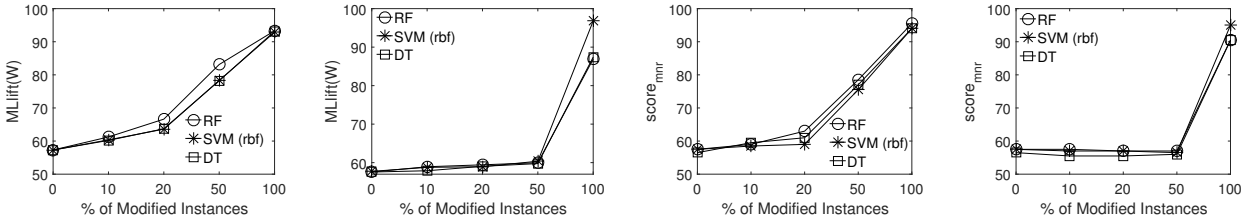
(a) MLlift-flipping black-box input (b) MLlift-flipping black-box output (c) FTS-flipping black-box input (d) FTS-flipping black-box output

Fig. 7. Census income-Artificial Discrimination



(a) MLlift-flipping black-box input (b) -MLlift-flipping black-box output (c) FTS-flipping black-box input (d) FTS-flipping black-box output

Fig. 8. German credit-Artificial Discrimination



(a) MLlift flipping black-box input (b) MLlift flipping black-box output (c) FTS flipping black-box input (d) FTS flipping black-box output

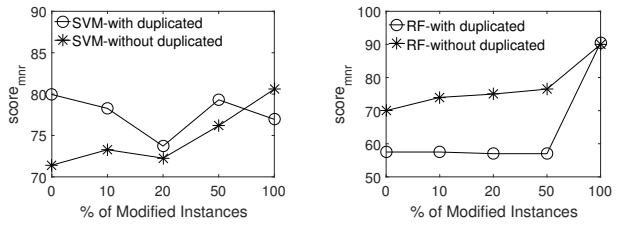
Fig. 9. COMPAS-Artificial Discrimination

Figure 7 shows the results of injecting bias into census income data. Census income has a small percentage of minority instances where they represent about 33% of the total population. Furthermore, the percentage of instances in the target group is very low. These instances represents less than 4% in the training data and less than 1% in the test data Q when using SVM-rbf as the black-box. Given these information and the initially high discrimination, the mild increase in ML*lift*($\mathcal{W}$) and FTS $score_{mnr}$ is acceptable.

Figure 8 presents the results of injecting artificial discrimination to German credit data. When manipulating 50% of the target group in the training data, that is equal to 10% of the training data, ML*lift*($\mathcal{W}$) has a minimum increase of 16% using all the black-box models and FTS $score_{mnr}$ has a minimum increase of 21%.

Artificial discrimination injection results with COMPAS data are shown in Figure 9. Both ML*lift*($\mathcal{W}$) and FTS $score_{mnr}$ have similar results. When injecting bias by flipping the labels of all the target group in training data D, which represent 24% of D, with both measures we get a minimum increase of 35% in discrimination. When injecting discrimination to all the target group instances in the black-box output, the minimum increase of discrimination is 29% with both measures and among all the black-box models.

There are some unexpected behaviors in the artificial discrimination injection results. An example of such unexpected behavior is with census income data and SVM as the black-box when flipping 100% of the target group in the training data. Both ML*lift* and FTS showed a decrease in the discrimination instead of the expected increase. One reason we examined is the existence of duplicate records that could confuse the decision making model when flipping class values. The classifier may find identical instances with contradicting labels. In Figure 10(a) we present the difference between SVM reaction to the artificial discrimination injection with and without the existence of duplicate records or instances with FTS. As we can see, the increase in the discrimination becomes consistent when removing duplicates. Another case of unexpected behavior is with flipping labels in test data Q with COMPAS data. COMPAS data has few features and therefore many duplicates. For this reason, when injecting bias to less than 100% of the target group in Q we only create



(a) Census income-FTS flipping black-box input

(b) COMPAS-FTS flipping black-box output

Fig. 10. Comparing artificial discrimination results where there is duplicate records and without duplicates

inconsistency where duplicates have conflicting labels. In this case we notice no increase in discrimination. In Figure 10(b) we can see greater increase in discrimination when removing duplicates compared to the original state with duplicates with RF as the black-box and FTS measure.

VI. Conclusion and Future Work

In this paper, we investigate the emerging problem of detecting discrimination in black-box classifiers. We created a correlation discovery model that can identify the most correlated features with a protected subgroup. We further proposed two different measures to detect indirect discrimination: ML*lift* and FTS. We demonstrated in our experimental results the success of our measures. We validated our measures by comparing their results to the disparate impact measured by the 80% rule recommended by EEOC. When we had a fair data, all the black-box models showed fair results as well. We also had discriminative datasets and we showed how all the black-box models inherited the discrimination using both ML*lift* and FTS. ML*lift* uses machine learning models to estimate class probabilities. The choice of which model to use is challenging because each model has its own bias to add to the final discrimination results. With the empirical evaluation we showed that the final discrimination results is mostly due to the bias created by the black-box classifier. That is because the other measures used, FTS and DI, also agreed on the same conclusion whether a black-box is discriminative or fair.

A possible future direction is to create a discrimination prevention model that decreases bias in black-box models detected by our different measures. Another possible direction is to study the relationship between discrimination and privacy. We would inspect the possibility of privacy leakage for instances in a protected subgroup even when the protected feature is hidden.

VII. Acknowledgment

The research reported herein was supported in part by NIH awards 1R0-1LM009989 & 1R01HG006844, NSF CNS-1111529, CNS-1228198, & CICI-1547324.

References

[1] D. Pedreschi, S. R. Franco, and Turini, "Discrimination-aware data mining." KDD, 2008.

[2] D. Pedreschi, S. Ruggieri, and F. Turini, "Measuring discrimination in socially-sensitive decision records." In Proceedings of SIAM DM, 2009.

[3] K. Mancuhan and C. Clifton, "Combating discrimination using bayesian networks," vol. 22, no. 2. Artificial Intelligence and Law, 2014, pp. 211–238.

[4] B. T. Luong, S. Ruggieri, and F. Turini, "k-nn as an implementation of situation testing for discrimination discovery and prevention," vol. 21, no. 2. KDD, 2011, p. 502510.

[5] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact." KDD, 2015.

[6] K. Mancuhan and C. Clifton, "Discriminatory decision policy aware classification." ICDM Workshop, 2012.

[7] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," vol. 21, no. 2. Data Mining and Knowledge Discovery, 2010, p. 277292.

[8] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning." ICDM, 2010.

[9] L. Sweeney, "Discrimination in online ad delivery," vol. 56, no. 5. Communications of the ACM, 2013, p. 4454.

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

[11] P. Adler, C. Falk, S. A. Friedler, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian, "Auditing black-box models by obscuring features." arXiv:1602.07043v1, 2016.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier." ACM SIG KDD, 2016.

[13] "United states census bureau." [Online]. Available: http://www.census.gov/en.html

[14] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning." KDD, 2001.

[15] G. H. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix." SIAM, 1995.

[16] "U.S. Federal Legislation." [Online]. Available: http://www.usdoj.gov

[17] "The U.S. EEOC. uniform guidelines on employee selection procedures," March 2, 1979.

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

[19] ProPublica, "Macine bias," 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[20] C. DeNavas-Walt and B. D. Proctor, "Income and poverty in the United States: 2014," *U.S. Census Bureau*, p. 60252, 2015.