Fooling SHAP with Stealthily Biased Sampling.

Gabriel Laberge
Polytechnique Montréal, Québec
gabriel.laberge@polymtl.ca

Ulrich Aïvodji
École de technologie supérieure, Québec
ulrich.aivodji@etsmtl.ca

Satoshi Hara
Osaka University, Japan
satohara@ar.sanken.osaka-u.ac.jp

Abstract

SHAP explanations aim at identifying which features contribute the most to the difference in model prediction at a specific input versus a background distribution. Recent studies have shown that they can be manipulated by malicious adversaries to produce arbitrary desired explanations. However, existing attacks focus solely on altering the black-box model itself. In this paper, we propose a complementary family of attacks that leave the model intact and manipulate SHAP explanations using stealthily biased sampling of the data points used to approximate expectations w.r.t the background distribution. In the context of fairness audit, we show that our attack can reduce the importance of a sensitive feature when explaining the difference in outcomes between groups, while remaining undetected. These results highlight the manipulability of SHAP explanations and encourage auditors to treat post-hoc explanations with skepticism.

1 Introduction

As Machine Learning (ML) gets more and more ubiquitous in high-stake decision contexts (e.g., healthcare, finance, and justice), concerns about its potential to lead to discriminatory models are becoming prominent. The use of auditing toolkits [1, 2, 3] is getting popular to circumvent the use of unfair models. However, although auditing toolkits can help model designers in promoting fairness, they can also be manipulated to mislead both the end-users and external auditors. For instance, a recent study [4] has shown that malicious model designers can produce a benchmark dataset as fake “evidence” of the fairness of the model even though the model itself is unfair.

Another approach to assess the fairness of ML systems is to explain their outcome in a *post hoc* manner [5]. For instance, SHAP [6] has risen in popularity as a means to extract model-agnostic local feature attributions. Feature attributions are meant to convey how much the model relies on certain features to make a decision at some specific input. The use of feature attributions for fairness auditing is desirable for the cases when the interest is on the direct impact of the sensitive attributes on the output of the model. One such situation is in the context of causal fairness [7]. In some practical cases, the outputs cannot be independent from the sensitive attribute unless we sacrifice much of prediction accuracy. For example, any decisions based on physical strength are statistically correlated to gender due to biological nature. The problem in such a situation is not the statistical bias (such as demographic parity), but whether the decision is based on the physical strength or gender, i.e. the attributions of each feature.

The focus of this study is on manipulating the feature attributions so that the dependence on sensitive attributes is hidden and the audits are misleading as if the model is fair even if it is not the case. Recently, several studies reported that such a manipulation is possible, *e.g.* by modifying the

Preprint. Under review.

black-box model to be explained [8, 9, 10], by manipulating the computation algorithms of feature attributions [11, 12], and by poisoning the data distribution [13]. With these findings in mind, the current possible advice to the auditors is not to rely solely on the reported feature attributes for fairness auditing. A question then arises about what "evidence" we can expect in addition to the feature attributions, and whether they can be valid "evidence" of fairness.

In this study, we show that we can craft fake "evidence" of fairness for SHAP explanations, which provides the first negative answer to the last question. In particular, we show that we can produce not only manipulated feature attributions but also a benchmark dataset as the fake "evidence" of fairness. The benchmark dataset ensures the external auditors reproduce the reported feature attributions using the existing SHAP library. In our study, we leverage the idea of stealthily biased sampling introduced in [4] to cherry-pick which data points to be included in the benchmark. Moreover, the use of stealthily biased sampling allows us to keep the manipulation undetected by making the distribution of the benchmark sufficiently close to the true data distribution. Figure 1 illustrates the impact of our attack in an explanation scenario with the Adult Income dataset.

Our contributions can be summarized as follows:

- Theoretically, we formalize a notion of foreground distribution that can be used to extend Local Shapley Values to Global Shapley Values (GSV), which can be used to decompose fairness metrics among the features (**Section 2.2**). Moreover, we formalize the task of manipulating the GSV as a Minimum Cost Flow (MCF) problem (**Section 4**).
- Experimentally (**Section 5**), we illustrate the impact of the proposed manipulation attack on a synthetic dataset and two popular datasets, namely Adult Income and COMPAS. We observed that the proposed attack can reduce the importance of a sensitive feature while keeping the data manipulation undetected by the audit.

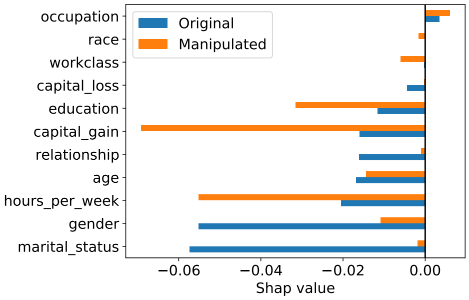


Figure 1: Example of our attack on the Adult Income dataset. After the attack, the feature gender moved from the 2^{nd} most negative attribution to the 5^{th}, hence hiding some of the explicit model bias.

Our results indicate that SHAP explanation is not robust and can be manipulated when it comes to explaining the difference in outcomes between groups. Even worse, our results confirm we can craft a benchmark dataset so that the manipulated feature attributions are reproducible by the external audits. Henceforth, we alert that auditors to treat post-hoc explanations methods with skepticism even if it is accompanied by some additional evidences.

2 Shapley Values

Before introducing the proposed attack on post-hoc explanations, we take the time to discuss Shapley values in detail.

2.1 Local Shapley Values

Shapley value are omnipresent in post-hoc explainability because of their fundamental mathematical properties [14] and their implementation in the popular SHAP Python library [6]. SHAP provides local explanations in the form of feature attributions i.e. given an input of interest $\boldsymbol{x}$, SHAP returns a score ϕ_i for each feature $i = 1, 2, \ldots, d$. These scores are meant to convey how much the model f relies on feature i to make its decision $f(\boldsymbol{x})$. Shapley values have a long background in coalitional game theory, where multiple players collaborate toward a common outcome. In the context of explaining model decisions, the players are the input features and the common outcome is the model output $f(\boldsymbol{x})$ which we are trying to explain. In coalitional games, players (features) are either present or absent. Since one cannot physically remove an input feature once the model has already been fitted, SHAP removes features by averaging them out. Formally, given an input of interest $\boldsymbol{x}$, a subset of features $S \subset \{1, 2, \ldots, d\}$ that are considered active, and a **random** input $\boldsymbol{z}$, the replace function

$\boldsymbol{r}_S : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ defined as

$$r_S(\boldsymbol{z}, \boldsymbol{x})_i = \begin{cases} x_i & \text{if } i \in S \\ z_i & \text{otherwise.} \end{cases} \tag{1}$$

is used to simulate the action of activating features in S. The random inputs $\boldsymbol{z}$ are sampled from a distribution $\mathcal{B}$ colloquially referred to as the *background*, and are typically chosen as the empirical distribution over the training set. Now, if we let π be a random permutation of d features, and π_i denote all features that appear before i in π, the Shapley values are computed via

$$\phi_i(f, \boldsymbol{x}, \mathcal{B}) = \mathop{\mathbb{E}}_{\substack{\pi \sim \Omega \\ \boldsymbol{z} \sim \mathcal{B}}} \big[\, f(\, \boldsymbol{r}_{\pi_i \cup \{i\}}(\boldsymbol{z}, \boldsymbol{x})\,) - f(\, \boldsymbol{r}_{\pi_i}(\boldsymbol{z}, \boldsymbol{x})\,)\big], \tag{2}$$

where Ω is the uniform distribution over 2^d permutations. One of the desirable theoretical properties satisfied by the Shapley values is the so-called efficiency axiom [6], which ensures that

$$\sum_{i=1}^{d} \phi_i(f, \boldsymbol{x}, \mathcal{B}) = f(\boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathcal{B}}[f(\boldsymbol{z})]. \tag{3}$$

Simply put, the difference between the model prediction at $\boldsymbol{x}$ and the average prediction across the background is shared among the different features.

2.2 Global Shapley Values

The previous definitions only apply to a specific $\boldsymbol{x}$ at which one wishes to explain the output $f(\boldsymbol{x})$. However, for auditing model fairness and detecting biases, a more global analysis is required. To do so, taking inspiration from Begley et al. [9], we can compute Global Shapley Values (GSV) by averaging Local Shapley Values $\boldsymbol{\phi}(f, \boldsymbol{x}, \mathcal{B})$ over inputs $\boldsymbol{x}$ sampled from a distribution $\mathcal{F}$ called the foreground

$$\boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{F}}[\boldsymbol{\phi}(f, \boldsymbol{x}, \mathcal{B})]. \tag{4}$$

Proposition 2.1. *The GSV have the following property*

$$\sum_{i=1}^{d} \Phi_i(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{B}}[f(\boldsymbol{x})]. \tag{5}$$

The proof is given in Appendix A.1.

2.3 Monte-Carlo Estimates

In practice, computing the expectations w.r.t the whole background and foreground distributions may be prohibitive and hence Monte-Carlo estimates are used. For instance, when a dataset is used to represent a background distribution, many explainers in the `SHAP` library such as the `ExactExplainer` and `PermutationExplainer` will subsample this dataset[1] by selecting 100 instances uniformly at random when the size of the dataset exceeds 100. Even explainers that are advertised as exact such as the `TreeExplainer` still encourage users to approximate expectations w.r.t the background when its size is too large [2].

More formally, let

$$\mathcal{C}(S, \boldsymbol{\omega}) = \sum_{\boldsymbol{x}^{(j)} \in S} \omega_j \delta(\boldsymbol{x}^{(j)}) \tag{6}$$

represent a categorical distribution over a finite set of input examples S, where $\delta(\cdot)$ is the dirac probability measure, $w_j \geqslant 0 \ \forall j$, and $\sum_j \omega_j = 1$. Estimating expectations with Monte-Carlo amounts to sampling M instances

$$S_1 \sim \mathcal{B}^M \qquad S_0 \sim \mathcal{F}^M, \tag{7}$$

and compute the estimates

$$\widehat{\boldsymbol{\Phi}}(f, S_0, S_1) := \boldsymbol{\Phi}(f, \, \mathcal{C}(S_0, \mathbf{1}/M), \, \mathcal{C}(S_1, \mathbf{1}/M)) \approx \boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B}). \tag{8}$$

The estimates $\widehat{\boldsymbol{\Phi}}$ are employed by practitionners as the model explanation which we see as a vulnerability. As discussed in Section 4, the Monte-Carlo estimation is the key ingredient that allow us to manipulate the Shapley values in favor of a dishonest entity.

[1] `https://github.com/slundberg/shap/blob/0662f4e9e6be38e658120079904899cccda59ff8/shap/maskers/_tabular.py#L54-L55`

[2] `https://github.com/slundberg/shap/blob/0662f4e9e6be38e658120079904899cccda59ff8/shap/explainers/_tree.py#L142-L144`

3 Audit Scenario

This section introduces an audit scenario to which the proposed attack of SHAP can apply. This scenario involves two parties: a company and an audit. The company has a dataset $D = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ with $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{0,1\}$ that contains N input-target tuples and also has a model $f : \mathcal{X} \rightarrow [0,1]$ that is meant to be deployed in society. The binary feature with index s (i.e. $x_s \in \{0,1\}$) represents a sensitive feature with respect to which the model should not explicitly discriminate. Both the data D and the model f are highly private so the company is very careful when providing information about them to the audit. Hence, f is a black box from the point of view of the audit.

At first, the audit asks the company for the necessary data to compute fairness metrics of the black box. To compute the demographic parity

$$\mathbb{E}[f(\boldsymbol{X})|X_s = 0] - \mathbb{E}[f(\boldsymbol{X})|X_s = 1], \tag{9}$$

the audit requires access to the model outputs for all inputs with different values of the sensitive feature i.e $f(D_0)$ and $f(D_1)$, where $D_0 = \{\boldsymbol{x}^{(i)} : x_s^{(i)} = 0\}$ and $D_1 = \{\boldsymbol{x}^{(i)} : x_s^{(i)} = 1\}$ are subsets of the input data of sizes N_0 and N_1 respectively. Doing so does not force the company to share values of features other than x_s nor does it requires direct access to the inner workings of the proprietary model. Hence this demand respects privacy requirements and the company will accept to share the model outputs across all instances, see Figure 2a. At this point, the audit confirms that the model is indeed biased in favor of $x_s = 1$ and puts in question the ability of the company to deploy such a model.

Now, the company argues that, although the model exhibits a disparity in outcomes, it does not mean that the model explicitly uses the feature x_s to make its decision. If such is the case, then the disparity could be explained by other features statistically associated with x_s. Some of these other features may be acceptable basis for decisions.

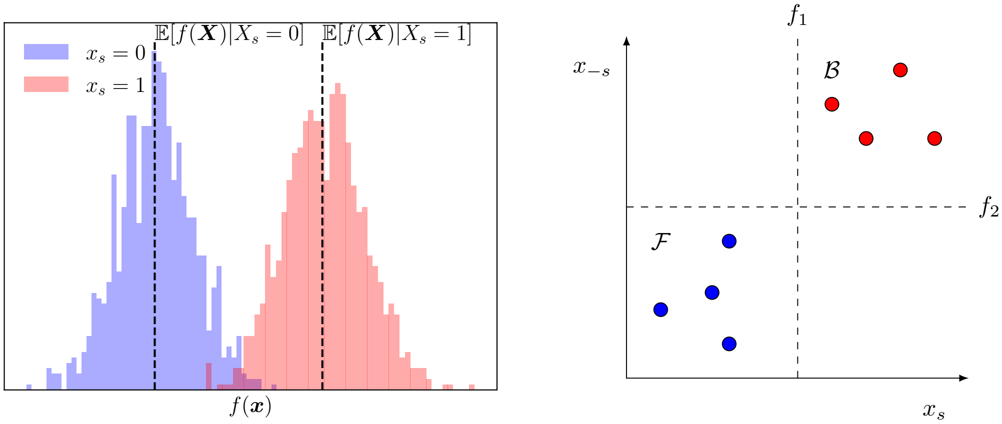
To verify such a claim, the audit decides to employ post-hoc techniques to explain the disparity. Since the model is a black-box, the audits shall compute the GSVs. The background $\mathcal{B}$ and foreground $\mathcal{F}$ are chosen to be the data distributions conditioned on $x_s = 1$ and $x_s = 0$ respectively

$$\mathcal{B} := \mathcal{C}(D_1, \mathbf{1}/N_1) \qquad \mathcal{F} := \mathcal{C}(D_0, \mathbf{1}/N_0). \tag{10}$$

According to Equation 5, the resulting GSVs will sum up of the demographic parity (cf Equation 9). If the sensitive feature has a large negative GSV Φ_s, then this would mean that the model is **explicitly** relying on x_s to make its decisions and the company would be forbidden from deploying the model. If the GSV has a small amplitude, however, the company could still argue in favour of deploying the model in spite of having disparate outcomes. Indeed, the difference in outcomes by the model could be attributed to other more acceptable features. See Figure 2b for a toy example illustrating this reasoning.

To compute the GSV, the audit demands the two datasets of inputs D_0 and D_1, as well as the ability to query the black box f at arbitrary points. Because of privacy concerns on sharing values of $\boldsymbol{x}$ across the whole dataset, and because GSV must be estimated with Monte-Carlo, both parties agree that the company shall only provide subsets $S_0 \subset D_0$ and $S_1 \subset D_1$ of size M to the audit so they can compute a Monte-Carlo estimate $\widehat{\boldsymbol{\Phi}}(f, S_0, S_1)$.

The company first estimate GSV on their own by choosing S_0, S_1 uniformly at random from $\mathcal{F}$ and $\mathcal{B}$ (cf Equation 7) and observe that $\widehat{\Phi}_s$ indeed has a large negative value. They realise they must be careful which data points they send to the audit otherwise they will see that the model has an explicit bias toward $x_s = 1$ and the model will not be deployed. Moreover, the company understands that the audit currently has access to the data $f(D_0)$ and $f(D_1)$ representing the model predictions on the whole dataset (see Figure 2a). Therefore, if the company does not share subsets S_0, S_1 that where chosen uniformly at random from D_0, D_1, it is possible for the audit to detect this fraud by doing a statistical test comparing $f(S_0)$ to $f(D_0)$ and $f(S_1)$ to $f(D_1)$. The company needs a method to select **misleading subsets** S_0', S_1' in a manner that manipulates the GSV in their favour, while remaining undetected by the audit. Such a method is the subject of the next section.



(a) The data initially provided by the company to the audit is $f(D_0)$ and $f(D_1)$ i.e. the model predictions for all instances in the private dataset for different values of x_s. This dataset can later be used by the audit to assess whether or not the subsets S_0', S_1' provided by the company where cherry-picked.

(b) Two models f_1 and f_2 (decision boundaries in dashed lines) with perfect accuracy exhibit a disparity in outcomes w.r.t groups with $x_s < 0$ and $x_s > 0$. Here, $\Phi_s(f_1, \mathcal{F}, \mathcal{B}) = 1$ while $\Phi_s(f_2, \mathcal{F}, \mathcal{B}) = 0$. Hence, f_2 is **indirectly** unfair toward x_s because of correlations in the data.

Figure 2: Illustrations of the audit scenario.

4 Fooling SHAP with Stealthily Biased Sampling

For simplicity, we shall refer to the jth instances from D_1 as $\boldsymbol{z}^{(j)}$ and ith instances from D_0 as $\boldsymbol{x}^{(i)}$.

4.1 Manipulation

To fool the audit, the company can decide to indeed sample S_0' uniformly at random i.e. $S_0' \sim \mathcal{F}^M$. Then, given this choice of foreground sub-sample, they can manipulate the background distribution to cherry-pick M instances. Formally, the company must compute a non-uniform background distribution $\mathcal{B}' := \mathcal{C}(D_1, \boldsymbol{\omega})$ with $\boldsymbol{\omega} \neq \mathbf{1}/N_1$ from which to sample M points $S_1' \sim \mathcal{B}'^M$. To make the model look fairer, the company needs the $\widehat{\Phi}_s$ computed with these cherry-picked points to have a small magnitude.

Hence, manipulating the GSV is done by computing non-uniform weights $\boldsymbol{\omega} \neq \mathbf{1}/N_1$ of the background distribution. The critical observation to motivate our approach is that the estimated GSV converges in probability to a **linear** function of these weights.

Proposition 4.1. *Let S_0' be **fixed**, and let $\xrightarrow{p}$ represent convergence in probability as the size M of the set $S_1' \sim \mathcal{B}'^M$ increases, we have*

$$\widehat{\Phi}_s(f, S_0', S_1') \xrightarrow{p} \sum_{j=1}^{N_1} \omega_j \, \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}). \tag{11}$$

The Proof is given in Appendix A.1.

We note that the coefficients $\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})$ in Equation 11 are tractable and can be computed and stored by the company. We discuss in more detail how to compute them in supplementary materials.

Remember that to manipulate the GSV, the company must tune the weights $\boldsymbol{\omega}$ such that the explanation $\widehat{\Phi}_s$ is manipulated but ensure that the manipulated distribution $\mathcal{B}'$ remains *similar* to the original $\mathcal{B}$. Otherwise, the fraud could be detected by the audit. Here the notion of similarity between distributions will be captured by the Wasserstein distance in output space.

Definition 4.1 (Wassertein Distance)**.** *Any probability measure π over $D_1 \times D_1$ is called a coupling measure between $\mathcal{B}$ and $\mathcal{B}'$, denoted $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$, if $1/N_1 = \sum_j \pi_{ij}$ and $\omega_j = \sum_i \pi_{ij}$. The Wassertein distance between $\mathcal{B}$ and $\mathcal{B}'$ mapped to the output-space is defined as*

$$\mathcal{W}(f(\mathcal{B}), f(\mathcal{B}')) = \min_{\pi \in \Delta(\mathcal{B}, \mathcal{B}')} \sum_{i,j} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \pi_{ij}, \tag{12}$$

representing the cost of the optimal transport plan that distributes the probability mass from one distribution to the other.

Given this definition, we propose the following optimization problem, where a hyper-parameter λ trades off manipulating the GSV with proximity to the original background.

Definition 4.2. *The problem of manipulating the GSV of a sensitive feature s while remaining hard to detect by the audit is*

$$\begin{aligned} \min_{\boldsymbol{\omega}} \quad & -\sum_{j=1}^{N_1} \omega_j \, \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \mathcal{W}(f(\mathcal{B}), f(\mathcal{B}')), \\ \textit{with} \quad & \mathcal{B}' = \mathcal{C}(D_1, \boldsymbol{\omega}). \end{aligned} \tag{13}$$

We show in the Appendix A.2 that this linear program can be reformulated as a Minimum Cost Flow and hence can be solved in polynomial time (more precisely $\widetilde{\mathcal{O}}(N_1^{2.5})$. as in [4]).

4.2 Detection

We now discuss ways the audit can detect manipulation of the sampling procedure done by the company. Recall that the audit has previously been given access to $f(D_1), f(D_0)$ representing the model outputs across all instances in the private dataset. The audit will then be provided sub-samples S_1', S_0' of D_1, D_0 on which they can compute the output of the model on compare with $f(D_1), f(D_0)$. To assess whether or not the sub-samples provided by the company were sampled uniformly at random, the audit has to conduct statistical tests. The null hypothesis of these tests will be that S_1', S_0' were sampled uniformly at random from D_1, D_0 and hence p-values smaller than a pre-specified level of significance α will lead to the detection of a fraud.

4.2.1 KS test

A first test that can be conducted is a two-samples Kolmogorov-Smironov (KS) test [15]. If we let

$$\widehat{F}_S(x) = \frac{1}{|S|} \sum_{z \in S} \mathbb{1}(z \leqslant x) \tag{14}$$

be the empirical Cumulative Distribution Function (CDF) of observations in the set S. Given two sets S and S', the KS statistic is

$$\text{KS}(S, S') = \sup_{x \in \mathbb{R}} |\widehat{F}_S(x) - \widehat{F}_{S'}(x)|. \tag{15}$$

Under the null-hypothesis $H_0 : S \sim \mathcal{D}^{|S|}, S' \sim \mathcal{D}^{|S'|}$ for some univariate distribution $\mathcal{D}$, this statistic is expected to not be too large with high probability.

Hence, when the company provides the subsets S_0', S_1', the audit can sample their own two subsets $f(S_0), f(S_1)$ uniformly at random from $f(D_0), f(D_1)$ and compute the statistics $\text{KS}(f(S_1), f(S_1'))$ and $\text{KS}(f(S_0), f(S_0'))$ to detect a fraud.

4.2.2 Wald test

An alternative is the Wald test, which is based on the central limit theorem. If $S_1 \sim \mathcal{B}^M$, then the empirical average of the model output over S_1 is asymptotically normally distributed as M increases i.e.

$$\text{Wald}(f(S_1), f(\mathcal{B})) := \frac{\frac{1}{M}\sum_{z \in f(S_1)} z - \mu}{\sigma/\sqrt{M}} \rightsquigarrow \mathcal{N}(0, 1), \tag{16}$$

where $\mu := \mathbb{E}_{z \sim f(\mathcal{B})}[z]$ and $\sigma^2 := \mathbb{V}_{z \sim f(\mathcal{B})}[z]$ are the expected value and variance of the model output across the whole background. The same reasoning holds for S_0 and the foreground $\mathcal{F}$.

Applying the Wald test with significance α would detect a fraud when

$$\text{Wald}(f(S_1'), f(\mathcal{B})) > F^{-1}_{\mathcal{N}(0,1)}(1 - \alpha/2), \tag{17}$$

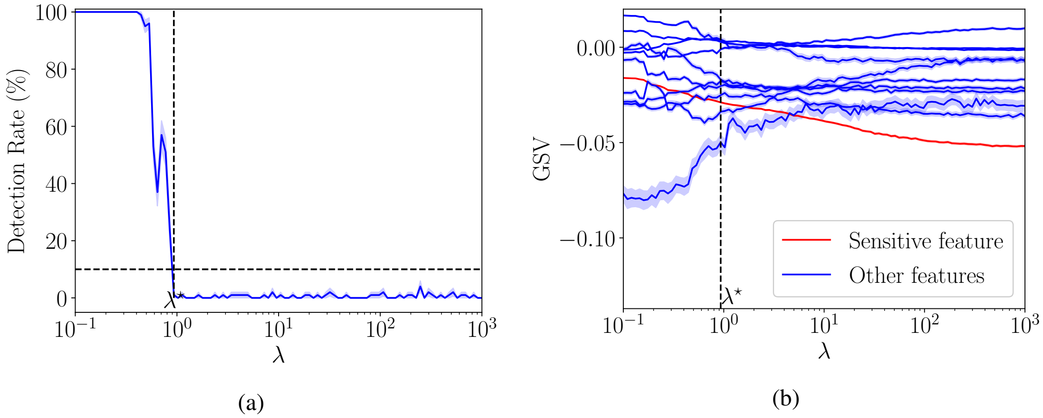


Figure 3: Example of log-space search over values of λ using a XGBoost classifier fitted on Adults. (a) The detection rate as a function of the parameter λ of the attack. The attacker uses a detection rate threshold $\tau = 10\%$. (b) For each value of λ, the vertical slice of the 11 curves is the GSV obtained with the resulting biased background distribution. The goal here is to reduce the amplitude of the sensitive feature (red curve) in order to hide its direct effect when explaining the disparity in model outcomes.

where $F^{-1}_{\mathcal{N}(0,1)}$ is the inverse-CDF of the standard normal distribution.

The detection algorithm used by the audit is presented in appendix, see Algorithm 2. This algorithm uses both the KS and Wald tests with Bonferri corrections to bound the probability of false positives (i.e. the probability that the audit detects a fraud when there is none).

4.3 Whole procedure

For the attacker to provide subsets S'_0, S'_1 to the audit, they need to fine-tune the hyperparameter λ of Definition 4.2. For simplicity, we assume that the attacker has access to the detection tool that is going to be employed by the audit. We propose to use a log-space search between λ_{min} and λ_{max}, and for each value of λ, the attacker computes manipulated GSVs $\widehat{\Phi}_s(f, S'_0, S'_1)$ and attempts to detect the fraud by repeatedly sampling $S'_1 \sim \mathcal{B}'^M$. The attacker will choose the value of manipulated background weights that reduces the magnitude of $\widehat{\Phi}_s$ the most while having a detection rate below some threshold τ. The whole procedure is presented in Algorithm 1 in supplementary. This algorithm returns the subsets S'_0, S'_1 that will be provided to the audit to compute the final GSV. An example of search over λ on a real-world dataset is presented in Figure 3.

4.4 Limitations

We discuss some of the limitations of the manipulation of Shapley values.

1. Given our formulation of the problem (Definition 4.2), we have a guarantee of reducing the magnitude of $\widehat{\Phi}_s$, but not $\widehat{\Phi}_i$ for other features i that could be considered sensitive. That is, if a dataset contains several sensitive features, we cannot reduce all of them simultaneously using a linear program.

2. Our choice of Wasserstein distance $\mathcal{W}(f(\mathcal{B}), f(\mathcal{B}'))$ considers uni-variate distributions in the output space of the model. We provide no guarantee that the distributions $\mathcal{B}$ and $\mathcal{B}'$ will be similar in input space. Therefore, we assume that the audit has very limited access to the distributions of input features across the whole dataset. This does not seem like a far-fetched assumption in practice since companies are very concerned about the privacy of their datasets.

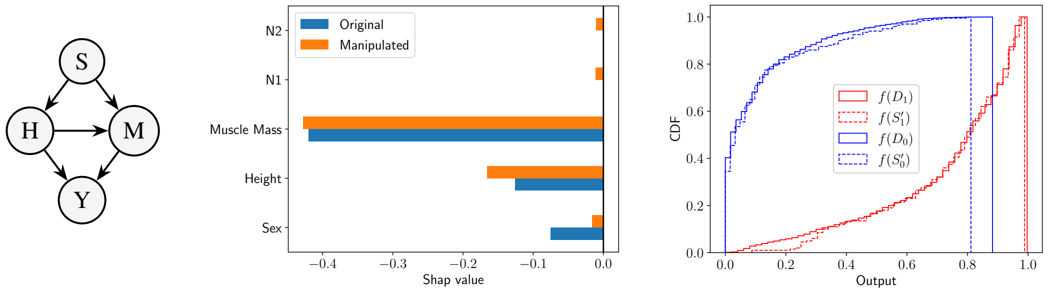


Figure 4: Toy example of hiring data for a job with specific physical requirements. Left: Causal graph. Middle: GSV before and after the attack with $M = 200$. Right: Comparison of the CDF of the misleading subsets $f(S'_0), f(S'_1)$ and the CDF over the whole data. $f(D_0), f(D_1)$. Here the audit is not able to detect the fraud using their detection algorithm.

5 Experiments

Code to reproduce the following experiments is available online[3].

5.1 Toy experiment

We start with a simple toy example to show how the attack works. The task considered was about predicting whether or not an individual is going to be hired for a job that requires carrying heavy objects. The causal graph for this toy data is presented in Figure 4 (left). We observe that sex (S) influences height (H), and that both these features influence the Muscular Mass (M). In the end, the hiring decisions (Y) are only based on the two attributes relevant to the job: H and M. To slightly complexity the problem, two noise features $N1, N2$ were added. More details and justifications for this causal graph are discussed in Appendix D.

Since strength and height (the two important qualifications for applicants) are correlated with sex, any model f that fits the data will exhibit some disparity in hiring rates between sexes. Although, if the decisions made by the model do not rely strongly on feature S, the company can argue in favor of deployment. GSVs are used by the audit to measure the amount by which the model relies on the sex feature, see Figure 4 (Middle). By manipulating `SHAP` with $M = 200$, the company is able to reduce the amplitude of the GSV of feature S. More importantly, the audit is not able to detect that the provided samples S'_0, S'_1 were cherry-picked, see Figure 4 (Right).

5.2 Datasets

Two real-world datasets were investigated, Adult-Income and COMPAS, which are often studied in the Fairness literature.

- **COMPAS** regroups 6,150 records from criminal offenders in Florida collected from 2013 and 2014. This binary classification task consists in predicting which individual will re-offend within two years. The sensitive feature s is `race` with values $x_s = 0$ for African-Americans and $x_s = 1$ for Caucasians.
- **Adult Income** contains demographic attributes of 48,842 individuals from the 1994 U.S. census. It is a binary classification problem with the goal of predicting whether or not a particular person makes more than 50K USD per year. The sensitive feature s if this dataset is `gender`, which took values $x_s = 0$ for female, and $x_s = 1$ for male.

Three models were considered for the two datasets: Multi-Layered Perceptrons (MLP), Random Forests (RF), and eXtreme Boosted Trees (XGB). One model of each type was fitted on the datasets for 5 different train/test splits seeds, resulting in 30 models total. Values of test set accuracy's and the demographic parities for each model type and dataset are presented in Appendix D.

[3] `https://github.com/gablabc/Fool_SHAP`

Table 1: Probability of False Positives (%) by the Detector.

	mlp	rf	xgb
COMPAS	0.96	0.82	0.66
Adult	0.92	0.82	1

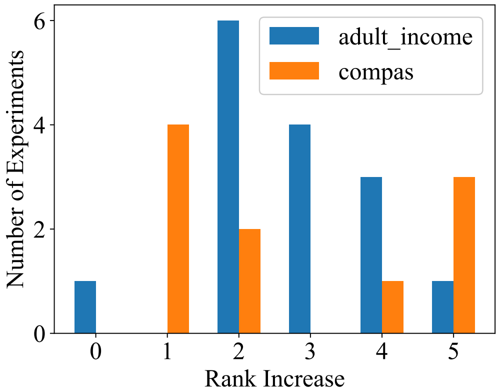


Figure 5: Increase in rank of the GSV $\widehat{\Phi}_s$ induced by the attack (i.e. $\text{rank}_{\text{after attack}} - \text{rank}_{\text{before attack}}$). Note that ranks consider the sign of the feature attribution hence a small rank (e.g. 1, 2) implies that the feature has the most negative effect on demographic parity. Increasing the rank of Φ_s means that the true negative effect of the sensitive is being masked by other features.

5.3 Detector Calibration

Detector calibration refers to the assessment that, assuming the null hypothesis to be true, the probability of rejecting it (i.e. false positive) should be bounded by the significance level α. Remember that the null hypothesis is that the sets S_0', S_1' provided by the company are sampled uniformly from D_0, D_1. Hence, to test the detector, the audit can sample their own subsets $f(S_0), f(S_1)$ uniformly from $f(D_0), f(D_1)$, run the detection algorithm, and count the number of defections over 1000 repeats. Table 1 presents the probabilities of false positives over the five train-test splits using a significance level $\alpha = 1\%$. We observe that the probability of false positives is indeed bounded by α for all model types and datasets. Hence. we can safely assess that the detector employed by the audit is calibrated.

5.4 Attack Results and discussion

The attack was conducted on all 30 models and the `ExactExplainer` of `SHAP` was used by the company to extract the necessary coefficients to run the linear program of Definition 4.2. As stated previously, solving the linear program can be done in polynomial time in terms of N_1, which was tractable for a small dataset like COMPAS, but not for a large dataset like Adult. To solve this issue, as was done in [4], we compute the manipulated weights multiple times using 5 bootstrap sub-samples of D_1 of size 2000 to obtain a set of weights $\boldsymbol{\omega}^{[1]}, \boldsymbol{\omega}^{[2]}, \ldots, \boldsymbol{\omega}^{[5]}$ which we average to obtain the final weights $\boldsymbol{\omega}$. Results of the attack for $M = 200$ are shown in Figure 5. Notice that only 10 attacks are presented on COMPAS while 15 were done on Adult-Income. The reason is that for 5 models trained on COMPAS, the GSV of *race* was already negligible compared to other features and hence conducting the attack was not required. We note that in all experiments but one, we are able to increase the rank of the GSV $\widehat{\Phi}_s$ hence masking the negative effect of the sensitive feature by other features. In none of these experiments, the audit was able to detect the fraud using statistical tests. This observation raises concerns about the risk that SHAP explanations can be attacked to return not only manipulated attributions but also non-detectable fake evidence of fairness.

6 Conclusion

To conclude, we proposed a novel attack on Shapley values that does not require modifying the model but rather manipulates the sampling procedure that estimates expectations w.r.t the background distributions. We show on a toy example and two fairness datasets that our attack can reduce the perceived importance of a sensitive feature when explaining the difference in outcomes between groups using SHAP. Moreover, the sampling manipulation is hard to detect by an audit that is given limited access to the data and model. These results raise concerns about the viability of using Shapley values to assess model fairness.

We leave as future work the use of Shapley values to decompose other fairness metrics such as predictive equality, equal opportunity etc. Moreover, we wish to move to use-cases beyond fairness, as we believe that the vulnerability of Shapley values that was demonstrated can apply to many other fields.

References

[1] Julius A Adebayo et al. Fairml: Toolbox for diagnosing bias in predictive modeling. Master's thesis, Massachusetts Institute of Technology, 2016.

[2] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018.

[3] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[4] Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. Faking fairness via stealthily biased sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 412–419, 2020.

[5] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[6] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[7] Yoichi Chikahara, Shinsaku Sakaue, Akinori Fujino, and Hisashi Kashima. Learning individually fair classifier with path-specific causal-effect constraint. In *International Conference on Artificial Intelligence and Statistics*, pages 145–153. PMLR, 2021.

[8] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[9] Tom Begley, Tobias Schwedes, Christopher Frye, and Ilya Feige. Explainability for fair machine learning. *arXiv preprint arXiv:2010.07389*, 2020.

[10] Botty Dimanov, Umang Bhatt, Mateja Jamnik, and Adrian Weller. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods. In *SafeAI@ AAAI*, 2020.

[11] Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *International Conference on Machine Learning*, pages 161–170. PMLR, 2019.

[12] Ulrich Aïvodji, Hiromi Arai, Sébastien Gambs, and Satoshi Hara. Characterizing the risk of fairwashing. *Advances in Neural Information Processing Systems*, 34, 2021.

[13] Hubert Baniecki, Wojciech Kretowicz, and Przemyslaw Biecek. Fooling partial dependence via data poisoning. *arXiv preprint arXiv:2105.12837*, 2021.

[14] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, pages 307–317, 1953.

[15] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[16] Larry Wasserman. *All of Statistics: A concise course in statistical inference*. Springer, 2004.

[17] Ian Janssen, Steven B Heymsfield, ZiMian Wang, and Robert Ross. Skeletal muscle mass and distribution in 468 men and women aged 18–88 yr. *Journal of applied physiology*, 2000.

A Proofs

A.1 Proofs for GSV

Proposition A.1 (Proposition 2.1). *The GSV have the following property*

$$\sum_{i=1}^{d} \Phi_i(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{B}}[f(\boldsymbol{x})]. \tag{18}$$

Proof. As a reminder, we have defined the vector

$$\boldsymbol{\Phi}(f, \mathcal{F}, \mathcal{B}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[\boldsymbol{\phi}(f, \boldsymbol{x}, \mathcal{B})], \tag{19}$$

whose components sum up to

$$\begin{aligned}
\sum_{i=1}^{d} \Phi_i(f, \mathcal{F}, \mathcal{B}) &= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[\phi_i(f, \boldsymbol{x}, \mathcal{B})] && (20)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}\left[\sum_{i=1}^{d} \phi_i(f, \boldsymbol{x}, \mathcal{B})\right] && (21)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}\left[f(\boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{B}}[f(\boldsymbol{z})]\right] && (22)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{B}}[f(\boldsymbol{z})] && (23)\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{F}}[f(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{B}}[f(\boldsymbol{x})], && (24)
\end{aligned}$$

where at the last step we have simply renamed a dummy variable. □

Proposition A.2 (Proposition 4.1). *Let S'_0 be* ***fixed****, and let $\xrightarrow{p}$ represent convergence in probability as the size M of the set $S'_1 \sim \mathcal{B}'^M$ increases, we have*

$$\widehat{\Phi}_s(f, S'_0, S'_1) \xrightarrow{p} \sum_{j=1}^{N_1} \omega_j\, \widehat{\Phi}_s(f, S'_0, \{\boldsymbol{z}^{(j)}\}). \tag{25}$$

Proof.

$$\begin{aligned}
\widehat{\Phi}_s(f, S'_0, S'_1) &:= \Phi_s(f, \mathcal{C}(S'_0, \mathbf{1}/M), \mathcal{C}(S'_1, \mathbf{1}/M))\\
&= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{C}(S'_0,\mathbf{1}/M)}\left[\mathop{\mathbb{E}}_{\substack{\pi\sim\Omega\\ \boldsymbol{z}\sim\mathcal{C}(S'_1,\mathbf{1}/M)}}\big[f(\,\boldsymbol{r}_{\pi_s\cup\{s\}}(\boldsymbol{z},\boldsymbol{x})\,) - f(\,\boldsymbol{r}_{\pi_s}(\boldsymbol{z},\boldsymbol{x})\,)\big]\right]\\
&= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{C}(S'_1,\mathbf{1}/M)}\left[\mathop{\mathbb{E}}_{\substack{\pi\sim\Omega\\ \boldsymbol{x}\sim\mathcal{C}(S'_0,\mathbf{1}/M)}}\big[f(\,\boldsymbol{r}_{\pi_s\cup\{s\}}(\boldsymbol{z},\boldsymbol{x})\,) - f(\,\boldsymbol{r}_{\pi_s}(\boldsymbol{z},\boldsymbol{x})\,)\big]\right]\\
&= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{C}(S'_1,\mathbf{1}/M)}\big[\widehat{\Phi}_s(f, S'_0, \{\boldsymbol{z}\})\big]
\end{aligned} \tag{26}$$

Since S'_0 is assumed to be fixed, then the only random variable in $\widehat{\Phi}_s(f, S'_0, \{\boldsymbol{z}\})$ is $\boldsymbol{z}$ which represents an instance sampled from the empirical distribution over S'_1. Therefore, we can define $\psi(\boldsymbol{z}) := \widehat{\Phi}_s(f, S'_0, \{\boldsymbol{z}\})$ and we get

$$\begin{aligned}
\widehat{\Phi}_s(f, S'_0, S'_1) &= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{C}(S'_1,\mathbf{1}/M)}\big[\widehat{\Phi}_s(f, S'_0, \{\boldsymbol{z}\})\big]\\
&= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathcal{C}(S'_1,\mathbf{1}/M)}\big[\psi(\boldsymbol{z})\big]\\
&= \frac{1}{M}\sum_{\boldsymbol{z}^{(j)}\in S'_1} \psi(\boldsymbol{z}^{(j)}) \qquad \text{with } S'_1 \sim \mathcal{B}'^M
\end{aligned} \tag{27}$$

By the weak law of large number, the following holds as M goes to infinity [16, Theorem 5.6]

$$\frac{1}{M}\sum_{\boldsymbol{z}^{(j)}\in S_1'}\psi(\boldsymbol{z}^{(j)}) \xrightarrow{p} \underset{\boldsymbol{z}\sim\mathcal{B}'}{\mathbb{E}}[\psi(\boldsymbol{z})] \tag{28}$$

Now, as a reminder, the manipulated background distribution is defined as $\mathcal{B}' := \mathcal{C}(D_1, \boldsymbol{\omega})$ with $\boldsymbol{\omega} \neq \mathbf{1}/N_1$. Therefore

$$\begin{aligned}\widehat{\Phi}_s(f, S_0', S_1') &\xrightarrow{p} \underset{\boldsymbol{z}\sim\mathcal{B}'}{\mathbb{E}}[\psi(\boldsymbol{z})]\\ &= \underset{\boldsymbol{z}\sim\mathcal{C}(D_1,\boldsymbol{\omega})}{\mathbb{E}}[\psi(\boldsymbol{z})]\\ &= \sum_{j=1}^{N_1}\omega_j\psi(\boldsymbol{z}^{(j)})\\ &= \sum_{j=1}^{N_1}\omega_j\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})\end{aligned} \tag{29}$$

concluding the proof. □

A.2 Proofs for Optimization Problem

A.2.1 Technical Lemmas

We provide some technical lemmas that will be essential when proving Theorem A.1. These lemmas and proofs are provided here for completeness and are not meant as contributions by the authors.

Let us first write the formal definition of the minimum of a function.

Definition A.1 (Minimum)**.** *Given some function $f : D \to \mathbb{R}$, the minimum of f over D (denoted $f^\star$) is defined as follows:*

$$f^\star = \min_{x\in D} f(x) \iff \exists x^\star \in D \text{ s.t. } f^\star = f(x^\star) \leqslant f(x) \quad \forall x \in D.$$

Basically, the notion of minimum coincides with the notion of infimum (highest lower bound) of $f(D)$ when this lower bound is attained for some $x^\star \in D$. For the rest of this appendix, we shall only study constrained optimization problems where points from the feasible set $D = \{(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}_x \subset \mathcal{Y}\}$ can be *selected* by the following procedure

1. Choose some $x \in \mathcal{X}$
2. Given the selected x, choose some $y \in \mathcal{Y}_x \subset \mathcal{Y}$ where the set $\mathcal{Y}_x$ is non-empty and depends on the value of x.

When optimizing objective functions on this types of domains, one can optimize in two steps as highlighted in the following lemma.

Lemma A.1. *Given a feasible set D of the form described above and an objective function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the following holds*

$$\min_{(x,y)\in D} f(x, y) = \min_{x\in\mathcal{X}}\min_{y\in\mathcal{Y}_x} f(x, y).$$

Proof. Let $\widetilde{f}(x) := \min_{y\in\mathcal{Y}_x} f(x, y)$, which is a well defined function on $\mathcal{X}$ since $\mathcal{Y}_x$ is non-empty for any $x \in \mathcal{X}$. By the definition of the minimum, we have

$$\forall x \in \mathcal{X}, \exists y^\star(x) \in \mathcal{Y}_x \quad \text{s.t.} \quad \widetilde{f}(x) = f(x, y^\star(x)) \leqslant f(x, y) \quad \forall y \in \mathcal{Y}_x. \tag{30}$$

Now, we can optimize $\widetilde{f}$ with respect to x i.e. $f^\star = \min_{x\in\mathcal{X}} \widetilde{f}(x)$. By applying once again the definition of the minimum, we get

$$\exists x^\star \in \mathcal{X} \quad \text{s.t.} \quad f^\star = \widetilde{f}(x^\star) \leqslant \widetilde{f}(x) \quad \forall x \in \mathcal{X}. \tag{31}$$

By virtue of Equation 30, we have that $\widetilde{f}(x^\star) = f(x^\star, y^\star(x^\star)) = f(x^\star, y^\star)$, where we labeled $y^\star := y^\star(x^\star)$ for convenience. We get

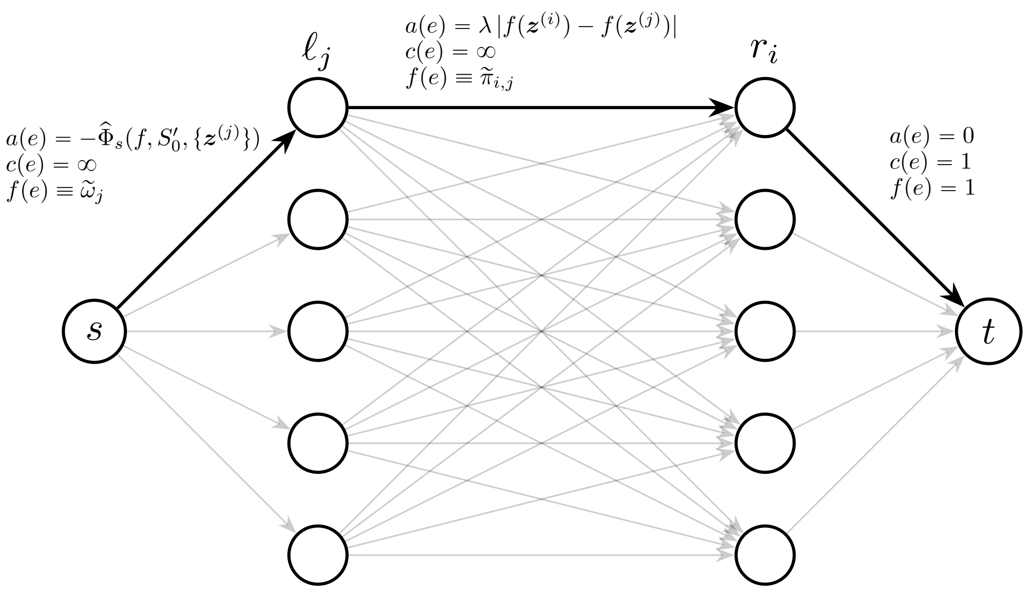
$$\begin{aligned} \exists (x^\star, y^\star) \in D \quad \text{s.t.} \quad f(x^\star, y^\star) &\leqslant f(x, y^\star(x)) \quad \forall\, x \in \mathcal{X} && \text{(cf. Equation 31)} \\ &\leqslant f(x, y) \qquad\quad \forall\, y \in \mathcal{Y}_x . && \text{(cf. Equation 30)} \end{aligned}$$

Hence we have proven that $\exists (x^\star, y^\star) \in D \quad \text{s.t.} \quad f(x^\star, y^\star) \leqslant f(x, y) \;\; \forall (x, y) \in D$, which concludes the proof. □

Lemma A.2. *Given a feasible set D of the form described above and two functions $h : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$, then*

$$\min_{(x,y)\in D} \Big(h(x) + g(y) \Big) = \min_{x\in\mathcal{X}} \Big(h(x) + \min_{y\in\mathcal{Y}_x} g(y) \Big)$$

Proof. Applying Lemma A.1 with the function $f(x, y) := h(x) + g(y)$ leads to the desired result. □



ℓ_j
$a(e) = \lambda\,|f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})|$
$c(e) = \infty$
$f(e) \equiv \widetilde{\pi}_{i,j}$
r_i
$a(e) = -\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})$
$c(e) = \infty$
$f(e) \equiv \widetilde{\omega}_j$
$a(e) = 0$
$c(e) = 1$
$f(e) = 1$
s
t

Figure 6: Graph $\mathbb{G}$ on which we solve the MCF. Note that the total amount of flow is $d = N_1$ and there are N_1 left and right nodes ℓ_j, r_i.

A.2.2 Minimum Cost Flows

Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be a graph with vertices $v \in \mathcal{V}$ with directed edges $e \in \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, $c : \mathcal{E} \to \mathbb{R}^+$ be a capacity and $a : \mathcal{E} \to \mathbb{R}$ be a cost. Moreover, let $s, t \in \mathcal{E}$ be two special vertices called the source and the sink respectively, and $d \in \mathbb{R}^+$ be a total flow. The Minimum-Cost Flow (MCF) problem of $\mathbb{G}$ consists of finding the flow function $f : \mathcal{E} \to \mathbb{R}^+$ that minimizes the total cost i.e.

$$\begin{aligned} \min_f \quad & \sum_{e \in \mathcal{E}} a(e) f(e) \\ \text{s.t.} \quad & 0 \leqslant f(e) \leqslant c(e) \ \forall e \in \mathcal{E} \\ & \sum_{e \in u^+} f(e) - \sum_{e \in u^-} f(e) = \begin{cases} 0 & u \in \mathcal{V} \backslash \{s, t\} \\ d & u = s \\ -d & u = t \end{cases} \end{aligned} \tag{32}$$

where $u^+ := \{(u, v) \in \mathcal{E}\}$ and $u^- := \{(v, u) \in \mathcal{E}\}$ are the outgoing and incoming edges from u respectively. The terminology of *flow* arises from the constraint that, for vertices that are not the source nor the sink, the outgoing flow must equal the incoming one, which is reminiscent of conservation laws in fluidic. We shall refer to $f((u, v))$ as the flow from u to v.

Now that we have introduced minimum cost flows, let us introduce the specific graph that will be employed to bias manipulate GSV, see Figure 6. We label the flow going from the sink s to one of the left vertices as $\widetilde{\omega}_i \equiv \omega_i / N_1$, and the flow going from ℓ_j to r_i as $\widetilde{\pi}_{i,j} \equiv \pi_{i,j} / N_1$. The required flow is fixed at $d = N_1$

Theorem A.1. *Solving the MCF of Figure 6 leads to a solution of the linear program in Definition 4.2.*

Proof. We begin by showing that the flow conservation constraints in the MCF imply that π is a coupling measure (i.e. $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$), and ω is constrained to the probability simplex $\Delta(N_1)$.

Applying the conservation law on the left-side of the graph leads to the conclusion that the whole flow that enters vertices ℓ_j must sum up to N_1, and hence ω is must be part of the probability simplex.

The total amount of flow that leaves a specific vertex ℓ_j must also be $\widetilde{\omega}_j$, hence

$$\sum_i \widetilde{\pi}_{ij} = \widetilde{\omega}_j.$$

For any edge going from r_i to the sink t, the flow must be 1 because of the conservation law on t and the fact that $c(e) = 1$. As a consequence of the conservation law on a specific vertex r_i, the amount of flow that goes into each r_i is 1 i.e.

$$\sum_j \widetilde{\pi}_{ij} = 1.$$

Putting everything together, from the conservation laws on $\mathbb{G}$, we have that $\omega \in \Delta(N_1)$, and $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$. To make the parallel between the MCF and Definition 4.2, we must use Lemma A.2. As a reminder, this Lemma states that for optimization problems on certain types of domains, one can solve the optimization problem in two optimization steps. Note that ω is restricted to the probability simplex, while π is restricted to be a coupling measure. However, the set of all possible coupling measures $\Delta(\mathcal{B}, \mathcal{B}')$ is different for each ω (and non-empty). Hence, we study a feasible set with the same structure as the ones tackled in the Lemma A.2 (where $x \in \mathcal{X}$ becomes $\omega \in \Delta(N_1)$) and $y \in \mathcal{Y}_x$ becomes $\pi \in \Delta(\mathcal{B}, \mathcal{B}')$) and we can apply the Lemma A.2 to the objective function of the MCF.

$$\begin{aligned}
\min_f \sum_{e \in \mathcal{E}} f(e)a(e) &= \min_{\widetilde{\omega}, \widetilde{\pi}} \sum_{j=1}^{N_1} -\widetilde{\omega}_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \sum_{i,j} \widetilde{\pi}_{ij} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \\
&= \min_{\widetilde{\omega}, \widetilde{\pi}} \frac{N_1}{N_1} \left(\sum_{j=1}^{N_1} -\widetilde{\omega}_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \sum_{i,j} \widetilde{\pi}_{ij} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\widetilde{\omega}, \widetilde{\pi}} \left(\sum_{j=1}^{N_1} \frac{-\widetilde{\omega}_j}{N_1} \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \sum_{i,j} \frac{\widetilde{\pi}_{ij}}{N_1} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\omega \in \Delta(N_1), \pi \in \Delta(\mathcal{B}, \mathcal{B}')} \left(\sum_{j=1}^{N_1} -\omega_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \sum_{i,j} \pi_{i,j} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\omega \in \Delta(N_1), \pi \in \Delta(\mathcal{B}, \mathcal{B}')} \Big(h(\omega) + g(\pi) \Big) \\
&= N_1 \min_{\omega \in \Delta(N_1)} \Big(h(\omega) + \min_{\pi \in \Delta(\mathcal{B}, \mathcal{B}')} g(\pi) \Big) \qquad \text{(cf Lemma A.2)} \\
&= N_1 \min_{\omega \in \Delta(N_1)} \left(\sum_{j=1}^{N_1} -\omega_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \min_{\pi \in \Delta(\mathcal{B}, \mathcal{B}')} \sum_{i,j} \pi_{i,j} |f(\boldsymbol{z}^{(i)}) - f(\boldsymbol{z}^{(j)})| \right) \\
&= N_1 \min_{\omega \in \Delta(N_1)} \left(\sum_{j=1}^{N_1} -\omega_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) + \lambda \mathcal{W}(f(\mathcal{B}), f(\mathcal{B}')) \right)
\end{aligned}$$

which (up to a multiplicative constant N_1) is a solution of Definition 4.2. □

B Compute the $\widehat{\Phi}_s$ coefficients

Solving the linear program of Definition 4.2 requires computing the coefficients $\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})$ for $j = 1, 2, \ldots, N_1$. To compute them, first note that

$$\begin{aligned}
\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) &= \mathop{\mathbb{E}}_{\substack{\pi \sim \Omega \\ \boldsymbol{x} \sim \mathcal{C}(S_0', \mathbf{1}/M)}} \Big[f(\, \boldsymbol{r}_{\pi_s \cup \{s\}}(\boldsymbol{z}^{(j)}, \boldsymbol{x})\,) - f(\, \boldsymbol{r}_{\pi_s}(\boldsymbol{z}^{(j)}, \boldsymbol{x})\,) \Big] \\
&= \frac{1}{M} \sum_{\boldsymbol{x}^{(i)} \in S_0'} \mathop{\mathbb{E}}_{\pi \sim \Omega} \Big[f(\, \boldsymbol{r}_{\pi_s \cup \{s\}}(\boldsymbol{z}^{(j)}, \boldsymbol{x}^{(i)})\,) - f(\, \boldsymbol{r}_{\pi_s}(\boldsymbol{z}^{(j)}, \boldsymbol{x}^{(i)})\,) \Big] \\
&= \frac{1}{M} \sum_{\boldsymbol{x}^{(i)} \in S_0'} \widehat{\Phi}_s(f, \{\boldsymbol{x}^{(i)}\}, \{\boldsymbol{z}^{(j)}\}).
\end{aligned} \tag{33}$$

The coefficients $\widehat{\Phi}_s(f, \{\boldsymbol{x}^{(i)}\}, \{\boldsymbol{z}^{(j)}\})$ are computed deeply in the `SHAP` code and are accessible with some Monkey-Patching. For instance in our experiments, these coefficients were extracted by modifying the `ExactExplainer`. The code is provided as a fork the `SHAP` repository [4].

C Algorithms

Algorithm 1 presents the whole methodology that must be applied by the company to obtained the misleading subsets S_0', S_1' of size M.

Algorithm 1 Pseudo-code to manipulate GSV and fool audit

```
procedure MANIPULATE_GSV($D, f, M, \lambda_{\text{min}}, \lambda_{\text{max}}, \tau, \alpha$)
    // Setup
    $D_0 = \{\boldsymbol{x}^{(i)} : x_s^{(i)} = 0\}$ and $D_1 = \{\boldsymbol{x}^{(i)} : x_s^{(i)} = 1\}$
    $S_0' \sim \mathcal{F}^M$                                        ▷ $S_0'$ is sampled without cheating
    Compute $\widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\}) \quad \forall \boldsymbol{z}^{(j)} \in D_1$     ▷ cf. Section B
    $\mathcal{B}^\star = \mathcal{C}(D_1, \mathbf{1}/N_1)$
    $\Phi_s^\star = \frac{1}{N_1} \sum_{j=1}^{N_1} \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})$
    for $\lambda = \lambda_{\text{min}}, \ldots, \lambda_{\text{max}}$ do
        // Background Manipulation
        $\mathcal{B}' =$ Solve Linear Program of Definition 4.2 with given $\lambda$
        // Audit Detection
        Detection_rate = 0
        for rep $= 1, \ldots, 100$ do
            $S_1' \sim \mathcal{B}'^M$
            Detection_rate += DETECT_FRAUD($f(D_0), f(D_1), f(S_0'), f(S_1'), \alpha, M$)
        end for
        // Is this a better solution?
        if $|\sum_{j=1}^{N_1} \omega_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})| < |\Phi_s^\star|$ and Detection_rate $< 100\tau$ then
            $\mathcal{B}^\star = \mathcal{B}'$
            $\Phi_s^\star = \sum_{j=1}^{N_1} \omega_j \widehat{\Phi}_s(f, S_0', \{\boldsymbol{z}^{(j)}\})$
        end if
    end for
    $S_1' = \mathcal{B}^{\star M}$
    return $S_0', S_1'$
end procedure
```

Algorithm 2 presents the detection method that is employed by the audit to assess whether or not the subsamples S_0', S_1' provided by the audit were sampled uniformly at random.

D Methodological Details

D.1 Toy Example

The toy dataset was constructed to closely match the results of the following empirical study comparing skeletal mass distributions between men and women [17]. First of, the sex feature was sampled from a Bernoulli

$$S \sim \text{Bernoulli}(0.5). \tag{34}$$

Then according to the Table 1 of [17], the average height of women participants was 163 cm while it was 177cm for men. Both height distributions had the same standard deviation of 7cm. Hence we sampled height via

$$\begin{aligned} H|S = \texttt{man} &\sim \mathcal{N}(177, 49) \\ H|S = \texttt{woman} &\sim \mathcal{N}(163, 49) \end{aligned} \tag{35}$$

[4] `https://github.com/gablabc/shap/tree/biased_sampling`

Algorithm 2 Detection with significance α

```
procedure DETECT_FRAUD($f(D_0), f(D_1), f(S'_0), f(S'_1), \alpha, M$)
    for $i = 0, 1$ do
        // Kolmogorov-Smironov
        for rep $= 1, \dots, 10$ do
            $f(S_i) \sim \mathcal{C}(f(D_i), \mathbf{1}/N_i)^M$
            $p = \text{KS}(f(S_i), f(S'_i))$
            if $p < \alpha/40$ then
                return 1
            end if
        end for
        // Wald
        $p = \text{Wald}(\, f(S'_i), \mathcal{C}(f(D_i), \mathbf{1}/N_i)\,)$
        if $p < \alpha/4$ then
            return 1
        end if
    end for
    return 0;
end procedure
```

It was noted in [17] that there was approximately a linear relationship between height and skeletal muscle mass for both sexes. Therefore, we computed the muscle mass M as

$$\begin{aligned} &M|\{H\!=\!h, S\!=\!\text{man}\} = 0.186h + 5\epsilon \\ &M|\{H\!=\!h, S\!=\!\text{woman}\} = 0.128h + 4\epsilon \\ &\text{with } \ \epsilon \sim \mathcal{N}(0,1) \end{aligned} \tag{36}$$

The values of coefficients 0.186, 0.128 and noise levels 5 and 4 were chosen so the distributions of $M|S$ would approximately match that of Table 1 in [17]. Finally the target was chosen following

$$\begin{aligned} &Y|\{H\!=\!h, M\!=\!m\} \sim \text{Bernoulli}(\,P(H,M)\,) \\ &\text{with } \ P(H,M) = \big[1 + \exp\{100 \times \mathbb{1}(H < 160) - 0.3(M - 28)\}\big]^{-1}. \end{aligned} \tag{37}$$

Simply put, the chances of being hired in the past (Y) were impossible for individuals with a smaller height than 160cm. Moreover, individuals with a higher mass skeletal mass were given more chances to be admitted. Yet, individuals with less muscle mass could still be given the job if they displayed sufficient determination. In the end we generated 6000 samples leading to the following disparity in Y

$$\mathbb{P}(Y = 1|S\!=\!\text{man}) = 0.733 \qquad \mathbb{P}(Y = 1|S\!=\!\text{woman}) = 0.110. \tag{38}$$

D.2 Real Data

The datasets were first divided into train/test/deployment subsets with ratio $\frac{2}{3} : \frac{1}{6} : \frac{1}{6}$. The models were trained on the training set and evaluated on the test set. The deployment set was meant to be use to obtain local post-hoc explanations on new instances seen during deployment (with unknown labels). However, these results ended up not being used for the experiments presented in this paper. Hence, only the training and test were relevant. All categorical features were on-hot-encoded which resulted in a total of 11 and 40 features used by the models fitted on COMPAS and Adults respectively.

A simple random search was conducted to fine-tune the hyper-parameters with cross-validation on the training set. The resulting test set performance for all models and datasets, aggregated over 5 random data splits, are reported in Table 3.

Table 2: Models Test Accuracy (mean ± stddev).

	mlp	rf	xgb
COMPAS	0.73 ± 0.01	0.726 ± 0.005	0.73 ± 0.01
Adult	0.908 ± 0.004	0.914 ± 0.005	0.922 ± 0.004

Table 3: Models Demographic Parity (mean ± stddev).

	mlp	rf	xgb
COMPAS	-0.12 ± 0.01	-0.114 ± 0.005	-0.12 ± 0.02
Adult	-0.20 ± 0.01	-0.194 ± 0.005	-0.196 ± 0.005