# On Detecting Biased Predictions with Post-hoc Explanation Methods

Matteo Ruggeri*
Purdue University

Alice Dethise†
Nokia Bell Labs

Marco Canini
KAUST

## ABSTRACT

We develop a methodology for the analysis of machine learning (ML) models to detect and understand biased decisions and apply it to two specific scenarios. In particular, we show how analyzing model predictions across the dataset, comparing models trained on different subsets of the original data, and applying model-agnostic post-hoc explanation tools can help identify bias in a model in general as well as in specific instances. Further, we consider several definitions of bias and fairness, and show how each provides a different interpretation of the model decisions.

Our results show that the analysis of models through the lens of statistical analysis and post-hoc explanations helps to detect and understand bias. We also observe that post-hoc explanations often fail to detect individual biased instances, and caution against using this category of tools to guarantee model fairness. Finally, we provide insights on how this analysis can help understand the origin and shape of bias.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

## KEYWORDS

Explainable Machine Learning, Post-hoc Explanations, Feature Analysis

**ACM Reference Format:**
Matteo Ruggeri, Alice Dethise, and Marco Canini. 2023. On Detecting Biased Predictions with Post-hoc Explanation Methods. In *Explainable and Safety Bounded, Fidelitous, Machine Learning for Networking (SAFE '23), December 8, 2023, Paris, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3630050.3630179

## 1 INTRODUCTION

Machine learning models are ubiquitous in our everyday lives, including emerging applications in networking [5, 8, 15, 19, 31] and network security [3, 14, 22, 25, 32, 34]. However, concerns arise when those models are used for decision-making in critical or sensitive applications. One of the main concerns for instance is whether the models may discriminate unfairly, either by flaws introduced during training or, more straightforwardly, by perpetuating biases that already exist in current datasets. To address the issue of fairness in ML, researchers have developed various ways to identify bias through statistical definitions [6, 33, 35], to detect bias in trained models [11], to train bias-free models through dataset curation [6] and adversarial training [33, 35], and to interpret the model results [17, 18, 23, 24, 26]. However, due to the challenges of learning unbiased models from biased real-world data and the many different forms bias can take, the challenges of how the different biases and model interpretations are related and navigating the trade-off between fairness versus accuracy are still open problems.

We empirically investigate how bias and accuracy change with different dataset setups and how these biases relate to specific features and decisions of supervised binary classifiers. For this purpose, we combine the analysis of ML decisions through existing statistical definitions of bias and the use of recent model-agnostic explanation tools to understand the results of two well-known datasets (UCI adult [4] and German credit [13]) commonly used for bias investigations. In particular, we propose a methodology to analyze the statistical correlation between features and biases with a concrete application to these datasets and explore how post-hoc explainers [18, 23, 24] help detect and understand learned biases.

In this paper, we report several interesting findings. We show that not all fairness definitions lead to similar bias detection and the protected feature is not necessarily the feature that influences the model biases (§ 4.1). We reveal that post-hoc explanation tools do not always agree on their explanation of the output of the model (§ 4.2). We show that there exist many different ways in which biased instances can be detected with high probability through purely statistical analysis and post-hoc explanation tools (§ 4.3-§ 4.4). Finally, we give evidence that cautions against using post-hoc explanation tools to detect bias in individual predictions (§ 4.4).

We believe that this paper will contribute to improving known techniques for detecting and, ultimately, correcting model biases. While our approach is rather general, we believe that the questions we address are particularly relevant for this workshop on applications of ML in networking given the potentially large user base that these applications may affect.

## 2 PRELIMINARIES

We start by introducing our setup and several definitions of model bias and fairness adapted from prior works.

As the predictor model, we consider a binary classifier mapping the input features $X$ to a boolean decision $Y$, such that the model output $\hat{Y} = f(X)$ for any particular instance is $\hat{y} \in \{+, -\}$. We assume the input $X$ contains one *protected feature* $Z$, which is a feature that should not be used as part of the decision process to ensure fairness. The classifier is not aware, during training, of

*Work done in part while author was visiting at KAUST.
†Work done while author was with KAUST.


*SAFE '23, December 8, 2023, Paris, France*

ACM ISBN 979-8-4007-0449-9/23/12…$15.00
https://doi.org/10.1145/3630050.3630179

which feature is the protected one, and thus cannot be artificially constrained to avoid it.

The protected feature $Z$ has only two values $z \in \{g, \bar{g}\}$. As defined in [6], a *deprived group* is a group of individuals with a specific value in the protected feature $z = g$, which the model discriminates against. On the other hand, individuals belonging to a *favored group* ($z = \bar{g}$) are given an unfair advantage by the model.

Mehrabi et al. [20] present several definitions of fairness that can detect biased models and individual biased predictions. We pick three of the most common definitions used to detect biases inside the entire model. The bias is measured in the model by observing the relation between the value of $Z$ and the probability of selecting a positive output $\hat{y}^+$. Those definitions quantify bias as a single metric, such that a positive value indicates a bias against individuals in the deprived group.

**Definition 2.1.** Demographic parity. A predictor $\hat{Y}$ satisfies demographic parity if $P(\hat{Y}|z = g) = P(\hat{Y}|z = \bar{g})$.

A model is considered unbiased when the probability of a prediction being positive or negative is the same for two instances that are identical except for the protected feature $Z$. Analogously for a dataset, the labels have to be the same. This definition is widely used to detect discrimination both in models and in datasets [6, 33, 35]. Specifically, we use the formulation of demographic parity from [6, 33], defined as:

$$disc_{demo}\,(f) = P(\hat{y}^+|z = \bar{g}) - P(\hat{y}^+|z = g) \tag{1}$$

to measure discrimination in a model $f$, and the same equation with $y$ can be used to measure the discrimination in a dataset $D$.

**Definition 2.2.** Equalized odds. A predictor $\hat{Y}$ satisfies equalized odds with respect to protected feature $Z$ and outcome $Y$, if $\hat{Y}$ and $Z$ are independent conditional on $Y$, i.e., $P(\hat{Y}^+|z = g, Y) = P(\hat{Y}^+|z = \bar{g}, Y)$.

Specifically, given an outcome $Y$ the prediction of the model needs to be independent of the protected feature $Z$. This definition is used in [6, 35] to detect a biased model by comparing the false positive rate ($FPR$) and false negative rate ($FNR$) of the deprived and favored group. We reduce the metric to a single value quantifying the amount of bias in the model as:

$$disc_{odds}\,(f) = \frac{FNR_g - FNR_{\bar{g}}}{FNR_g + FNR_{\bar{g}}} + \frac{FPR_{\bar{g}} - FPR_g}{FPR_{\bar{g}} + FPR_g} \tag{2}$$

where $FPR_g$ and $FNR_g$ are the false positive and false negative rates of the deprived group, $FPR_{\bar{g}}$, and $FNR_{\bar{g}}$ are the false positive and false negative rates of the favored group.

**Definition 2.3.** Equal opportunity. A binary predictor $\hat{Y}$ satisfies equal opportunity with respect to $Z$ and $Y$ if $P(\hat{Y}^+|z = g, Y^+) = P(\hat{Y}^+|z = \bar{g}, Y^+)$.

The prediction and the protected features are conditionally independent on a *positive* label. This definition was used by Zhang et al. [35] to remove model bias through adversarial learning. In this approach, we quantify the discrimination of the model as the true positive rate difference between the favored and deprived groups.

$$disc_{opp}\,(f) = TPR_{\bar{g}} - TPR_g \tag{3}$$

In addition to the three definition of bias metrics above, we also pick two definitions of fairness to detect whether a *specific* prediction is biased: *fairness through awareness* [11] and *counterfactual fairness* [16].

**Definition 2.4.** Fairness though awareness. A predictor is fair if it gives similar predictions to similar instances.

This definition is used by Dwork et al. [11] to train an unbiased model by detecting whether a specific prediction is biased. One way is to use statistical distance to group similar instances and detect variations in the model output. In our analysis, we instead define similar instances as instances where the input features have similar contributions to the decision, as described in § 3.3.

**Definition 2.5.** Counterfactual fairness. Predictor $\hat{Y}$ is counterfactually fair if under any context $W = w$ and $Z = z$, $P(\hat{Y}_{Z\leftarrow g}(U) = y|W = w, Z = z) = P(\hat{Y}_{Z\leftarrow \bar{g}}(U) = y|W = w, Z = z)$, for all $y$.

Where $X = Z \cup W$ and $U$ is the set of relevant latent features which are not observed. Therefore the prediction of the model has to be independent of the protected features.

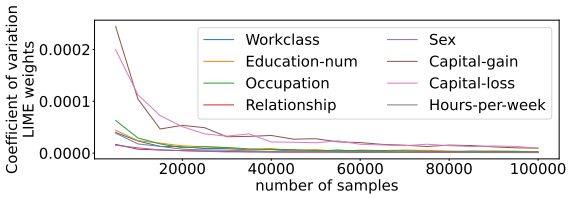
## 3 METHODOLOGY

We devise an empirical methodology to identify bias in binary classifiers. By varying the features and distribution of the inputs throughout various experimental scenarios, we observe how the bias changes in shape and intensity, and analyze the results to correlate recognizable symptoms of biased classifiers. The symptoms are gathered through a statistical analysis of the bias and by applying post-hoc explanation tools to the classifiers.

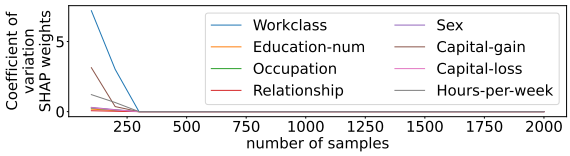
We release our codes and additional results at this repository: https://github.com/MatRug/detecting_individual_bias.

**Model explanations.** We build upon several *post-hoc explanation* tools to determine how the model decided on its output, i.e., attributing a score or weight to individual input features based on their effects on the output. The tools we use are LIME [23], SHAP [18], and Anchors [24]. The first two return a specific weight for each feature, indicating how much the feature influences the result toward an outcome or the others. Anchors instead returns the features that have the biggest influence on the model's prediction. The authors reported that one limitation of this tool is for predictions near the boundary decision function of the model or rare classes of predictions, where the number of *anchors* obtained is very high.

We note that LIME and SHAP sample the input space around the prediction to build a linear model. As the samples are taken randomly around the prediction, the number of samples must be selected carefully to minimize variance in the explanation weights. We extract 10 explanations for every instance in 0.15% of the testing set and compute the coefficient of variation for each feature of the weights computed by the explainer, iterating the number of samples from 5,000 (default value) to 100,000, with steps of 5,000. We use UCI adult dataset for this analysis, with quartile discretization enabled. For LIME, Fig. 1a shows that the coefficient of variation is almost constant after 50,000 samples. Instead for SHAP (Fig. 1b) the coefficient of variation reach a value of $10^{-17}$ for all features (i.e., no variation) after 300 samples. Thus, we select 50,000 and 300 sampled instances per prediction for LIME and SHAP, respectively.




**(a) LIME**




**(b) SHAP**

**Figure 1: Analysis of the coefficient of variation of the weights by varying the number of samples on the UCI adult dataset.**

**Datasets and models.** The UCI adult is a dataset for predicting if an individual has an annual income higher or lower than $50K, based on 14 features. This dataset is known to have biases against both gender and race. Among the protected features we exclude race and focus on gender. In addition to that, we also remove redundant features and differentiate between categorical and numerical features. The final features are: *Workclass*, *Education-num*, *Occupation*, *Relationship*, *Sex*, *Capital-gain*, *Capital-loss*, and *Hours-per-week*. The last 3 are numerical features and the rest are categorical ones, and hence are discretized by LIME and SHAP. The training set has 32,561 instances (21,790 males and 10,771 females) with $disc_{demo}\ (D) = 0.196$. The testing set has 16,281 instances; considering that some features were removed, some instances presented duplicates. With the removal of these duplicate instances, the testing set has 8,257 instances left (5,255 males and 3,002 females) with $disc_{demo}\ (D) = 0.159$ instead of the original testing set where $disc_{demo}\ (D) = 0.192$. In both datasets, the demographic parity is important because the model could learn the same bias when trained on them.

The German credit is a dataset for predicting if a person has good credit standing or not. It has been observed to be biased against gender [7]. We use the 9 features reported in [30]. The training set has 800 instances (552 males and 248 females) with $disc_{demo}\ (D) = 0.064$; the testing set has 200 instances (138 males and 62 females) and $disc_{demo}\ (D) = 0.119$.

We train several neural networks (NNs) that consist of one or more hidden layers with ReLU activation function and the last layer has two units and softmax activation function. Based on prior works [1, 6, 9, 12, 29, 33, 35], we use six different NN architectures for UCI adult, and one (128-64-32 units) for German credit. We use Adam optimizer and binary cross-entropy loss function.

## 3.1 Scenarios

We devise multiple scenarios to analyze and understand how distinct features influence the bias of the classifier and how bias correlates with test accuracy. A scenario refers to a transformation of the original dataset in which a subset of features are excluded or transformed. The base scenario does not exclude any feature. As considering all combinations of features is time prohibitive, we focus on 9 scenarios of interest based on insights on the feature semantics. This derives from knowledge of the protected features and the features correlated to them, which can be obtained through a Bayesian pre-analysis of the dataset.

| Scenario | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Relationship | v | b | x | v | b | x | t |
| Sex | v | v | v | x | x | x | v |

**Table 1: UCI adult features for each case of datasets used to train the models. All cases have Workclass, Education-num, Occupation, Capital-gain, Capital-loss, and Hours-per-week. In scenarios 2 and 5, *Relationship* feature is reduced to a binary value (e.g. *wife* or *other*) (b), and in 7 to *wife*, *husband*, or *other* (t). For all other scenarios a (v) means that the feature is selected and (x) not. In addition to that case there case-8 with equal number of *male* and *female*, and case-9 same as case-8 with demographic parity.**

For UCI adult, Table 1 reports the relevant variations with *Relationship* and *Sex* features across all scenarios. Case-1 is the base scenario with the original dataset. In case-2 and case-5, we use a binarized value of *Relationship* (either *wife* or *other*), to distinguish between bias introduced by the *Relationship* feature (value *wife*) and the *Sex* feature (value *female*). Case-3 excludes the *Relationship* feature altogether, as it is related to *Sex* because the value *wife* is always used for instances where *Sex* is *female*, and *husband* with *male*, hence we seek to observe the importance of this feature in the bias of the model.

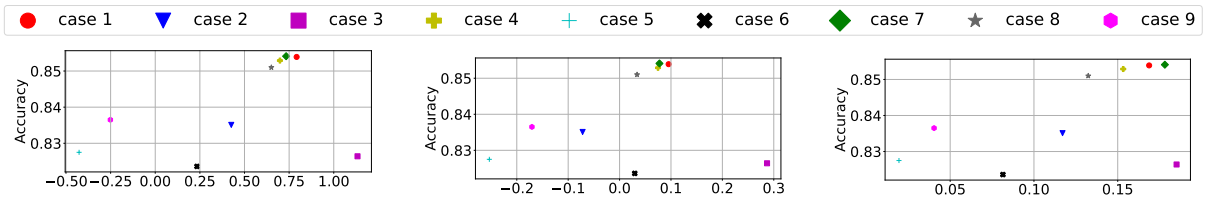
Case-4 excludes the feature *Sex*; therefore, the model can infer the gender of the individual only through the *Relationship* feature. In case-6, both *Relationship* and *Sex* are removed, and the model has no direct access to the gender of the individual represented by each instance. Case-7 has a similar purpose to case-2 and 5, but instead of using a binary value, it uses three values, *husband*, *wife*, and *other*.

Case-8 randomly removes instances from the dataset to guarantee it contains the same number of males and females, to see if the model bias is affected by having more training samples to learn from. Lastly, case-9 is the same as case-8 with demographic parity fairness, which guarantees that if the classifier turns out to be biased, it is due to the training process.

For German credit, we use only two training scenarios, one with all 9 features and one without *Sex*, for similarity with UCI adult: we call them case-1 and case-4. Due to the small number of instances, it is impossible to create cases similar to 8 and 9.

## 3.2 Feature analysis

To understand the reasoning behind the model predictions, i.e., which features and values are used to decide on the output, we analyze individual predictions through model-agnostic post-hoc explanation tools. To account for variations in explanations, we use three different tools: LIME, Anchors, and SHAP. For LIME and SHAP, the output is a weight of the importance of the feature for a particular instance. We use both real weights provided by the tools and weights normalized to the sum of the absolute value of each weight of the instance as in [10]. To compare LIME and SHAP with




(a) Equalized odds (b) Equal opportunity (c) Demographic parity

**Figure 2: Discrimination versus accuracy for UCI adult classifiers. The discrimination is computed according to Equalized odds in Fig. 2a, Equal opportunity in Fig. 2b, and Demographic parity in Fig. 2c.**

Anchors, we order the features by their weight for each instance, and we check the index of the features selected as *anchors*. We also show the difference between the weight of features selected as *anchors* and those that are not.

Instead of looking at individual instances to find correlations between individual biased predictions and the information of model agnostic tools, we cluster the instances whose explanations are similar and extracted the behavior of each cluster. We group instances with three different clustering algorithms (k-means [28], optics-clustering [2], and spectral-clustering [21]) in the case of LIME and SHAP. We select these algorithms because of their different clustering approaches and the different geometries of the clusters. For Anchors, we group instances that have identical features selected as anchors in the same cluster.

### 3.3 Individual biased prediction detection

For the fairness verification of the classifiers through the lens of *Fairness through awareness* as defined in § 2, similar instances were clustered according to their weight as measured by LIME with quartile discretization. This approach allows grouping the instances considered similar by LIME, considering that it assigns similar weights importance for all features of the group instances. Therefore the model has to predict the same outcome. Based on these results, we choose 7 possible ways to define similar instances. We firstly *exclude* the protected feature (*Sex*), and as the first definition of similarity, we consider identical any two instances that have every other feature at the same value. The other similarity groups are likewise selected by grouping instances that have similar weights for all features: *Hours-per-week* (weights divided in three groups), *Capital-gain* (2 groups), *Capital-loss* (2 groups), *Relationship* (2 groups), and *Education-num* (4 groups). Using this approach to define similarity groups, it is impossible to find similar instances for German credit, as the dataset is too small and results in a very small number of instances for each group.

Through the lens of *Counterfactual fairness* bias, we modify instances by changing the protected feature from the deprived group to the favored group or vice versa and observe that if the prediction change, then it is considered biased. For both datasets, this means changing from *female* to *male* or the opposite; in the case of UCI adult, we also change *wife* to *husband* and the opposite if the *Relationship* feature holds one of those (other values of that feature are not used for this analysis, as they do not correlate to the protected feature we are investigating). The predictions whose output changes after the features presented above are modified are considered biased.

## 4 RESULTS

First, we trained a neural network model on the various scenarios described in § 3.1 and validated their accuracy against existing works [1, 6, 9, 12, 29, 33, 35], then analyze how the accuracy and fairness changes depending on the dataset scenario in § 4.1. We also extracted feature importance from the classifiers through post-hoc explainers and report the results in § 4.2, comparing how different tools provide different results. Finally, we analyze in § 4.3 and § 4.4 how the information from the explainers is correlated to individual biased predictions, comparing the results between bias and unbiased predictions.

To validate our models, we compare the accuracy obtained with other works and state-of-the-art models, for the case-1 scenario (i.e., the original dataset). For UCI adult dataset, the architecture that reached results similar to state-of-the-art models is a feed-forward neural network with 256-128-64-32 nodes for each layer and 2 nodes in the last layer with Softmax activation function. Instead, for German credit dataset, the best neural network has 128-64-32 nodes per layer and the last one has 2 nodes and softmax activation function.

### 4.1 Accuracy vs. discrimination

Achieving an unbiased model often requires sacrificing some accuracy. This section reports the measured accuracy and discrimination obtained by each case, following the definitions of Equalized odds, Equal opportunity, and Demographic parity from § 2, where a positive value means that the model is biased on the protected feature *Sex* against the deprived group *female*, and negative values indicate a bias against the favored group *male*. Fig. 2 reports the discrimination and accuracy of the models for each case of UCI adult and shows that while the trend is similar between all definitions, some differences remain: in Demographic parity, case-1 has lower discrimination than case-7, and in Equal opportunity, case-2 is less biased than case-6. Moreover, demographic parity for case-1 is similar to the one of the dataset (§ 3). Therefore, we can conclude that the model is learning to be biased from the dataset.

We also observe that some of the scenarios invert the discriminatory aspect: a bias is created against the favored group in case-2, case-4, and case-9 for Equalized odds, and in case-4 and case-9 for Equal opportunity. Case-1, 4, 7, and 8 have similar values of accuracy and discrimination, in particular, case-4 does not have the

| | Case-1 | Case-4 |
|---|---|---|
| Accuracy | 0.745 | 0.745 |
| Equalized odds discrimination | 0.439 | 0.222 |
| Equal opportunity discrimination | 0.062 | 0.013 |
| Demographic parity discrimination | 0.127 | 0.074 |

**Table 2: Discrimination and accuracy results for models trained on the German credit dataset.**

protected feature in the dataset, and it still is biased, even though in the opposite sense. Therefore, we deduced that the model can extrapolate the protected feature from other features, such as *Relationship*. These plots also show that having an equal number of instances in the deprived and favored group (case-8) is not enough to make the model unbiased. Training the model on a dataset with demographic parity (case-9) gave the best result for accuracy and discrimination. Cardoso et al. [6] show a correlation between *Sex* and *Relationship*. For this reason, removing both features (case-6) leads to a minimal value of discrimination.

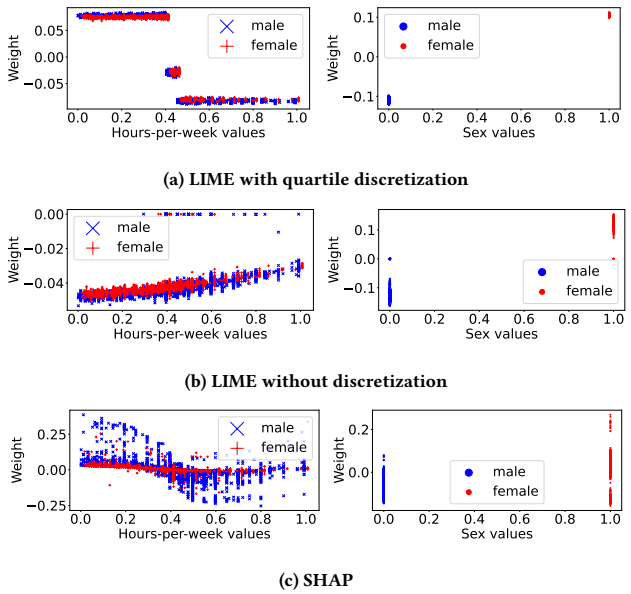
Table 2 reports the results for the German credit dataset. As for the UCI adult dataset, removing the protected feature does not change the accuracy of the model. However, unlike UCI adult, the discrimination values are halved for case-4 by Equalized odds and Demographic parity, and a fourth by Equal opportunity because there is no other feature correlate to *Sex* as *Relationship* in the case of UCI adult. Moreover, the Demographic parity value is more similar to the one of the testing set for case-1 and the training set for case-4. Therefore, despite the training set not being fully biased, it instills a biased behavior in the model that shows up when analyzing more biased data.

## 4.2 Importance of each feature

To better understand the prediction of the model, we analyzed the importance of each feature by LIME, SHAP, and Anchors. We report only the results obtained on the UCI adult case-1 dataset because it is the complete dataset and the results of the other scenarios have similar results when analyzing bias. We do not report the analysis on German credit because they are similar to UCI adult.

Fig. 3 show the importance of *Hours-per-week* and *Sex* features returned by the explainers, where a positive weight means the feature favors an income <= $50K$ and a negative weight > $50K$; in red are reported the values obtained for females, and in blue for males. *Hours-per-week* has a clear trend in the weights obtained by LIME with quartile discretization. In the other cases, the results are more spread, and the weights of LIME without discretization show an opposite trend for *Hours-per-week*. For the *Sex* feature, LIME always gives a negative value to the weight of *male* and positive to *female*. Instead, in SHAP, females received negative and positive weights, but males only received negative weights, which indicates that the model is biased because it favored males, not because it penalized females.

To compare Anchors results with LIME and SHAP, we compare the weights obtained by each feature in general and when it is chosen as an *anchor*. Fig. 4a shows the average normalized weight obtained by each feature, and the same results obtained when the features are selected as anchors, with LIME and Anchors with




**(a) LIME with quartile discretization**

**(b) LIME without discretization**

**(c) SHAP**

**Figure 3: Relation between feature values and weights obtained by LIME with quartile discretization on numerical features Fig. 3a, LIME without discretization on numerical features Fig. 3b, and SHAP Fig. 3c.**

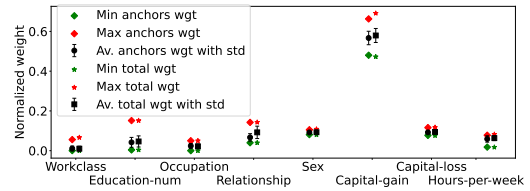
quartile discretization; LIME with decile discretization achieved similar results. We expected that the features selected as anchors should have a higher average weight than all together, as it is for the case of SHAP in Fig. 4b. Instead, for LIME, the results are counterintuitive, and the average weight of the features selected as anchors is lower than all of them together. These unexpected results can be caused by the fact that although the average weight of the anchors is lower than the others, the anchor feature still has greater importance. From other analysis, we noticed that *Relationship* has more anchors than *Sex*, and that can be explained because the average weight of the *Relationship* anchor is different from the general average weight, unlike *Sex*, where they are similar.

## 4.3 Fairness through awareness detection

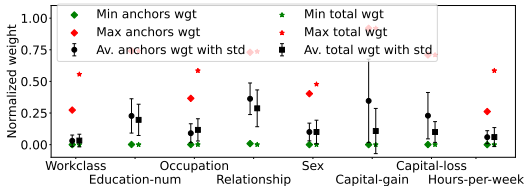
We notice that there is no correlation between the results obtained by the analyzing tools and the predictions detected as biased. The results show no correlation between the weights of the features and the biased predictions. This result may be caused by the low number of biased predictions in the groups, which range from 1 to 44 biased predictions, because, for each biased prediction, many more with similar weights are unbiased. The same is for Anchors, where there is no correlation in both groupings because each group has more instances with fair predictions than with biased ones.

## 4.4 Counterfactual fairness detection

Compared to *Fairness through awareness*, this definition labels more predictions as biased: 386 predictions for UCI adult case-1 and 28 for the German credit dataset. In this case, the weights of SHAP can detect predictions that are considered biased by the *counterfactual fairness* definition.




**(a) LIME with quartile discretization**




**(b) SHAP**

**Figure 4: Weights obtained for each feature for all the instances and the features selected as *anchors*, for LIME with quartile discretization Fig. 4a and SHAP Fig. 4b.**

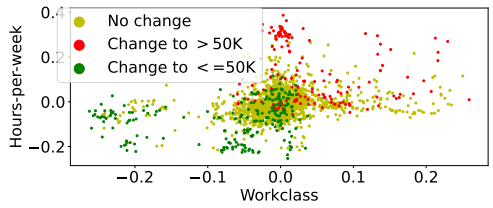



**Figure 5: Scatter plot of *Workclass* and *Hours-per-week* SHAP weights.**

Fig. 5 shows the weights obtained by SHAP and if the prediction changes due to a change in the protected feature value. It is possible to observe that considering only *Hours-per-week* is sufficient to detect some biased predictions because they have a large absolute weight value, without performing clustering. The same results are obtained for all cases (except case-6, where there are no protected features or features related to them, making it impossible to define biased instances through *counterfactual fairness*); in case-4, the biased predictions are detected only by the change from *husband* to *wife* or the opposite. For UCI adult dataset case-9, it is still possible to correlate the detected biased prediction with the weights of SHAP, and this definition detects 1,486 biased predictions. Even though the case-9 dataset has demographic parity, unlike the other cases, it still holds more biased predictions. The discrepancy in the results of the definition is because not all the definitions lead to the same notion of fairness, especially if they aim at a model or a single prediction. The feature *Sex* on the German credit dataset achieved the same results as *Hours-per-week* for the UCI adults dataset. For Anchors and LIME is not possible for either grouping or datasets to detect biased predictions obtained from this definition.

Therefore, we can conclude that it is impossible to distinguish individual biased predictions detected by *fairness through awareness* definition with all 3 model agnostic tools. Instead, SHAP can detect individual biased predictions obtained through *counterfactual fairness* definition, not only on models trained on the whole dataset but also in a model trained on the dataset without the protected feature. From this result and the one shown above, we can deduce that only SHAP is consistent with other theories and definitions.

## 5 RELATED WORK

Our work on detecting biased predictions with post hoc explanation methods mainly relates to two topics: post-hoc explanation of predictions and bias detection in individual predictions and in the entire model. Slack et al. [27] presented a work that belongs to the same category, though their methodology differs. They show that it is possible to fool post-hoc explanation tools like LIME and SHAP with scaffolding around the input data, which makes the model appear as unbiased even if it is biased. They also show that LIME is more vulnerable than SHAP, similarly to the results that we obtained, in which we show that it is possible to detect biased predictions with SHAP weights but not with LIME weights.

The work of Kusner et al. [16] belongs to bias detection, and they showed that using the *counterfactual fairness* definition makes it possible to detect individual biased predictions in a model trained with the protected features; they argue that protected features also have importance in the final prediction, and their complete removal can be counterproductive. Therefore, it is useful to have a way to detect individual biased predictions. Dwork et al. [11] dealt with individual fairness. They showed that it is possible to train a model that treats similar instances similarly (*Fairness thought awareness*) and that with this approach, it is also possible to satisfy *statistical parity* (referred to as *demographic parity* in our paper). Their work applies to the discrimination prevention area, but they suggest that their method can also be used to detect discrimination in individual biased predictions.

## 6 CONCLUSION

Starting from several common definitions of model bias and fairness, we analyzed with the aid of post-hoc explanation tools and statistical correlation how to detect the presence of bias, quantifying how different definitions have dissimilar results. This finding is not surprising considering the large number of definitions, which are not necessarily compatible, but it shows the need to analyze correlations among these tools and the fairness metrics to understand both of them better. At the same time, we show that post-hoc explanation tools can play an important role in this type of investigation. While our study is focused on two specific datasets, we believe that the ensuing methodology has broader applicability.

## Acknowledgments

For computer time, this research used the resources of the Supercomputing Laboratory at KAUST.

## REFERENCES

[1] Abubakar Abid. 2018. Which Machine Learning Classifier Is the Best on 18 UCI Datasets? https://abidlabs.github.io/uci-datasets/ Accessed 11 Oct 2023.
[2] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure. In *SIGMOD*.
[3] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and Don'ts of Machine Learning in Computer Security. In *USENIX Security*.

[4] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20

[5] Raouf Boutaba, Mohammad A Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M Caicedo. 2018. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications* 9, 1 (2018), 1–99.

[6] Rodrigo L. Cardoso, Virgilio Almeida, Wagner Jr Meira, and Mohammed J. Zaki. 2019. A Framework for Benchmarking Discrimination-Aware Models in Machine Learning. In *AIES*.

[7] Falk Casey, Friedler Sorelle A., Nix Tionney, Rybeck Gabriel, Scheidegger Carlos, Smith Brandon, and Venkatasubramanian Suresh. 2018. Auditing Black-box Models for Indirect Influence. *Knowl. Inf. Syst.* 54 (Jan 2018), 95–122.

[8] Li Chen, Justinas Lingys, Kai Chen, and Feng Liu. 2018. Auto: Scaling deep reinforcement learning for datacenter-scale automatic traffic optimization. In *SIGCOMM*. 191–205.

[9] Vidya Chockalingam, Sejal Shah, and Ronit Shaw. 2017. Income Classification using Adult Census Data.

[10] Arnaud Dethise, Marco Canini, and Srikanth Kandula. 2019. Cracking Open the Black Box: What Observations Can Tell Us About Reinforcement Learning Agents. In *NetAI*.

[11] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through Awareness. In *ITCS*.

[12] M. Hendra Herviawan. 2019. Predicting German Credit Default. https://www.kaggle.com/hendraherviawan/predicting-german-credit-default.

[13] Hans Hofmann. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. https://doi.org/10.24432/C5NC77

[14] Arthur S Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A Ferreira, Arpit Gupta, and Lisandro Z Granville. 2022. AI/ML for network security: The emperor has no clothes. In *CCS*. 1537–1551.

[15] Vadim Kirilin, Aditya Sundarrajan, Sergey Gorinsky, and Ramesh K Sitaraman. 2019. Rl-cache: Learning-based cache admission for content delivery. In *Workshop on Network Meets AI & ML*. 57–63.

[16] Matt Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeurIPS*.

[17] Zachary C Lipton. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018).

[18] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS*.

[19] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *SIGCOMM*. 197–210.

[20] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys* 54 (2019), 1–35.

[21] Andrew Ng, Michael Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an algorithm. In *NeurIPS*.

[22] In Search of netUnicorn: A Data-Collection Platform to Develop Generalizable ML Models for Network Security Problems. 2023. Roman Beltiukov and Wenbo Guo and Arpit Gupta and Walter Willinger. *arXiv* (2023). https://doi.org/arXiv:2306.08853

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you": Explaining the Predictions of Any Classifier. In *SIGKDD*.

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI*.

[25] Jorge Rivero, Bernardete Ribeiro, Ning Chen, and Fátima Silva Leite. 2017. A grassmannian approach to zero-shot learning for network intrusion detection. In *ICONIP*. 565–575.

[26] Andrew D. Selbst and Solon Barocas. 2018. The Intuitive Appeal of Explainable Machines. *Fordham Law Review* 87 (2018), 1085–1139.

[27] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *AIES*.

[28] Hugo Steinhaus. 1956. Sur la division des corps matériels en parties par. *Bulletin de l'académie polonaise des sciences* 4 (1956), 801–804.

[29] Dhruv Dhanesh Thanawala. 2019. *Credit Risk Analysis using Machine Learning and Neural Networks*. Technical Report. Michigan Technological University.

[30] UCI Machine Learning. 2017. German Credit Risk. https://www.kaggle.com/uciml/german-credit Accessed 31 Oct 2023.

[31] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang. 2017. Machine learning for networking: Workflow, advances and opportunities. *IEEE Network* 32, 2 (2017), 92–99.

[32] Wei Wang, Ming Zhu, Jinlin Wang, Xuewen Zeng, and Zhongzhen Yang. 2017. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *ISI*. 43–48.

[33] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *BigData*.

[34] Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. 2021. CADE: Detecting and Explaining Concept Drift Samples for Security Applications. In *USENIX Security*. 2327–2344.

[35] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating Unwanted Biases with Adversarial Learning. In *AIES*.