Automated detection of bias symptoms to support fairness auditing processes: yay or nay?

***Abstract*—Due to the pervasiveness of AI-based systems, fairness auditing processes have become crucial to assess the fairness of involved algorithms or models, particularly in sensible contexts such as hiring and lending, where biased decisions could have significant consequences. Fairness auditing encompasses diverse steps, from identifying sensitive variables to computing well-defined metrics. However, computing such metrics depends on the elicited sensitive variables and requires predictions from a trained baseline model. In this respect, devising such a process requires *i)* deep knowledge of the domain application and *ii)* computational power to run the experiments to get the final predictions. In this paper, we propose a novel approach employing datasets' characteristics (i.e., *bias symptoms*) that are functional for the early identification of possible variables leading to bias in the system. First, we synthesize a bias-symptoms dataset by considering 24 datasets from different application domains used in fairness auditing. Next, we perform an extensive empirical study to assess the capability of these symptoms to support the prediction of possible bias concerning three widely adopted fairness metrics. Our empirical findings reveal that *bias symptoms* can support an early prediction of bias-inducing variables, avoiding multiple training phases of the underpinning ML classifiers. Furthermore, the proposed approach can effectively predict the fairness metrics values within certain thresholds. More importantly, the significant improvement in computational performances compared to traditional approaches paves the way for adopting bias symptoms in the fairness auditing process.**

I. Introduction

Given the increasing prevalence of machine learning (ML) intensive systems within sensitive domains such as health, crime, or education [1], addressing and mitigating bias within these systems has emerged as a critical priority [2], [3]. The concept of *software fairness* is central, denoting an intelligent system free from any prejudice or favouritism toward an individual or group based on their inherent or acquired characteristics [4]–[6] which embrace ethical, social, and legal aspects of complex software systems. The importance of fairness is also highlighted by the newly approved AI Act from the European Union, which highlights fairness as one of the requirements for *critical systems* [7]. For this reason, the software engineering (SE) community has widely investigated this topic [8]–[10] since fairness has become a key quality property of learning-enabled systems [11]. Starting from a definition of bias for a particular domain, a general *fairness auditing process* encompasses different phases that span from the manual identification of *sensitive variables* to the application of well-founded *fairness metrics* [12]. On one hand, performing manual steps in eliciting sensitive variables or selecting the proper metrics requires a deep knowledge of the application domain and the fairness concepts [4]. On the other hand, conducting automated fairness audit processes may be demanding from the computation point of view. They typically require the selection and training of predictive models, as well as assessing bias within the training data by analyzing the resulting output. A plethora of approaches [13]–[15] have been proposed to promote automatic bias assessment. However, all these techniques assess bias starting from the predictions of the abovementioned predictive model, which hence needs to be trained beforehand, increasing the computational complexity of the whole approach [16], [17].

In this paper, we propose a novel approach that identifies potential bias-inducing variables by relying on structural features, namely *bias symptoms*, extracted from widely adopted datasets in the fairness domain, thus avoiding the expensive training phase of the baseline model. The process starts with the identification of symptoms and the rationale behind their selection. Subsequently, we obtain a dataset of symptoms from 24 binary and multi-class datasets well-known in the fairness literature. To the best of our knowledge, this represents the largest number of datasets utilized in a fairness study focusing on tabular data. In addition, we evaluate the ability of the selected symptoms to detect *High* and *Low* values of three commonly used bias metrics, i.e., *Statistical Parity (SP)* [18], *Equal Opportunity (EO)* [19], and *Average Odds (AO)* [19], with respect to introduced thresholds.

To assess our approach, the evaluation process analyzes two dimensions, i.e., *effectiveness* and *resource consumption*. Concerning the former, we trained three state-of-the-art ML classifiers using the synthesized symptoms dataset to understand to what extent the symptoms can help predict *High* and *Low* bias values. Concerning the latter, we compare our approach with two existing fairness auditing tools by comparing them in terms of execution time and CPU consumption. The conducted evaluation demonstrates that traditional ML classifiers trained with the proposed symptoms dataset can effectively predict *High* and *Low* SP values using a given threshold, while predictions for EO and AO are less effective, but not meaningless. In addition, the performed evaluation reveals that using bias symptoms can effectively reduce overall resource consumption, outperforming the two traditional approaches considered.

Our analysis shed light on the practicality of using bias symptoms as an efficient and effective methodology to improve the current fairness auditing process, even though the identified symptoms cannot be generalized as they strongly depend on the elicited datasets. While we anticipate that further analyses are required to strengthen our claims, we set

a first stepping stone to support fairness assessment, avoiding the usage of resource-demanding models.

The main contributions of the paper are as follows:
➢ A novel approach to detect bias-inducing variables and bias metric values by exploiting bias symptoms;
➢ A rigorous empirical evaluation that assesses the accuracy and the efficiency of the bias symptoms using well-founded statistical indexes and metrics;
➢ A replication package to facilitate research in this domain [20].

II. Motivation and background

A. Auditing fairness on tabular data

This section discusses the aspects of the generic fairness auditing process [6], [8], [9], [12] that is performed on tabular data as depicted in Figure 1. In particular, the process is divided into two phases, i.e. *effectiveness assessment* and *fairness assessment*. The two phases encompass both *manual* (identified by the *hand* icon) and *automated* (identified by the *gear* icon) steps.

First, the *sensitive variable extraction* has to be performed on the initial *tabular dataset*, i.e., the variables belonging to the dataset that can lead to bias in the system. In most cases, this phase is carried out manually, thus requiring a deeper knowledge of the application domain [6], [21]. In addition, the initial dataset is split into a training and testing set. The former is used to feed a *predictive model* that produces the variables' labels as outcomes. The latter is employed to carry out the *effectiveness assessment* phase by computing traditional effectiveness metrics [22]. Afterward, the developer must select a set of *fairness metrics* to enable the fairness assessment given identified variables and the predicted outcomes. In this respect, existing literature has defined a set of fairness metrics that can be employed on tabular data, including *Statistical Parity* [18] and *Equalized Odd* [19], that need to be selected according to the specific dataset. In other words, the selection of those metrics strongly depends on the application domain. The last step consists of computing the elicited *fairness metrics* to collect the results. The obtained values are used to claim if the system shows any bias according to the *fairness thresholds* specified for each metric.

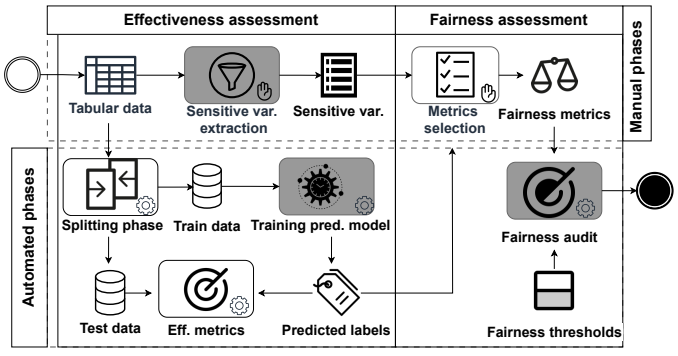


Fig. 1. General fairness audit workflow

Although the process depicted in Figure 1 is well-established in the current literature [8], [9], [12], grey highlighted phases are still challenging. Concerning the manual steps, expert domains are required to elicit both the *sensitive variables* and the proper *fairness metrics* [1]. Concerning the automatic tasks, training the selected *predictive model* can be demanding in terms of computational resources since it depends on the size of the dataset. Moreover, *fairness audit* depends on the employed metrics and thresholds, thus leading to possible erroneous outcomes if they are not properly defined. Automating the whole process is out of the scope of this paper. Instead, our approach will exploit bias symptoms to predict *i)* variables that can be the cause of the bias and *ii)* predict the value of the fairness metrics to assist the fairness assessment sub-process. Consequently, our approach can be seen as an alternative strategy that can enhance the abovementioned manual and automated steps in the traditional fairness assessment workflow.

Remark #1: The fairness audit process involves a sequence of manual and automatic steps, thus requiring a deep knowledge of possible bias-inducing variables in the specific considered application domain. In addition, selecting the proper classifier is a trial-and-error process that may be demanding in terms of time and computational resources.

B. Existing approaches

In the scope of the paper, we aim to find existing approaches that propose automated or semi-automated methodologies for the early identification of sensitive variables that can lead to bias. In addition, we check if existing approaches can cover the fairness metrics prediction. To this end, we conduct a widely adopted strategy to search relevant works in software engineering [23]. First, we executed a query on the Scopus digital library[1] with the following set of keywords: *(i)* bias OR unbias* OR fair* *(ii)* assessment OR auditing OR testing. All such keywords are combined by using the AND operator. We considered papers published over the last five years in top-tier SE venues.[2]

In such a way, we obtained 94 peer-reviewed papers. Afterward, we filtered this initial set by carefully inspecting the title and abstract, thus obtaining 20 works that are relevant to our study. We made a further selection by focusing on approaches that provide bias detection strategies to support the fairness auditing process in software systems. Meanwhile, foundational papers, empirical studies, and approaches focused on bias mitigation are excluded from our study. The process ended with 7 relevant works that have been analyzed in terms of the following features:
➢ **Bias-detection strategy**: It describes which strategy or algorithm has been used to perform the early variable detection.
➢ **Predictive model(s)**: It describes the adopted model used to evaluate the proposed bias-detection strategy. In particular,

[1]https://www.scopus.com/
[2]The complete list of papers and venues are available here [20]

TABLE I
COMPARISON OF THE EXISTING APPROACHES

Approach	Bias-detection strategy	Domain	Predictive model(s)	SD	MP	Datasets
Oneto *et al.* [26]	Multi-task Learning	Social	Multi-class	●	○	2
Openja *et al.* [27]	Counterfactual approach	Mixed	Mixed	●	○	5
Ferrara *et al.* [28]	Ontology-based	Financial	Multi-class	●	○	1
Galhotra *et al.* [29]	Causal fairness	Mixed	Mixed	◐	○	5
Mecati *et al.* [30]	Mutation technique	Financial	Multi-class	◐	○	1
Yik *et al.* [31]	Functional Specified Complexity	Social	Binary	◐	○	2
Constantin *et al.* [32]	Human feedback	Social	Multi-class	○	◐	1
Our approach	Bias Symptoms	Mixed	Multi-class	●	●	24

○= not implemented ◐= partially implemented ●= fully implemented

Multi-class and *Binary* refer to ML model classifiers. When the approach compares different kinds of models, we label the column as *Mixed*.

➢ **Domain**: It represents the application domains defined by the datasets employed by the approach. In particular, we report that *Social* and *Financial* are the most spread ones since they refer to benchmarking datasets used in the fairness auditing process, e.g., Adult [24] or Credit [25].

➢ **Sensitive variable detection (SD)**: This feature represents to what extent the approach supports the prediction of sensitive variables using the proposed strategy. In particular, we consider full support only when the approach works without an initial set of variables.

➢ **Fairness metrics prediction (MP)**: Similarly to the previous feature, we assess if and how the approach covers the fairness metrics prediction.

➢ **Datasets**: The number of datasets used to evaluate the approach. This feature aims to assess the generalizability of the approach in terms of covered domains.

Table I summarizes the relevant methodologies that are relatedo to our approach.

It is worth noting that only three works are able to support the SD feature without considering an initial set of sensitive features. Oneto *et al.* [26] employs a multi-task learning (MLT) approach to predict sensitive features based on non-sensitive ones. Openja *et al.* [27] employs a counterfactual approach to remove bias-inducing features. First, the divergence measurement metrics are computed to assess if there is some potential bias in the distribution. Then, the authors apply a data-swapping strategy to measure the impact of the features on the final predictions. ReFair [28] is a framework to automatically classify sensitive features from the textual requirements using a tailored fairness ontology. Due to the lack of publicly available requirements datasets for fairness auditing, the authors built a synthetic dataset of user stories using the GPT language model.

By relying on the definition of *causal fariness* [33], Galhotra *et al.* [29] suggests a new set of features that can or cannot introduce bias given an initial set of variables extracted from the considered datasets, including a synthetic one used for assessing the complexity of the underpinning algorithm. This approach can be seen as complementary to ours since it is able to add an extended set of feature variables given an initial one. Mecati *et al.* [30] adopts a mutation-based approach to predict possible discrimination using four different balance indexes, i.e., Gini, Shannon, Simpson, and Imbalance Ratio. Starting from an initial set of sensitive variables, the approach generates synthetic datasets according to different level of balance in the data. Yik *et al.* [31] uses a functional specified complexity algorithm to identify potential bias in the dataset without training the ML classifiers. The approach is efficient in terms of computation even for a large number of ten different sensitive variables. Constantin *et al.* [32] propose FairAlign, a toolkit that supports the fairness auditing process using human feedback. In particular, the proposed tool has been used to annotate the variables that are perceived as biased. Afterward, FairAlign computes a set of fairness metrics and compares the prediction with the human judgment collected. In this respect, it is the only approach that can partially cover the MP feature, even though it is time-consuming since the whole procedure is performed manually.

Notably, none of the considered approaches considers the efficiency of the adopted bias-detection strategy apart from Yik *et al.* [31]. Moreover, we remark that all approaches consider at most only five datasets, thus posing issues to the overall generalizability of the proposed techniques. In this respect, our approach (highlighted in grey) considers 24 different datasets, including a dataset related to the IoT domain [34], that have been used to support both SP and MP features.

Remark #2: Detecting bias-inducing features has been explored in the recent literature even though few approaches can cover the bias detection without prior knowledge of the dataset variables. Moreover, we report the lack of coverage in terms of application domains since the existing approaches consider at most five datasets in the fairness auditing process.

III. METHODOLOGY

In this section, we describe the methodology we followed to identify the bias symptoms used for the early identification of possible variables leading to bias in the system. First, we describe the bias metrics considered in this work. Second, we describe the identified symptoms and explain the rationale behind their selection. Third, we define the process of collecting those symptoms and experimenting with them by creating a dataset of bias symptoms starting from 24 datasets from the fairness literature. Finally, we show some statistics about the collected symptoms and their relationships.

A. Selected fairness metrics and relative thresholds

Several metrics available in the literature can be used to assess the amount of bias towards sensitive groups [6], [21], [35]. In this work, we focus on *three fairness metrics* that are widely adopted in the literature [10], [36]–[38]:[3]

➢ **Statistical (Demographic) Parity (SP) [39]** This metric belongs to the *independence* fairness definitions [40] and measures the probability of having the positive outcome predicted being in the privileged (i.e., $S = 1$) or unprivileged (i.e., $S = 0$) group: $P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)$

[3]In the following definitions, $\hat{Y}$ represents the model's predictions, S is a generic sensitive variable, and TPR and FPR are the true and false positive rates.

➢ **Equal Opportunity (EO) [19]** This metric belongs to the *separation* fairness definitions [40] and measures the difference between the true positive rates of the unprivileged and privileged groups: $TPR_{unpriv} - TPR_{priv}$

➢ **Average Odds (AO) [19]** This metric measures the average difference between true positives and false positive rates of the unprivileged and privileged groups: $\frac{1}{2}[TPR_{unpriv} - TPR_{priv}] + [FPR_{unpriv} - FPR_{priv}]$

Following previous works [5], [36], we consider absolute values for all these metrics, meaning that a value of zero means optimal fairness, while a value of one means complete bias. In addition, we consider the following *thresholds* to distinguish *High* and *Low* bias: 0.2 for SP (following the 80% rule [41], [42]), and 0.15 for EO and AO. Since there is still no agreement on which threshold to use for EO and AO [43], [44], this value has been empirically selected to reduce the unbalance between items with high bias and items with low bias in the collected symptoms dataset (see Section III-D).

B. Symptoms identification

The first step in our work was the identification of characteristics for the early identification of variables leading to *High* bias with respect to the thresholds previously discussed. Such characteristics represent *bias symptoms*, which have been selected by analyzing the metrics formulations shown above, performing empirical analyses of the datasets employed in our study (see Section III-C) and reading fairness-related works. In total, we have identified 13 different symptoms, which we describe below:

➢ **Absolute Probability Difference:** This value is a version of SP which considers ground truth labels instead of the model's predictions (i.e., $|P(Y = 1|S = 0) - P(Y = 1|S = 1)|$) [39].

➢ **Unprivileged Positive Probability (UPP) and Privileged Positive Probability (PPP):** These values represent the single probabilities that compose the above symptom [39].

➢ **Gini Index, Simpson Diversity, Shannon Diversity, and Imbalance Ratio (IR):** These values have been used in [30] to measure the unbalance in the values of a sensitive variable. In [30] the authors show how these values are able to reflect variations in SP, EO, and AO. Following their approach, we 0-1 normalize these values such that, for Gini, Simpson, and Shannon, a value of 0 means that a variable is entirely unbalanced, while a value of 1 means that a variable is fully balanced. Concerning IR, a value of 1 means that the variable is entirely unbalanced, while a value of 0 means full balance.

➢ **Unprivileged Group Unbalance and Privileged Group Unbalance:** These values are used to represent the unbalance of a variable with respect to the positive value of the ground truth label. In particular, they are the ratio between the expected and observed sizes of the unprivileged and privileged groups with respect to the ground truth label ($\frac{W_{obs}}{W_{exp}}$). A value of one means that the groups are fully balanced (i.e., the ratio of items having a positive and negative label value is the same), a value > 1 means that the group is oversampled (i.e., the observed size is higher than expected). In contrast, a value < 1 implies that the group is undersampled (i.e., the observed size is lower than expected). These values have been used by previous works to develop methods able to mitigate *unbalanced groups bias* [36], [45].

➢ **Kurtosis and Skewness:** These values have been included to represent the distribution of a variable. They have been used by several works in the Auto ML domain, and it has been shown how they can influence the predictions of an ML model [46]–[48].

➢ **Kendall's τ:** This value represents the correlation between a variable and the ground truth label. We adopted Kendall's τ to measure the correlation because it is non-parametric and more robust than the other non-parametric Spearman's correlation coefficient [49]. It ranges between -1 and 1, where -1 means absolute negative correlation, 1 means absolute positive correlation, and 0 implies no correlation. Intuitively, a variable shall be highly correlated with the ground truth label to lead to *High* bias in the predictions of a model. To confirm this intuition, we computed the Kendall τ between each binary variable and the ground truth label on 24 datasets from the fairness literature (see Section III-C) and then grouped the results by variables having *High* and *Low* values of SP, EO and AO following the thresholds defined in Section III. Figure 2 shows the mean and the 95% confidence interval of the Kendall τ grouped by high and low values of each bias metric: variables with high values of SP and AO are also more positively correlated with the ground truth label. We also computed the *Welch's t-test* (a non-parametric test to assess the null hypothesis that two groups with different numbers of samples have the same mean [50]), which confirmed a statistically significant difference of the means concerning SP and AO (following previous works [9], [36], we consider a statistical test significant if the p-value is < 0.05).

➢ **Mutual Information:** This value is a non-parametric metric that measures the mutual dependency between two random variables (continuous or discrete). It ranges between 0 and 1, where 0 means complete independence, while 1 means complete dependence [51]. Like Kendall's τ, we included this metric to represent how much dependency exists between a variable and the ground truth label. As before, we empirically compared the mean values of Mutual Information between items with high and low values of SP, EO, and AO. Figure 3 reports the results of our evaluation. Differently from Kendall's τ, we observe a statistically significant difference in the mean values for all the considered bias metrics. In particular, the

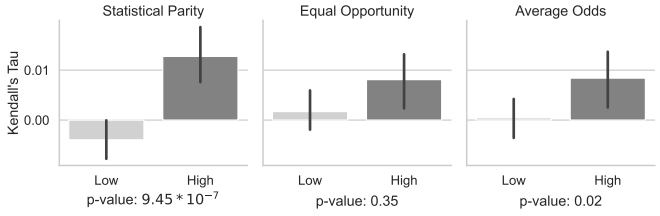


Fig. 2. Mean and 95% confidence interval of Kendall τ between binary variables and ground truth labels grouped by High and Low SP, EO and AO values. For each metric, we report the Welch's t-test p-value.

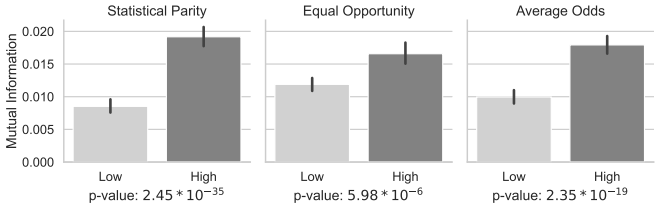


Fig. 3. Mean and 95% confidence interval of Mutual Information between binary variables and ground truth labels grouped by high and low SP, EO, and AO values. For each metric, we report the Welch's t-test p-value.

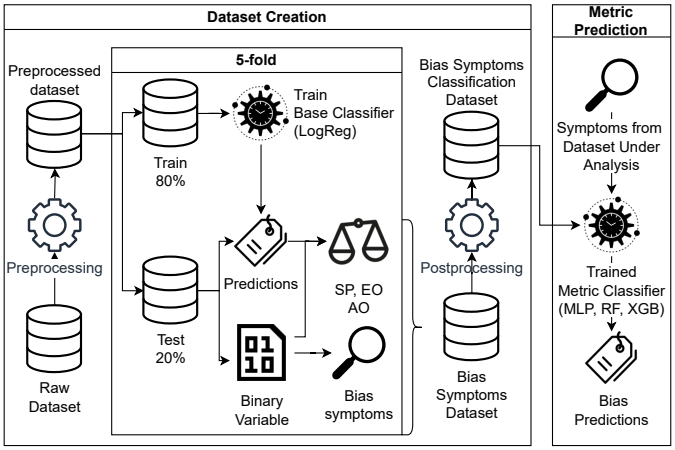


Fig. 4. Dataset creation and metric prediction workflow

mean Mutual Information is higher for items with higher bias values.

C. Proposed Approach

Figure 4 shows an overview of the proposed approach. In particular, we identify two macro-phases: *Dataset Creation* and *Metric Prediction*. The former concerns the extraction and collection of the 13 symptoms described in Section III-B to create a dataset of bias symptoms. The latter involves the prediction of the values referred to the fairness metrics defined in Section III-A.

Dataset Creation. We employed 24 tabular datasets from the literature on bias and fairness [1], [28], [35], [52]. The list of the employed datasets and additional statistics are reported in the online appendix, which includes also a replication package [20]. In particular, we selected datasets that are: publicly available, suitable for classification tasks, and contain at least one binary variable different from the label. It is worth noticing that, differently from other works on fairness [5], [10], [53], we employ both binary and multiclass datasets, i.e., where the possible values of the label are > 2.

Before starting the symptoms extraction process, all the datasets were preprocessed by removing missing values and one-hot-encoding categorical columns. After this process, each dataset has been split into training (80%) and testing (20%) sets. The train set has been used to train a Logistic Regression (LogReg) *Base Classifier* [54]. Previous works on fairness have driven our selection of LogReg as a base classifier [10], [36], [53], [55]. Following the same related works, we employed the LogReg implementation from the *scikit-learn* Python library [56] with default hyperparameters. After training the model, we use the test set to predict the label needed to compute ground truth values of SP, EO, and AO following the formulations provided in Section III-A. The test set has also been used to extract the bias symptoms. Following the definition of *privileged* and *unprivileged* groups [6], we selected all binary columns from each dataset and computed the bias metrics and symptoms for any of each. For each binary variable, we assumed that 0 identifies the unprivileged group, while 1 identifies the privileged group. The positive label of each dataset has been derived by its relative source paper and is reported in our online appendix [20]. To compute SP, EO, and AO, we adopted the implementations provided in [36], which are an extension of the metrics available in the *IBM AIF360* library [42] for the multiclass classification task. It is worth noticing that, as described in Section III-B, the bias symptoms are computed on the ground truth label of the testing set and not on the LogReg predictions. To increase the overall amount of collected bias symptoms, we repeat the whole train and test phase five times using a 5-fold approach (i.e., on each fold, we select a different subset of data for testing and the rest for training). The ground truth bias metrics and symptoms compose our *bias symptoms dataset*. However, since we are interested in detecting variables leading to *high* bias, we performed a postprocessing operation to map each metric value into low (i.e., 0) and high (i.e., 1) classes using the thresholds defined in Section III.

Metric Prediction. Next, the dataset has been used to train different *Metric Classifiers* for the early identification of variables leading to *High* bias. Given a new dataset to analyse, we first extract bias symptoms from it and then use them as input to the model to predict *High* and *Low* metric values. More details on this phase are provided in Section IV.

D. Bias symptoms dataset description

In the following, we report some statistics about the collected bias symptoms dataset.

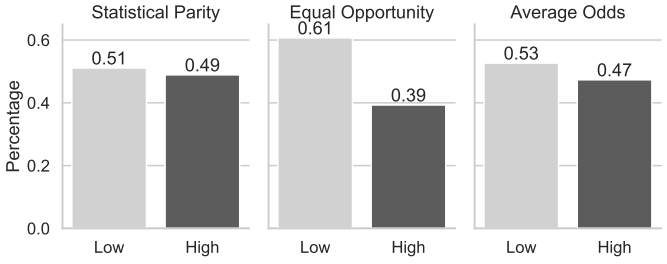


Fig. 5. Percentage of items with high and low values of SP, EO, and AO

The dataset comprises 5,930 instances and 16 features (i.e., the 13 symptoms defined in Section III-B and the three bias metrics). Figure 5 reports the percentage of instances with high and low SP, EO, and AO values based on the thresholds we defined above. We observe that, while the items with high and low values of SP and AO are pretty balanced, there is still a

tiny unbalance among items with a low EO value (61% vs 39%). On the one hand, EO values are generally lower than SP and AO. On the other hand, we believe that reducing the threshold for EO too much may lead to loose possible patterns between the symptoms and high values of this metric.

The comparison of SP, EO, and AO values is shown in Figure 6. We can observe how values for EO and AO are generally lower than those of SP. Especially EO has a median value close to zero. This can also be due to an overall low effectiveness of the base classifier in identifying true positives for both privileged and unprivileged groups.

After analyzing the metrics' distribution, we computed the Kendall τ correlation between the different symptoms and raw bias metrics (i.e., before the postprocessing mapping to 0 and 1) to glimpse the relationship between them. Table II reports the results of this analysis. In the table, correlations $> |0.4|$ are highlighted in grey, while non-statistically significant correlations (i.e., with a p-value ≥ 0.05) are marked with a *. Concerning the correlation between symptoms (first 13 rows and 13 columns in the table), we first observe a positive correlation between the *Privileged Unbalance* and the *Privileged Positive Probability (PPP)* (0.643), meaning that the more the privileged group is oversampled with respect to the positive label, the higher the probability of having an associated positive label. Surprisingly, we do not observe the same correlation for the unprivileged group. Instead, we observe a positive correlation between the *Unprivileged Positive Probability (UPP)* and the *Simpson* and *Shannon* indexes (0.571 and 0.566, respectively). This means that the more a variable is diverse, the higher the probability of having a positive label for the unprivileged group. Next, we observe a negative correlation between the *Privileged* and *Unprivileged Unbalance* (-0.428), which can be explained by the definition of these two values [36], [45]. We also observe positive correlations between the *Kurtosis*, *Skewness*, and *Gini* indexes (0.837 between *Skewness* and *Kurtosis*, 0.824 between *Gini* and *Kurtosis*, and 0.984 between *Gini* and *Skewness*). Since all these metrics measure the variability of a variable, a positive correlation between them is expected [57]. For the same reason, we also observe a negative correlation between these symptoms and the *Imbalance Ratio (IR)* (-0.984, -0.836, and -0.834).

Concerning the correlations between symptoms and bias metrics (last three rows and first 13 columns of the table or vice versa), we observe only a medium-high negative correlation between SP and the *Privileged Unbalance* (-0.45). This means that the SP increases when the privileged group is more balanced or undersampled with respect to the positive label. However, it is worth noticing also a slight positive correlation between SP and the *Unprivileged Unbalance* (0.346) and a slight negative correlation between SP and *PPP* (-0.326). A similar slightly positive correlation is observed between AO and *Skewness* and *Kurtosis* (0.328 and 0.335 respectively) and a slightly negative correlation between AO and the *Privileged Unbalance* (-0.309). EO has, instead, a low correlation with all the identified symptoms. Finally, concerning the correlation between bias metrics (the last three rows and columns of the table), we observe a positive correlation between SP and AO (0.544), meaning that an increase in one metric also implies an increase in the other.

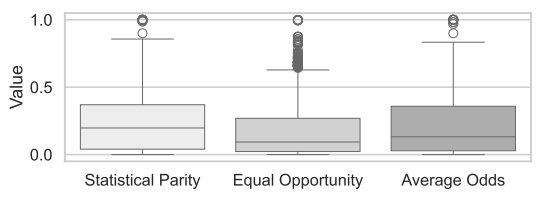


Fig. 6. Median and inter-quartile range of SP, EO and AO values

We can take the following three main takeaways from this analysis of correlations: *i)* high correlations between symptoms can be mostly explained by their definitions, and we do not observe any unexpected high correlation between different symptoms; *ii)* EO is not medium-highly correlated (i.e., correlations $\geq |0.3|$) with any of the identified symptoms; *iii)* SP and AO are positively correlated, meaning that the increase or decrease of one metric implies the increase or decrease of the other.

IV. Evaluation methods and materials

In this section, we describe the experimental evaluation conducted on our work. In particular, we aim to answer the following research questions (RQ):

➢ **RQ$_1$:** *How effective are the identified symptoms in predicting if Statistical Parity, Equal Opportunity, and Average Odds are above or below a given threshold?* This RQ analyses how an ML classifier trained with the bias symptoms can correctly predict if SP, EO, and AO are above or below the thresholds defined in Section III.

➢ **RQ$_2$:** *Which symptoms are the most helpful in predicting if Statistical Parity, Equal Opportunity, and Average Odds are above or below a given threshold?* This RQ aims to identify which symptoms are the most significant for the ML classifier concerning the prediction task performed in RQ1.

➢ **RQ$_3$:** *How general are the identified symptoms concerning the underlying base classifier?* This RQ focuses on analyzing if a change of the base classifier in the *Dataset Creation* phase (see Section III-C) significantly impacts the effectiveness of the ML classifier trained with the derived symptoms.

➢ **RQ$_4$:** *How efficient is an ML model trained with the bias symptoms compared to traditional approaches for fairness assessment in terms of execution time and CPU consumption?* To motivate our research, this RQ analyses the computational efficiency of an ML model trained with the identified symptoms compared with baseline approaches for fairness assessment. The comparison is performed in terms of execution time and CPU consumption.

A. Experimental Settings

RQ$_1$: We employ three ML classifiers to predict *High* or *Low* values of SP, EO and AO. The classifiers are Multi-Linear Perceptron (MLP), Random Forest (RF), and Extreme

TABLE II
KENDALL'S τ CORRELATION BETWEEN SYMPTOMS AND BIAS METRICS. CORRELATIONS HIGHER THAN $|0.4|$ ARE HIGHLIGHTED. NON-STATISTICALLY SIGNIFICANT CORRELATIONS (p-VALUE ≥ 0.05) ARE MARKED WITH A *

	Kendall τ	Mutual Info	UPP	PPP	Unpriv Unbal	Priv Unbal	Kurtosis	Skewness	Gini	Simpson	Shannon	IR	Prob Diff	SP	EO	AO
Kendall τ	**1.0**	0.015*	0.018	-0.109	-0.038	-0.071	-0.002*	0.015*	0.016*	0.026	0.025	0.001*	0.042	-0.022	-0.172	0.009*
Mutual Info	0.015*	**1.0**	0.148	0.031	0.083	-0.007*	-0.290	-0.274	-0.275	0.275	0.272	0.295	0.341	0.022	0.017	0.018
UPP	0.018	0.148	**1.0**	0.333	0.132	-0.001*	-0.120	-0.059	-0.055	**0.571**	**0.566**	0.125	0.105	0.062	0.005	0.059
PPP	-0.109	0.031	0.333	**1.0**	-0.255	**0.643**	-0.155	-0.143	-0.146	0.356	0.354	0.157	-0.020	-0.326	0.008	-0.227
Unpriv Unbal	-0.038	0.083	0.132	-0.255	**1.0**	**-0.428**	-0.069	-0.001*	0.007*	0.026	0.027	0.069	-0.0*	0.346	-0.028	0.250
Priv Unbal	-0.071	-0.007*	-0.001*	**0.643**	**-0.428**	**1.0**	-0.104	-0.117	-0.118	0.093	0.092	0.105	-0.074	**-0.45**	0.004	-0.309
Kurtosis	-0.002*	-0.290	-0.120	-0.155	-0.069	-0.104	**1.0**	**0.837**	**0.824**	-0.331	-0.321	**-0.984**	0.309	0.038	0.101	0.264
Skewness	0.015*	-0.274	-0.059	-0.143	-0.001*	-0.117	**0.837**	**1.0**	**0.984**	-0.277	-0.268	**-0.836**	0.262	0.048	0.082	0.328
Gini	0.016*	-0.275	-0.055	-0.146	0.007*	-0.118	**0.824**	**0.984**	**1.0**	-0.280	-0.272	**-0.834**	0.254	0.056	0.083	0.335
Simpson	0.026	0.275	**0.571**	0.356	0.026	0.093	-0.331	-0.277	-0.280	**1.0**	**0.99**	0.336	0.065	-0.033	-0.003	-0.047
Shannon	0.025	0.272	**0.566**	0.354	0.027	0.092	-0.321	-0.268	-0.272	**0.99**	**1.0**	0.326	0.066	-0.033	-0.003	-0.046
IR	0.001*	0.295	0.125	0.157	0.069	0.105	**-0.984**	**-0.836**	**-0.834**	0.336	0.326	**1.0**	-0.306	-0.039	-0.102	-0.264
Prob Diff	0.042	0.341	0.105	-0.020	-0.0*	-0.074	0.309	0.262	0.254	0.065	0.066	-0.306	**1.0**	0.025	0.124	0.242
SP	-0.022	0.022	0.062	-0.326	0.346	**-0.45**	0.038	0.048	0.056	-0.033	-0.033	-0.039	0.025	**1.0**	0.033	**0.544**
EO	-0.172	0.017	0.005	0.008	-0.028	0.004	0.101	0.082	0.083	-0.003	-0.003	-0.102	0.124	0.033	**1.0**	0.138
AO	0.009*	0.018	0.059	-0.227	0.250	-0.309	0.264	0.328	0.335	-0.047	-0.046	-0.264	0.242	**0.544**	0.138	**1.0**

Gradient Boosting (XGB). We adopted these methods because they have been widely adopted for classification tasks [5], [10], [36], [37] and natively support multi-label predictions (i.e., prediction tasks where the number of labels to predict is > 1, in our case the three bias metrics). We employ the Python implementation of the ML methods [56], [58] and, for each prediction task, we first perform a 5-fold grid search optimization of their hyper-parameters. The list of adopted hyper-parameters for each task is reported in our online appendix [20].

RQ_2: We first train the classifiers employed for RQ_1 with the full bias symptoms dataset and then analyze feature importance in predicting the single metrics. To assess feature importance, we use the widely adopted *permutation importance* technique, which is a model-agnostic approach that involves randomly shuffling the values of a single feature and observing the resulting degradation of the model's score [59]. For each model, we permute each dataset feature 10 times.

RQ_3: We adopted RF and MLP as base classifiers in the *Dataset Creation* process described in Fig. 4. The choice of these methods has been driven by the fact that they have been used in previous works on fairness [10], [36], [37] and natively support multi-class classification tasks. Following the works cited above, we adopted the default hyper-parameters from the *scikit-learn* Python library. After extracting the bias symptoms from these new base classifiers, we repeat the same evaluation process to address RQ_1.

RQ_4: We compare the computational efficiency of an XGBoost classifier trained with bias symptoms[4] with two benchmarks: an approach that replicates a standard fairness assessment pipeline [12] using the adaptation of the AIF360 metrics for multi-class classification proposed in [36] and the widely-adopted *Aequitas* fairness assessment toolkit [43]. In particular, we measure the training time and the CPU consumption required to assess the amount of bias for all the 24 datasets discussed in Section III concerning the sensitive variables reported in their respective papers (see our online appendix [20]). For both baselines, we first train a LogReg classifier using 80% of the data and then compute bias metrics using the remaining 20%. Note that we do not employ any of the approaches considered in Section II-B because no one fully covers the two features considered in our work (i.e., sensitive variable detection and metric prediction). These experiments were conducted on a Dell XPS 13 2019 with Ubuntu 22.04.4 LTS, an Intel Core i7 processor, and 16GB of RAM, with no other background process running.

[4]We chose this method because it was the one with better computational efficiency among the three employed classifiers and, as reported in Section V-A, there is no statistical difference in the effectiveness of the three models

B. Metrics

We adopt *Accuracy (Acc)* [60], *Precision (Prec)* [61], *Recall (Rec)* [61], *F1 Score (F1)* [22], and *Average Precision (AP)* [62] to assess the effectiveness of the prediction of *High* or *Low* bias metric values. We choose *Acc*, *Prec*, *Rec*, and *F1* because they have been widely adopted in previous fairness works [9], [10]. *AP* is instead defined as the area under the *Precision-Recall curve* and allows us to have a more comprehensive view of the prediction effectiveness at different classification thresholds. We adopted this metric instead of *AUC-ROC Score* because AP has been shown to be more reliable in case of unbalanced data (which is our case concerning EO distributions, see figure 5) [63]. To avoid possible data selection bias in the computation of effectiveness metrics, we perform a 5-fold cross-validation, using symptoms extracted from 80% of the original biased datasets for training and the rest for testing.

For the RQ_2, we consider AP as a reference metric to compute feature importance since it gives a wider view of a model's effectiveness [63]. For the RQ_4, we measure the CPU consumption in CPU Power Package (*pkg*) which is the power that is consumed by the CPU cores, cache, and other cores to answer while the execution time is expressed in seconds.

C. Statistical Tests

We perform the *Kruskal-Wallis H-test*, i.e., a non-parametric test to verify the null hypothesis that the population medians of multiple groups are equal [64], to check if there is any statistically significant difference between the effectiveness of the employed classifiers in RQ_1 and RQ_3. In addition, we perform the *Mann-Whitney U test* (a nonparametric test of

the null hypothesis that the distribution of two samples is the same [65]) to group symptoms leading to a non-statistically significant AP loss in the RQ_2. The same test was also used in the RQ_4 to assess if the differences in CPU consumption and execution times are statistically significant. Following previous works [9], a statistic is significant if its p-value is < 0.05.

V. Results

A. Addressing RQ_1

Table III reports the mean and standard deviation of the effectiveness metrics described in Section IV-B. In the table, rows refer to effectiveness metrics, and columns refer to the employed classification methods and bias metrics (i.e., *SP*, *EO*, and *AO*). The rightmost group reports the *Average* effectiveness of a model between the predictions for all metrics. For each bias metric, the best results are highlighted in grey.

It is important to note that the SP metric has achieved the highest results in terms of bias, with the RF classifier producing the best mean accuracy of 0.779 and the best mean F1 score of 0.715 overall the experiments. Moreover, the results indicate high precision and relatively low recall (best mean values of 0.832 and 0.692, respectively), which implies that there may be many false negatives in the results. However, the high values for average precision (with the best mean value of 0.823 for RF) suggest that modifying the classification threshold of the models could lead to better precision and recall.

Contrariwise, we witness a degradation of performance for the other two considered metrics, i.e., EO and AO accuracy scores are lower compared to SP. Regarding the EO metric, we observe low precision, recall, and, consequently, F1 score (best mean values of 0.651, 0.552, and 0.579 for XGBoost, respectively). The average precision values also align with these results (with a best mean value of 0.659 for XGBoost), meaning these low results are not related to a wrong classification threshold. However, we also observe a high standard deviation among all the results, meaning that the value of these metrics changes significantly when different training and testing sets are employed. This may be related to the unbalance of high and low values of EO (see Section III-D) and to the overall unequal distribution of datasets' binary variables in the dataset bias symptoms (see Section VI), leading to situations in which the number of high EO values is very low in the training set.

Concerning AO metrics, it presents a slightly higher variability in the results than SP, but not as high as EO. In addition, the balanced distribution of high and low values (see Section III-D) leads to the exclusion that the low effectiveness is caused by a data unbalance problem. Instead, the low effectiveness is mainly related to a systematic low ability of the identified symptoms to represent high and low values of AO. However, the results are not harmful (i.e., an average precision of 0.701 for XGBoost), meaning that the identified symptoms are not meaningless in predicting high and low values of AO.

Finally, from the *Average* category, we observe no clear winner between the three classification models employed in predicting high and low values of the bias metrics. This similarity has also been confirmed by the *H-test*, which did not report a statistically significant difference between the effectiveness results of the three models for all bias metrics.

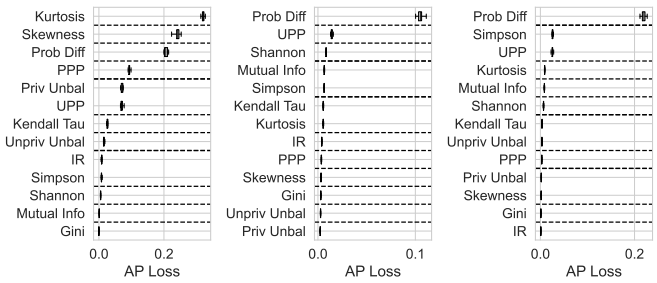
Answer to RQ_1: Using the identified symptoms, the involved ML classifiers can predict *High* and *Low* values of SP with high effectiveness, while predictions for EO and AO are less effective.

B. Addressing RQ_2

Figure 7 reports the results of the permutation importance for MLP, RF, and XGB in predicting high and low values of SP. In each plot, features are ordered in ascending order, where features on the top are the most important. A dotted line separates features whose difference in AP loss is statistically significant. Hence, in the following, we assume that features not separated by a line have the same importance. From the figure, we can observe how Random Forest and XGBoost share the two most important symptoms, namely *Probability Difference* and *Unprivileged Positive Probability (UPP)* (for XGBoost, UPP has the same statistical importance of *Simpson*). Furthermore, we observe a significant difference in AP loss between the first and second most important symptoms for both classifiers, meaning that *Prob Diff* is considered much more important compared with other symptoms. *Kurtosis* is instead the most important feature for MLP, followed by *Skewness*. However, for MLP, we observe less difference in AP loss between features, meaning that the importance assigned to each symptom is more spread.

Figure 8 reports the feature importance for EO. In this case, MLP and XGBoost share the most important symptom, which is *Kurtosis*. Particularly for MLP, *Kurtosis* has high importance since its permutation causes an AP loss of almost 50%. Results for RF are less meaningful since a permutation of the symptoms causes, at most, an AP loss of slightly more than 2%. We also observe how the AP loss of XGBoost is lower compared with SP and more spread among symptoms.

Finally, Figure 9 reports the feature importance for AO. We observe how, like for SP, RF and XGBoost share *Probability Difference* and *UPP* as the first and second most important symptom, respectively. This can be related to the high corre-

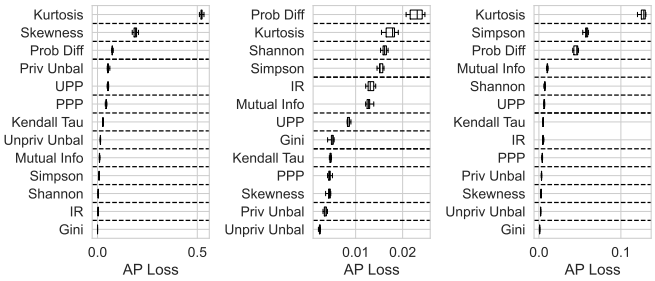


(a) MLP (b) Random Forest (c) XGBoost

Fig. 7. Permutation importance for Statistical Parity (SP)

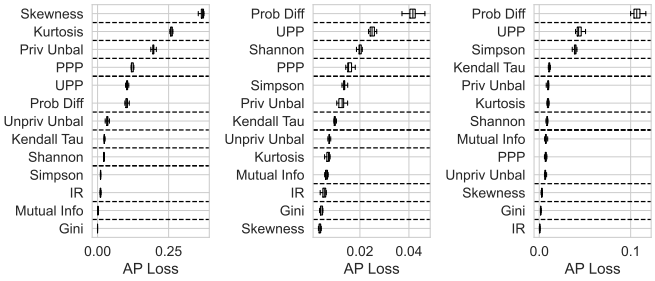
TABLE III
MEAN AND STANDARD DEVIATION OF EFFECTIVENESS METRICS FOR MLP, RF AND XGBOOST IN PREDICTING HIGH AND LOW VALUES OF EACH BIAS METRIC. THE LAST COLUMN REPRESENTS THE AVERAGE EFFECTIVENESS IN PREDICTING ALL METRICS. THE BEST RESULTS ARE HIGHLIGHTED

Metrics	Statistical Parity (SP)			Equal Opportunity (EO)			Average Odds (AO)			Average		
	MLP	RF	XGBoost	MLP	RF	XGBoost	MLP	RF	XGBoost	MLP	RF	XGBoost
AP	0.71 ± 0.052	**0.823 ± 0.075**	0.783 ± 0.114	0.562 ± 0.234	0.646 ± 0.304	**0.659 ± 0.299**	0.673 ± 0.154	0.694 ± 0.152	**0.701 ± 0.154**	0.648 ± 0.166	**0.721 ± 0.202**	0.714 ± 0.197
Acc	0.695 ± 0.099	**0.779 ± 0.077**	0.766 ± 0.09	0.751 ± 0.078	0.741 ± 0.077	**0.76 ± 0.072**	**0.676 ± 0.074**	0.664 ± 0.087	0.654 ± 0.106	0.707 ± 0.085	**0.728 ± 0.089**	0.727 ± 0.099
Prec	0.684 ± 0.109	**0.832 ± 0.043**	0.797 ± 0.089	0.625 ± 0.244	0.636 ± 0.34	**0.651 ± 0.319**	0.604 ± 0.201	0.677 ± 0.175	**0.687 ± 0.192**	0.638 ± 0.182	**0.715 ± 0.224**	0.712 ± 0.214
Rec	**0.692 ± 0.224**	0.644 ± 0.164	0.666 ± 0.2	0.494 ± 0.208	0.494 ± 0.21	**0.552 ± 0.177**	**0.721 ± 0.132**	0.621 ± 0.202	0.655 ± 0.273	**0.635 ± 0.206**	0.586 ± 0.191	0.624 ± 0.211
F1	0.662 ± 0.106	**0.715 ± 0.114**	0.707 ± 0.133	0.542 ± 0.211	0.529 ± 0.264	**0.579 ± 0.252**	**0.644 ± 0.15**	0.609 ± 0.045	0.606 ± 0.072	0.616 ± 0.159	0.618 ± 0.174	**0.631 ± 0.167**



(a) MLP (b) Random Forest (c) XGBoost

Fig. 8. Permutation importance for Equal Opportunity (EO)



(a) MLP (b) Random Forest (c) XGBoost

Fig. 9. Permutation importance for Average Odds (AO)

lation between SP and AO shown in Table II. However, in this case, their permutation generates a lower AP loss, especially for RF (around 4%), and the loss is, in general, more spread among symptoms. Finally, it is interesting to note how *Kurtosis* and *Skewness* have consistently been reported as the most important features of MLP. Altogether, we observe how there is no single symptom that emerges as the most important even though GB and RF both report significant importance of the *Prob Diff* in predicting high and low SP.

Answer to RQ$_2$: Our findings suggest a disagreement between the three classifiers concerning the most relevant symptoms. However, both XGB and RF models report a significant importance of the *Prob Diff* in predicting SP values.

C. Addressing RQ_3

Table IV reports the mean and standard deviation of average precision for MLP, RF, and XGBoost in predicting bias metrics

TABLE IV
MEAN AND STANDARD DEVIATION OF AP FOR MLP, RF AND XGBOOST USING MLP AND RANDOM FOREST BASE MODELS

	MLP Base Classifier			
	SP	**EO**	**AO**	**Average**
MLP	0.757 ± 0.2	0.556 ± 0.3	0.676 ± 0.14	0.663 ± 0.23
RF	**0.816 ± 0.172**	0.713 ± 0.3	0.737 ± 0.154	0.755 ± 0.2
XGBoost	0.814 ± 0.158	**0.746 ± 0.245**	**0.755 ± 0.12**	**0.772 ± 0.171**
	Random Forest Base Classifier			
	SP	**EO**	**AO**	**Average**
MLP	0.745 ± 0.08	0.578 ± 0.274	0.67 ± 0.16	0.665 ± 0.188
RF	**0.867 ± 0.048**	0.636 ± 0.333	**0.752 ± 0.134**	**0.752 ± 0.217**
XGBoost	0.828 ± 0.072	**0.645 ± 0.319**	0.743 ± 0.139	0.739 ± 0.205

employing MLP and RF base classifiers.[5] As done for the RQ$_1$, the best results are highlighted. The results confirm the trend shown in the RQ$_1$: SP is the bias metric achieving the highest effectiveness among the three, metrics for EO present a high standard deviation, and metrics for AO are generally less effective. In addition, like for RQ$_1$, the *H-test* did not report a statistically significant difference among the effectiveness metrics in all the analyzed cases, meaning that the analyzed classifiers do not impact the overall process.

Answer to RQ$_3$: Adopting different base classifiers for creating the bias symptoms dataset does not impact the overall effectiveness in predicting *High* and *Low* SP, EO, and AO values.

D. Addressing RQ_4

TABLE V
MEAN AND STANDARD DEVIATION OF CPU CONSUMPTION AND EXECUTION TIME OF AEQUITAS, STANDARD AND XGBSYMP

	Binary		All Data	
	CPU Cons (pkg)	**Time (s)**	**CPU Cons (pkg)**	**Time (s)**
Aequitas	$2.86 * 10^6 \pm 2.81 * 10^6$	0.10 ± 0.09	-	-
Standard	$11.12 * 10^6 \pm 15.55 * 10^6$	0.52 ± 0.72	$1.769 * 10^8 \pm 7.26 * 10^8$	9.94 ± 40.95
XGBSymp	$1.81 * 10^6 \pm 1.75 * 10^6$	0.07 ± 0.07	$2.51 * 10^6 \pm 8.56 * 10^6$	0.12 ± 0.34

To answer this question, we measure the CPU consumption of our approach and of the two identified baselines. Concerning the CPU consumption, we make use of the PyRapl Python library[6] to get the energy footprint of each method. Table V

[5]For space constraints, we report only these results while the full metrics are reported in our appendix [20]

[6]https://github.com/powerapi-ng/pyRAPL/tree/master

reports the mean and standard deviation of CPU consumption and execution time for fairness assessment using respectively *Aequitas*, a standard pipeline that employs an extension of the AIF360 metrics for multi-class classification (*Standard*), and an XGBoost model trained with the bias symptoms (*XGB-Symp*). Since *Aequitas* does not support fairness assessment on multi-class data, we performed two sets of evaluations: one employing only binary data and one using all 24 datasets described in Section III-C. The experiment shows that *XGB-Symp* exhibits significantly better computational performance in all evaluations, particularly when multi-class data is used. Although the training and hyperparameter optimization of *XGBSymp* are computationally intensive (with an average CPU consumption of $1.61 * 10^{10} \pm 9.67 * 10^{8}$ and training time of 680.63 ± 8.29 seconds), we argue that this training only needs to be done once. In contrast, other approaches require training an ML model for each fairness assessment. In the case of the *Standard* approach, this implies an average CPU consumption of $1.769 * 10^{8}$ pkg and an execution time of 9.94 seconds for each fairness assessment.

Answer to RQ_4: The conducted experiments show that the XGBoost model trained with the proposed bias symptoms is computationally more efficient than standard fairness assessment approaches regarding time and CPU consumption.

E. Discussion

The experiment found that the identified *bias symptoms* can effectively predict *High* and *Low* values of SP, regardless of the base classifier used to build the dataset. Two out of three classifiers (RF and XGBoost) reported that *Probability Difference* is the most important symptom to predict SP. This is because *Probability Difference* is already a version of SP that uses ground truth values instead of predictions. Therefore, it can effectively intercept the bias learned by the model if the dataset is already biased. On the other hand, MLP considers the distribution of a variable (*Skewness* and *Kurtosis*) to be an indicator of high and low values for all the bias metrics. The low effectiveness of the predictions for EO and AO can be explained, aside from the imbalance between high and low EO values, because no symptoms can effectively represent true and false positive rates, which are considered in the two metric formulations. However, this does not mean that the identified symptoms are meaningless for detecting high and low values of these metrics. Results for EO and AO are not entirely negative since we always obtain effectiveness results that are, on average, better than the ones of a random classifier (i.e. > 0.5). Finally, the high computational efficiency shown by an XGBoost model trained with bias symptoms compared to traditional approaches for fairness assessment motivates further research in this direction.

VI. Threats to Validity

This section discusses the threats that may hamper the results of our study.

Threats to *internal validity* concern the process adopted to build the bias symptoms dataset. Some of the considered datasets have a higher number of sensitive variables, thus having a higher representation in the bias symptoms dataset. To address this, we performed a 5-fold cross-validation to answer RQ_1 and RQ_3, mitigating a possible data selection bias in the evaluation. Furthermore, we repeated the statistics reported in Section III and the answer to RQ_2 filtering out datasets with a number of binary variables higher than 75% of the whole set. The results are reported in our appendix [20] and confirm all the most significant reported claims. Concerning the fairness metric prediction, the employed threshold may impact negatively the results. We motivated the selection of the thresholds in Section II, while further research can investigate the impact of adopting different thresholds. Finally, there could be other relevant bias symptoms that are not considered in this first version of the bias symptoms dataset. To address this, we motivated the selection of each symptom in Section III, while future research can investigate the impact of other symptoms to overcome the limitations highlighted in Section IV.

Regarding the *external validity* of our approach, we acknowledge that selecting a fixed number of datasets may limit the generalizability of our findings as the identified bias symptoms may produce varying results. However, to address this issue, we used 24 tabular datasets that cover a wide range of application domains, from social to software systems. We also recognize that our analysis only focuses on the group fairness definition and three metrics (SP, EO, and AO). Nonetheless, we have covered the most commonly used fairness metrics in the current literature and highlighted how several studies have proposed solutions to address other sources of bias, such as *algorithmic bias* [9].

Construction validity concerns the conducted empirical experiments to evaluate the accuracy, and the efficiency of using bias symptoms. Concerning the former, we experiment with three different ML baseline models that have been evaluated with the cross-fold validation strategy. Concerning the latter, we carried out an empirical evaluation including two existing baselines to measure the execution time and energy consumption using a dedicated Python library.

VII. Conclusion

Motivated by the need to automate fairness auditing processes, this paper proposed a new approach to automatically identify potentially biased variables in datasets. We call these *bias symptoms* and extracted them from 24 fairness benchmarking datasets. We then used the synthesized dataset to train three advanced classifiers and predict the value of three fairness metrics with a certain degree of tolerance. Our results show that our approach can help streamline the fairness auditing process by striking a balance between accuracy and efficiency. For future work, we plan to expand the datasets to cover more domains and consider additional baseline models and fairness metrics. Additionally, we believe that our *bias symptoms* can be used to suggest the most appropriate fairness metric for a specific application domain.

REFERENCES

[1] A. Fabris, S. Messina, G. Silvello, and G. A. Susto, "Algorithmic fairness datasets: the story so far," *Data Mining and Knowledge Discovery*, vol. 36, no. 6, pp. 2074–2152, Nov. 2022. [Online]. Available: https://doi.org/10.1007/s10618-022-00854-z

[2] A. Olteanu, C. Castillo, F. Diaz, and E. Kıcıman, "Social data: Biases, methodological pitfalls, and ethical boundaries," *Frontiers in big data*, vol. 2, p. 13, 2019.

[3] J. Chakraborty, T. Xia, F. M. Fahid, and T. Menzies, "Software Engineering for Fairness: A Case Study with Hyperparameter Optimization," Oct. 2019, arXiv:1905.05786 [cs]. [Online]. Available: http://arxiv.org/abs/1905.05786

[4] C. Ferrara, G. Sellitto, F. Ferrucci, F. Palomba, and A. De Lucia, "Fairness-aware machine learning engineering: how far are we?" *Empirical Software Engineering*, vol. 29, no. 1, p. 9, Nov. 2023. [Online]. Available: https://doi.org/10.1007/s10664-023-10402-y

[5] J. Chakraborty, S. Majumder, Z. Yu, and T. Menzies, "Fairway: a way to build fair ML software," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 654–665.

[6] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–35, Jul. 2021.

[7] European Commission, "Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts," 2021. [Online]. Available: https://eur-lex.europa.eu/resource.html?uri=cellar:e0649735-a372-11eb-9585-01aa75ed71a1.0001.02/DOC_1&format=PDF

[8] Y. Brun and A. Meliou, "Software fairness," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. Lake Buena Vista FL USA: ACM, Oct. 2018, pp. 754–759.

[9] Z. Chen, J. M. Zhang, M. Hort, F. Sarro, and M. Harman, "Fairness Testing: A Comprehensive Survey and Analysis of Trends," Aug. 2022, arXiv:2207.10223 [cs]. [Online]. Available: http://arxiv.org/abs/2207.10223

[10] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, "Fairness Improvement with Multiple Protected Attributes: How Far Are We?" Apr. 2024, conference Name: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE) Meeting Name: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE) Place: Lisbon, Portugal Publisher: IEEE/ACM Volume: 46. [Online]. Available: https://www.computer.org/csdl/proceedings/icse/2024/1RLIVDkr2xO

[11] H. Muccini and K. Vaidhyanathan, "Software Architecture for ML-based Systems: What Exists and What Lies Ahead," in *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, May 2021, pp. 121–128.

[12] G. d'Aloisio, A. Di Marco, and G. Stilo, "Democratizing quality-based machine learning development through extended feature models," in *Fundamental Approaches to Software Engineering*, L. Lambers and S. Uchitel, Eds. Cham: Springer Nature Switzerland, 2023, pp. 88–110.

[13] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, "MAAT: a novel ensemble approach to addressing fairness and performance bugs for machine learning software," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 1122–1134. [Online]. Available: https://dl.acm.org/doi/10.1145/3540250.3549093

[14] M. Hort, J. M. Zhang, F. Sarro, and M. Harman, "Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 994–1006. [Online]. Available: https://doi.org/10.1145/3468264.3468565

[15] M. H. Asyrofi, Z. Yang, I. N. B. Yusuf, H. J. Kang, F. Thung, and D. Lo, "Biasfinder: Metamorphic test generation to uncover bias for sentiment analysis systems," *IEEE Transactions on Software Engineering*, vol. 48, no. 12, pp. 5087–5101, 2022.

[16] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner, "Exploring the Carbon Footprint of Hugging Face's ML Models: A Repository Mining Study," in *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Oct. 2023, pp. 1–12, arXiv:2305.11164 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2305.11164

[17] R. Verdecchia, P. Lago, C. Ebert, and C. De Vries, "Green it and green software," *IEEE Software*, vol. 38, no. 6, pp. 7–15, 2021.

[18] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual Fairness," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html

[19] M. Hardt, E. Price, E. Price, and N. Srebro, "Equality of Opportunity in Supervised Learning," in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

[20] Anonymous Author(s), "Replication package." [Online]. Available: https://anonymous.4open.science/r/sensitive_variables_detection-FC96/

[21] S. Caton and C. Haas, "Fairness in Machine Learning: A Survey," Oct. 2020, arXiv:2010.04053 [cs, stat].

[22] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

[23] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, p. 625–637, jun 2011. [Online]. Available: https://doi.org/10.1016/j.infsof.2010.12.010

[24] B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5XW20.

[25] V. N. Dornadula and S. Geetha, "Credit card fraud detection using machine learning algorithms," *Procedia computer science*, vol. 165, pp. 631–641, 2019.

[26] L. Oneto, M. Doninini, A. Elders, and M. Pontil, "Taking Advantage of Multitask Learning for Fair Classification," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '19. New York, NY, USA: Association for Computing Machinery, Jan. 2019, pp. 227–237. [Online]. Available: https://dl.acm.org/doi/10.1145/3306618.3314255

[27] M. Openja, G. Laberge, and F. Khomh, "Detection and evaluation of bias-inducing features in machine learning," *Empirical Software Engineering*, vol. 29, no. 1, p. 22, Dec. 2023. [Online]. Available: https://doi.org/10.1007/s10664-023-10409-5

[28] C. Ferrara, F. Casillo, and C. Gravino, "ReFAIR: Toward a Context-Aware Recommenderfor Fairness Requirements Engineering," 2024.

[29] S. Galhotra, K. Shanmugam, P. Sattigeri, and K. R. Varshney, "Causal Feature Selection for Algorithmic Fairness," in *Proceedings of the 2022 International Conference on Management of Data*, ser. SIGMOD '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 276–285. [Online]. Available: https://dl.acm.org/doi/10.1145/3514221.3517909

[30] M. Mecati, A. Vetrò, and M. Torchiano, "Detecting Discrimination Risk in Automated Decision-Making Systems with Balance Measures on Input Data," in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 4287–4296. [Online]. Available: https://ieeexplore.ieee.org/document/9671443

[31] W. Yik, L. Serafini, T. Lindsey, and G. D. Montañez, "Identifying Bias in Data Using Two-Distribution Hypothesis Tests," in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '22. New York, NY, USA: Association for Computing Machinery, Jul. 2022, pp. 831–844. [Online]. Available: https://dl.acm.org/doi/10.1145/3514094.3534169

[32] R. Constantin, M. Dück, A. Alexandrov, P. Matošević, D. Keidar, and M. El-Assady, "How Do Algorithmic Fairness Metrics Align with Human Judgement? A Mixed-Initiative System for Contextualized Fairness Assessment," in *2022 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*, Oct. 2022, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9974353

[33] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu, "Interventional Fairness: Causal Database Repair for Algorithmic Fairness," in *Proceedings of the 2019 International Conference on Management*

of Data. Amsterdam Netherlands: ACM, Jun. 2019, pp. 793–810. [Online]. Available: https://dl.acm.org/doi/10.1145/3299869.3319901

[34] J. D. Rocco and C. D. Sipio, "Resyduo: Combining data models and cf-based recommender systems to develop arduino projects," 2023.

[35] M. Hort, Z. Chen, J. M. Zhang, M. Harman, and F. Sarro, "Bias Mitigation for Machine Learning Classifiers: A Comprehensive Survey," *ACM Journal on Responsible Computing*, p. 3631326, Nov. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3631326

[36] G. d'Aloisio, A. D'Angelo, A. Di Marco, and G. Stilo, "Debiaser for Multiple Variables to enhance fairness in classification tasks," *Information Processing & Management*, vol. 60, no. 2, p. 103226, Mar. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457322003272

[37] S. Biswas and H. Rajan, "Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness," in *Proceedings of the 28th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, 2020, pp. 642–653.

[38] ——, "Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline," in *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, 2021, pp. 981–993.

[39] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS '12. New York, NY, USA: Association for Computing Machinery, Jan. 2012, pp. 214–226.

[40] A. Castelnovo, R. Crupi, G. Greco, D. Regoli, I. G. Penco, and A. C. Cosentini, "A clarification of the nuances in the fairness metrics landscape," *Scientific Reports*, vol. 12, no. 1, p. 4209, 2022.

[41] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and Removing Disparate Impact," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney NSW Australia: ACM, Aug. 2015, pp. 259–268. [Online]. Available: https://dl.acm.org/doi/10.1145/2783258.2783311

[42] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," *IBM Journal of Research and Development*, vol. 63, no. 4/5, pp. 4:1–4:15, Jul. 2019, conference Name: IBM Journal of Research and Development.

[43] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani, "Aequitas: A Bias and Fairness Audit Toolkit," *arXiv:1811.05577 [cs]*, Apr. 2019, arXiv: 1811.05577. [Online]. Available: http://arxiv.org/abs/1811.05577

[44] S. Majumder, J. Chakraborty, G. R. Bai, K. T. Stolee, and T. Menzies, "Fair enough: Searching for sufficient measures of fairness," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 6, pp. 1–22, 2023.

[45] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, Oct. 2012. [Online]. Available: http://link.springer.com/10.1007/s10115-011-0463-8

[46] C. Yang, Y. Akimoto, D. W. Kim, and M. Udell, "OBOE: Collaborative Filtering for AutoML Model Selection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1173–1183. [Online]. Available: https://dl.acm.org/doi/10.1145/3292500.3330909

[47] M. Cerrada, L. Trujillo, D. E. Hernández, H. A. Correa Zevallos, J. C. Macancela, D. Cabrera, and R. Vinicio Sánchez, "Automl for feature selection and model tuning applied to fault severity diagnosis in spur gearboxes," *Mathematical and Computational Applications*, vol. 27, no. 1, p. 6, 2022.

[48] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and Robust Automated Machine Learning," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf

[49] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Statistical methods & applications*, vol. 19, pp. 497–515, 2010.

[50] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test," *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, 2006.

[51] T. E. Duncan, "On the calculation of mutual information," *SIAM Journal on Applied Mathematics*, vol. 19, no. 1, pp. 215–220, 1970.

[52] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, "A survey on datasets for fairness-aware machine learning," *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 3, p. e1452, 2022, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1452. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1452

[53] M. Hort, J. M. Zhang, F. Sarro, and M. Harman, "Search-based Automatic Repair for Fairness and Accuracy in Decision-making Software," *Empirical Software Engineering*, vol. 29, no. 1, p. 36, Jan. 2024. [Online]. Available: https://doi.org/10.1007/s10664-023-10419-3

[54] S. Menard, *Applied logistic regression analysis*. Sage, 2002, vol. 106.

[55] G. d'Aloisio, G. Stilo, A. Di Marco, and A. D'Angelo, "Enhancing Fairness in Classification Tasks with Multiple Variables: A Data-and Model-Agnostic Approach," in *International Workshop on Algorithmic Bias in Search and Recommendation*. Springer, 2022, pp. 117–129.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[57] R. B. Bendel, S. S. Higgins, J. E. Teberg, and D. A. Pyke, "Comparison of skewness coefficient, coefficient of variation, and Gini coefficient as inequality measures within populations," *Oecologia*, vol. 78, no. 3, pp. 394–400, Mar. 1989. [Online]. Available: https://doi.org/10.1007/BF00379115

[58] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[59] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[60] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Australasian joint conference on artificial intelligence*. Springer, 2006, pp. 1015–1021.

[61] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994, publisher: Wiley Online Library.

[62] K. Boyd, K. H. Eng, and C. D. Page, "Area under the precision-recall curve: point estimates and confidence intervals," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. Springer, 2013, pp. 451–466.

[63] B. Ozenne, F. Subtil, and D. Maucort-Boulch, "The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases," *Journal of clinical epidemiology*, vol. 68, no. 8, pp. 855–859, 2015.

[64] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[65] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.