FALCC: Efficiently performing locally fair and accurate classifications

Nico Lässig
University of Stuttgart, IPVS
Stuttgart, Germany
nico.laessig@ipvs.uni-stuttgart.de

Melanie Herschel
University of Stuttgart, IPVS
Stuttgart, Germany
melanie.herschel@ipvs.uni-stuttgart.de

ABSTRACT

Machine learning (ML) models are often used for decision support. Besides accuracy, applications may want or even require fair predictions. This paper addresses the largely unexplored problem of improving fairness not only globally, but also in more specific or *local* regions of the data, while not compromising accuracy. FALCC comprises both a general framework to study various facets of the problem and efficient algorithms for fair and accurate dynamic model ensemble selection in order to induce local fairness in classifications. Efficiency is gained by precomputing optimal models for local regions encompassing similar samples in an offline phase, thereby reducing the online model selection to be used to classify new samples to a similarity lookup. To further improve the quality of the results in terms of fairness, we introduce techniques for model diversification and proxy discrimination mitigation in the offline phase. Experiments validate that FALCC is competitive in terms of achieved accuracy and local fairness, while being significantly more efficient in terms of prediction runtime for a new sample.

1 INTRODUCTION

Machine learning (ML) models are widely used to predict various events and recommend actions that should be taken. In many classification scenarios (e.g., credit scoring [44], crime recidivism [11], or hiring systems [57]), it is critical for these predictions to be fair and not to exhibit bias towards specific groups of people. One example application where the use of ML models has exhibited discrimination against women was a recruitment tool developed by Amazon [19]. Another example is the PredPol software, used in some areas of the United States for crime prediction, which has been linked to an increase in racial profiling [58].

Bias in ML predictions. In the above examples, biased decisions were made towards groups of different gender or race. Such attributes are called *sensitive attributes* (or *protected attributes*). *Sensitive groups* are the groups resulting from the combination of sensitive attribute values, e.g., groups defined by both gender and race such as {*black* , *non-binary* } or {*asian* , *female* }. Today, laws may actually forbid the consideration of such protected attributes, e.g., for training ML models. Even though less regulated, non-protected attributes may correlate with protected ones, effectively serving as *proxy attributes* that may still steer the training towards a biased ML model that exhibits indirect, so-called proxy discrimination [38, 61, 65]. While the consensus is that sensitive attributes, in general, should not impact the overall prediction in order to be fair, no single formal fairness definition has emerged. Instead, there are multiple fairness definitions [61].

Algorithms for fair ML. Algorithms to alleviate the problem of biased ML predictions have been proposed and divide into pre-processing, in-processing, and post-processing algorithms [35]. *Pre-processing* strategies alter the training data. *In-processing* methods focus on developing fair models, independent of the training data. Finally, *post-processing* methods either modify the resulting models or the predictions. Most work focus on binary classification, some further limiting to scenarios with a single binary sensitive attribute. While theoretically, any classification problem can be turned into a classification problem with only binary sensitive groups (one being the privileged group and the other one being the protected group), this may result in discrimination of persons that share several protected traits [28, 60].

Local vs. global fairness. Another differentiating factor of solutions for fair ML is the considered notion of fairness. Most existing algorithms consider a specific *global group fairness* definition, hence they consider fairness among groups over the whole dataset. We will refer to them as *global fairness*. Global fairness encompasses several more nuanced definitions, including *demographic parity* (aka *statistical parity*) [24], *equalized odds* [36], *equal opportunity* [36], or *treatment equality* [5]. The notion of *individual fairness* [61] defines that similar people are treated equally. Individual fairness metrics include, e.g., *fairness through awareness* [24] and *consistency* [77]. Measuring individual fairness for an individual t requires determining *local regions* of the data that comprise individuals similar to t, i.e., a local region is a subset of a dataset in which each sample has high similarity.

Group fairness and individual fairness may conflict [8], raising the question of whether a more unified view can be defined. Indeed, while it was acknowledged that the similarity of individuals is important for fair classifications [8], from a legal perspective, group fairness metrics are used to determine underlying discrimination [26]. Recent work has proposed *local fairness* [55] that combines the fairness notions defined for global fairness with the locality of individual fairness. Local fairness is achieved when each group (i.e., both protected and not protected) is treated equally (as assessed by a global fairness metric) for subsets of the population that share lots of traits (i.e., that are similar when disregarding protected attributes).

We illustrate global, individual, and local fairness using Fig. 1. It assumes a binary classification problem. Colors indicate if an individual (e.g., males and females) gets a raise (blue) or not (red). The figure also visualizes the similarity of individuals through their distance to each other. Considering a particular global fairness definition (e.g., demographic parity), the classification qualifies as fair, since $\frac{2}{3}$ of both males and females receive a raise. However, within the circled local region, no female gets a raise, as opposed to all four males. Thus, although the individuals within this region are very similar (e.g., similar education degree, years of experience), they are not treated equally. While individual metrics, such as consistency [77], would mark the women as being treated unfairly, the overall (average) individual fairness is

Series ISSN: 2367-2005 10.48786/edbt.2024.19

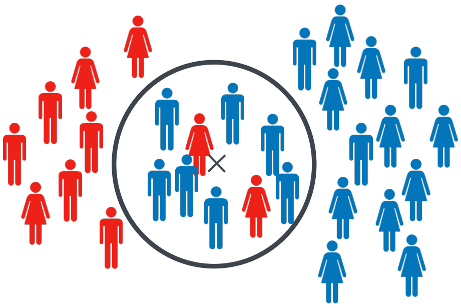


Figure 1: Example binary classification scenario

still high, as 7 out of 9 people within the region receive a positive outcome. In contrast, local fairness applies, e.g., demographic parity within the region, identifying it as unfair.

Contributions. This paper investigates solutions that support efficient and effective locally fair classifications. The goal is to minimize bias (defined by an existing global bias metric) for each local region of similar individuals. We focus on in-processing techniques, but note that achieving local fairness is independent of the processing technique, analogously to achieving global fairness. Thus, dedicated pre- or post-processing solutions are conceivable but out of the scope of this paper, which makes the following contributions.

We present a ***general framework*** to support efficient binary locally fair classifications for settings with possibly multiple, non-binary protected attributes, without jeoperdizing accuracy. Its general definition accommodates a large variety of algorithms with different fairness metrics, proxy discrimination mitigation approaches, ML algorithms, or local region identification algorithms. We further show that it covers global, individual, and local fairness definitions.

We propose the ***FALCC algorithm*** (short for Fair and Accurate Local Classifications by leveraging Clusters) that conforms to our framework and is the ***first efficient algorithm to make locally fair predictions***, while preserving their accuracy (compared to state-of-the-art methods). FALCC achieves efficiency by precomputing local regions in an offline component and identifying model ensembles well suited (both in terms of fairness and accuracy) for each region. This precomputed information is leveraged when a new sample needs to be classified in an online fashion.

To ensure the ***quality of results*** in terms of accuracy and fairness, we investigate two "tuning" mechanisms: (1) Clearly, the quality of a model ensemble (possibly a different one for each region) depends on the quality of the "pool" of individual models to choose from. We hypothesize and experimentally verify that the more diversified a set of ML models for forming ensembles is, the better suited it is to represent diversity in the regions and thereby foster fairness. Therefore, FALCC incorporates ***diversification*** of ML models. (2) Given that proxy discrimination is a perennial problem in classification tasks, we integrate and evaluate ***inline-processing techniques to counteract the effect of proxy discrimination***. The code of the FALCC framework is available in our repository [1].

Our ***comparative evaluation*** demonstrates that our implementation of FALCC is the first efficient system to offer locally fair, yet accurate classifications. While the results of our in-processing methods to counteract proxy discrimination show room for improvement, our evaluation confirms that our hypothesis that model diversification benefits fairness holds and that our proposed algorithm fares well in generating diverse ensembles.

Structure. We discuss related work in Sec. 2. We introduce our general framework and selected components in Sec. 3. We report on our evaluation in Sec. 4 and finally conclude.

2 RELATED WORK

In recent years, a lot of work has been published regarding fairness in classification problems. We refer to surveys for a general overview [14, 46, 61, 64]. We limit the discussion of related work to approaches most relevant to ours in terms of algorithmic methodology. We summarize these together with FALCC in Tab. 1. We structure our discussion along three aspects: First, we differentiate approaches based on their methodological approach. Second, we highlight supported fairness definitions. Last, we consider selected differentiating applicability or performance features. We end this section by highlighting the positioning of our contribution with respect to these aspects.

Methodological approach. Dynamic model ensemble selection algorithms have been around for over 20 years. The idea is to test the accuracy of trained classifiers on the training data within the local region of a prediction point. Several methods have been proposed to improve the local accuracy of predictions [17, 45, 73]. More recently, methods that leverage model ensembles in the context of fair classifications have emerged. Most of these consider non-dynamic ensembles. For instance, Calders and Verwer [13] train a naive Bayes model separately on the favored and discriminated groups. After the training phase, the method induces fairness by modifying probabilities of the classifiers. Other approaches (e.g., [6, 39]) use and adapt the AdaBoost [70] technique in order to reduce either global [39] or individual [6] bias. Both approaches use fairness metrics for updating sample weights during the training phase. Dwork et al. [25] propose two Decouple algorithms, in which several classifiers are first trained, and then all possible resulting classifier combinations (one classifier per sensitive group) are assessed against a metric which contains an accuracy and a fairness term. The best (global) model combination is then used to classify new samples. Lässig et al. [54, 55] combine the ideas of dynamic model ensembles and fair model ensembles, for which they present several FALCES algorithms. Intuitively, these algorithms evaluate all possible model combinations (one model per group) within the local region of a sample and then choose the most suited model combination for the classification of the sample, thus optimizing local fairness.

We further note that all approaches leveraging model ensembles qualify as in-processing techniques. This contrasts with approaches that reduce proxy discrimination that typically tackle the problem during pre-processing [27, 31, 40, 41, 69, 72]. Some approaches try to reduce proxy discrimination by relabeling [27, 41] or (re-)sampling data and reweighing [40, 72] data points closest to the "decision border". Causal fairness approaches that specifically aim at reducing proxy discrimination include [31, 69]). They use mutual independence tests to determine influences of the protected attribute on other attributes. [43] proposes an in-processing regularizer to remove discrimination. Hereby, mutual information metrics are used to detect indirect discrimination. The post-processing method proposed in [34] reduces the proxy discrimination by minimizing SHAP (Shapley additive explanations) [59] and the MDE (marginal direct effect). While other approaches previously mentioned might also implicitly reduce

[1] https://github.com/NicoLaessig/FALCC

	[17, 45, 73]	[13, 39]	[6]	[25]	[40, 41, 72]	[27, 31, 69]	[43]	[34]	[77]	[16, 50, 51]	[54, 55]	FALCC
Dynamic model ensembles	✓	×	×	×	×	×	×	×	×	×	✓	✓
Fair model ensembles	×	✓	✓	✓	×	×	×	×	×	×	✓	✓
Proxy discr. mitigation	×	×	×	×	✓	✓	✓	✓	×	×	×	✓
Intervention phase	-	in	in	in	pre	pre	in	post	pre	pre	in	in
Global fairness	×	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
Individual fairness	×	×	✓	×	×	×	×	✓	✓	✓	×	✓
Local fairness	×	×	×	×	×	×	×	×	×	×	✓	✓
Non-binary sensitive attributes	×	×	×	✓	×	✓	✓	×	×	✓	✓	✓
Runtime-efficiency	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓

Table 1: Properties of related work and FALCC

proxy discrimination, they neither tackle this issue directly nor consider it in their evaluation.

Fairness notion. Many existing approaches focus on either global [13, 25, 27, 31, 39–41, 43, 69, 72] or individual [6] fairness. In an effort to overcome the gap between these two notions of fairness, recent work (e.g., [16, 34, 50, 51, 77]) proposes to combine concepts from one and the other, or at least accommodate both definitions in a unified way. This includes for instance several approaches. The approach by Zemel et al. [77] transforms the dataset into a similar representation that has the bias removed. Lahoti et al. [50, 51] aim at improving the overall tradeoff between accuracy and fairness by using different approaches, called iFair [50] and Pairwise Fair Representation (short: PFR) [51]. Fair-SMOTE [16] is another recent pre-processing technique that aims at improving global and individual fairness. To the best of our knowledge, there is no in-processing method that considers both global and individual fairness. FALCES [54, 55] qualifies as in-processing method that combines both global fairness and the notion of local fairness.

Applicability and performance features. Given that we aim for a general efficient and effective solution, we also survey existing approaches with respect to the range of classification problems they support and their efficiency. Note that effectiveness is implicitly covered through the methods used by different approaches that we discussed above.

Concerning the applicability, we observe that the minority of surveyed approaches considers non-binary sensitive attributes as possible input [16, 25, 27, 31, 43, 50, 51, 54, 55, 69]. On the upside, most existing fairness algorithms efficiently classify new samples. Note that this runtime-efficiency does not necessarily extend to the training (or offline) phase, as it only has to be conducted once. The only exception to runtime-efficiency for fair classifications is FALCES [54, 55]. This is due to its reliance on dynamic model ensemble methods [17, 45, 73], that determine the local region for each new sample using a k-nearest neighbor (kNN) algorithm [7]. Additionally, several model combinations are assessed within the local region to determine the optimal one. Running these two steps for each new sample is slow.

FALCC. Combining dynamic model ensembles and fair model ensembles aligns well with the definition of local fairness. Therefore, similarly to FALCES, FALCC pursues this approach. However, FALCC's system design and the integration of new algorithms, e.g., to reduce proxy discrimination, make it the first system that supports efficient, yet high-quality classifications, both in terms of accuracy and local fairness. It supports a large variety of classification problems, including problems with multiple non-binary sensitive attributes.

3 FALCC FRAMEWORK AND ALGORITHMS

This section first provides an overview of the FALCC framework, which generalizes a large variety of possible solutions to our overarching problem. We then introduce a running example that we will use in our subsequent detailed discussion of individual components.

3.1 Framework overview

Our framework addresses the problem of locally fair and accurate classifications. The goal is to minimize a loss function $\hat{L}$ that encompasses both accuracy and fairness, for each local region LR. That is, $\forall r \in LR : \min(\hat{L} = \lambda \cdot inaccuracy + (1-\lambda) \cdot bias)$. Here, λ is a weight balancing accuracy and fairness. As a reminder, our goal is to study in-processing solutions that generally support efficient and effective locally fair classifications. We frame our research within the framework depicted in Fig. 2. The framework includes an *offline phase*, where we precompute relevant information to be efficiently accessed and processed during the *online phase*. That is, the offline phase is executed once, based on a *labeled input dataset D*, whereas the online phase is repeated each time a new *test sample t* to be classified arises. D includes one or more sensitive attributes $Sens$.

Offline phase. Starting with the offline phase, FALCC divides D into a training dataset D_{tr} and a validation dataset D_{val}. The first component of the offline phase is *diverse model training*. Its goal is to train a diverse set of classifiers on the training dataset D_{tr}. We hypothesize that a diverse set of classifiers mitigates the potential problem of a set of classifiers comprised in an ensemble being prone to the same problems (relating to fairness or accuracy). Our experiments validate this hypothesis. Note that our framework can in principle accommodate any training to produce classifiers. The training can be based on either the whole dataset D_{tr} or partitions of D_{tr} that commonly reflect different sensitive groups (as [25, 54, 55] show, training on a split dataset may improve accuracy and / or fairness). Given $Sens = \{A_1, \ldots, A_s\}$ and $dom(A_i)$ the domain of sensitive attribute A_i, we have sensitive groups $G = \{(a_1, \ldots, a_s) | a_1 \in dom(A_1), \ldots, a_s \in dom(A_s)\}$. Overall, diverse model training produces a set of models M and creates model combination candidates MC_{cand}, whereas one candidate $MC_{cand_i} = \{\forall_{g_j \in G} : (m_i, g_j) | m_i$ a model $, g_j \subseteq G$ the group on which m_i is applied on.$\}$

The next component in the offline phase offers the opportunity to study in-processing techniques that mitigate proxy discrimination. We suggest first baseline algorithms implementing this component and demonstrate in the evaluation that the framework can incorporate different algorithms.

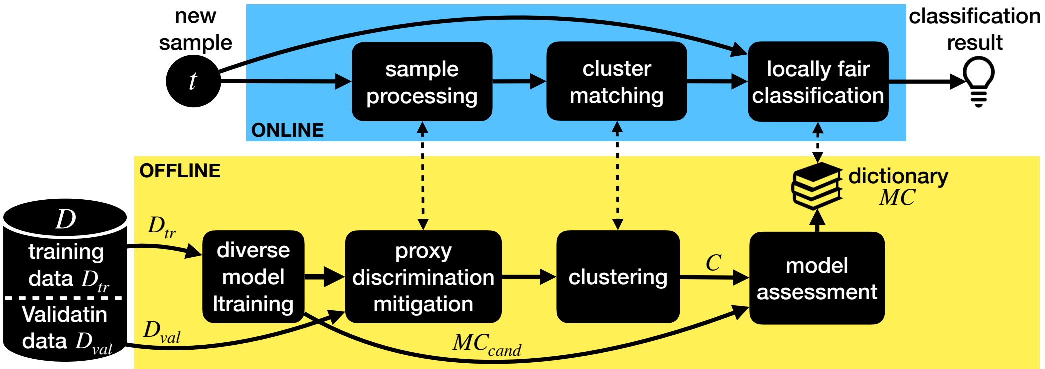


Figure 2: The FALCC framework

Next, *clustering* divides the validation dataset D_{val} into local regions using a clustering algorithm. This is reasonable, as clustering allows to group samples in D_{val} that are highly similar to each other. We denote the clusters determined by this component as $C = \{c_1, \ldots, c_n\}$. The choice of clustering algorithm is flexible. Our implementation relies on a kNN based algorithm with automated parameter estimation.

Finally, the *model assessment* component considers both the clusters in C and the set of models in M to (1) enumerate ensemble candidates for each cluster, (2) assess each combination with respect to a metric, and (3) keep only the best ensemble for each cluster. Hence, for each local region (represented by a cluster), we obtain its "best" model combination. The result of model assessment is a map MC that maps each cluster $c_i \in C$ to a model combination. Each combination contains one model per sensitive group in G.

Online phase. In the online phase, we classify new test samples. Depending on the chosen proxy discrimination mitigation technique, corresponding *sample processing* may be required before a new sample t gets assigned to a corresponding cluster. This is the task of the *cluster matching* component. It matches t to a cluster of points, c_m, that are similar to t. This cluster then represents the local region t belongs to. The *locally fair classification* component simply looks up the best model ensemble for c_m in MC, which is used to classify t.

Bridging the gap between global and individual fairness. By targeting local fairness, FALCC incorporates both local regions (via clustering), common to individual fairness definitions, and metrics for model assessment. These can be metrics either for global or individual fairness. As such, FALCC covers all three fairness notions (global, individual, and local). To achieve global fairness, the number of clusters to be used by the clustering component can simply be set to 1, as this amounts to setting the considered local region to the full dataset. When applying metrics defined for global fairness within the local regions, we effectively implement local fairness. This is an approach similar to FALCES [54], except that in our work, clustering defines local regions during the offline phase (as opposed to computing kNN to t in the online phase). Additionally, FALCC incorporates various fairness metrics, while FALCES concentrates on demographic parity. For individual fairness, the clustering component can be implemented according to the specific local region definitions and the model assessment metric set to an individual fairness metric. To the best of our knowledge, this makes our framework the first to cover the different fairness notions in a unified manner. Our implementation of various algorithms with varying fairness definitions across global, local, and individual in the evaluation showcases this generality.

3.2 Running example

After the overview of our framework, we introduce an example subsequently used to illustrate individual components.

We assume a decision-support scenario about employee raises, given a dataset with multiple attributes about employees, e.g., name, gender, sick leave days, management position *mgt*, department code *dpt* (cf. Tab. 2, ignoring data with grey background). For simplicity, the sole protected attribute gender only has two distinct values (e.g., male and female), yielding two sensitive groups, i.e., g_d (employees prone to discrimination) and g_f (employees typically favored).

Using our approach, we shall see that dedicated models (e.g., m_1, m_2, m_3) are trained for an optimized fairness-accuracy behavior for clusters of similar employees and sensitive groups therein. Eventually, this allows us to classify a new sample t that represents an employee of group g_d using the best model determined for similar employees of the same sensitive group, say m_1. Another employee t' very similar to t but belonging to g_f may then be classified with another model, e.g., m_3. How to reach this behavior will be illustrated as we discuss the individual steps of our approach, starting with diverse model training.

3.3 Diverse model training

Given D_{tr}, diverse model training determines a set of candidate model combinations MC_{cand}. Each model combination is a set of pairs (m_i, g_j) associating a model m_i to a sensitive group $g_j \in G$. To obtain the model combinations, our approach first determines a set of models M. Then, we enumerate all combinations of models $m_i \in M$ with groups g_j that satisfy that m_i has been trained on data comprising g_j.

Given the general nature of our framework, any set of models M can be trained using various techniques and be provided as input. However, to boost fairness, we propose to implement an algorithm that aims at training a diverse set of models. Here, diversity refers to the predictive behavior of the models, which does not necessarily imply the diversity of the type of models. For instance, two trained decision trees are diverse, if their predictions differ from each other. *Boosting* is a commonly used technique to improve the diversity of a model ensemble [47], which iteratively trains estimators based on the outcome of previous iterations. Another strategy is *bagging* [9], in which different subsets of the training data are formed, on which the base estimator is trained on. Commonly used strategies are AdaBoost [70] for boosting and Random Forests [10] for bagging [20, 68].

We implemented the approach described in this section both for AdaBoost and Random Forests. Through preliminary experiments in various settings, we verified previous results showcasing boosting as the more stable approach in inducing diversity [47], compared to bagging. Thus, AdaBoost is set as the default training strategy. In our implementation, we use Decision Tree [32] as the base estimator for the AdaBoost algorithm. Setting the parameters for the AdaBoost (and RandomForest) algorithm properly is important to achieve high diversity in the model ensemble. To achieve this, we apply hyperparameter tuning [74] based on grid search. We conducted several initial experiments to narrow down the search space for grid search, yielding number of estimators $\in \{5, 20\}$, maximum depth of a decision tree $\in \{1, 7\}$, and the splitting criterion for the decision tree $\in$ {gini, entropy}. All other parameters are fixed using the default values given by the scikit-learn package [62]. Among the metrics that can be used to measure the diversity of a model ensemble [48], we opt for the non-pairwise entropy [18].

Above procedure yields a diverse set of classifiers M. Diverse model ensembles are proven to be able to increase accuracy of predictions [47] and we hypothesize (and later validate experimentally) that this also transfers to improving fairness. As a final step, diverse model training enumerates all possible combinations of models per sensitive groups, forming the set of candidate model combinations MC_{cand}.

Example 3.1. In our example, we train the classifiers on the whole training dataset using AdaBoost with hyperparameter tuning. Assuming our approach yields, e.g., three models $M = \{m_1, m_2, m_3\}$. This results in nine candidate model combinations $MC_{cand} = \{\{(m_1, g_d), (m_1, g_f)\}, \{(m_1, g_d), (m_2, g_f)\}, ..., \{(m_3, g_d), (m_3, g_f)\}\}$. For instance, the candidate combination $\{(m_1, g_d), (m_2, g_f)\}$ models the possibility that employees from sensitive group g_d will be classified using m_1, whereas members of g_f will be classified using m_2.

3.4 Proxy discrimination mitigation

The FALCC framework allows to study the mitigation of proxy discrimination during in-processing, as opposed to most of current research (see Sec. 2). The idea underlying proxy discrimination reduction during in-processing within FALCC is to identify and assign weights to (proxy) attributes. Intuitively, the weights are used to "manipulate" the subsequent identification of local regions (the task of the clustering component) such that it is potentially less affected by the presence of proxy attributes and thus less prone to proxy discrimination. While this component is not essential for improving local fairness, we deem it to be an integral part of a general fairness framework, as it is a highly discussed topic in fairness research [61, 65]. The framework is designed to integrate any proxy discrimination technique, as long as it determines a list of proxy attributes (optionally with weights) that translates to an update of the original validation dataset.

Our current framework implementation implements two options to potentially reduce proxy discrimination. Both rely on the Pearson correlation [4] between attributes (more precisely, the correlation between a protected attribute and non-protected ones), which easily applies to both categorical and continuous data, albeit limiting to identifying monotonic relationships. Let $|Sens|$ be the number of sensitive attributes. We measure the pairwise correlation of each individual sensitive attributes $s \in Sens$ to any other attribute $a \in A$. The output is a weight that we apply as reweighing factor for attribute a. The reweighing formula is the following: $\forall a \in A$:

$$weight(a, Sens) = \frac{1}{|Sens|} \sum_{s \in Sens} \left(1 - \frac{\sum_{i=1}^{n} (s_i - \overline{s})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{n} (s_i - \overline{s})^2 \sum_{i=1}^{n} (a_i - \overline{a})^2}} \right) \tag{1}$$

where s_i and a_i are the respective values of the s- and a attributes of sample i, and $\overline{s}$ and $\overline{a}$ are the mean values of the s- and a-variables. It holds that $weight(a, Sens) \in [0, 1]$.

Based on determined weights, our first proxy discrimination mitigation option implements a *reweighing* technique that reweighs the data before applying the clustering approach, thereby distorting the data to be clustered. The clustering algorithm implemented in the next component tries to minimize the sum of squared distance. Attributes with a higher weight will be "spread" along a higher range, so their distances will increase, such that these attributes "declutter" the field and will potentially have a larger impact on cluster separation. Proxy-attributes will exhibit lower weights, so that the distortion will bring data points closer together, favoring samples to be in the same cluster irrespective of their values in these proxy attributes.

The second option completely ignores proxy attributes. Here we also measure the Pearson correlation and if the correlation value is above a given threshold δ with significance level $p > 0.05$, the attribute is removed for the clustering phase. Setting δ is a non-trivial problem, with values ranging from 0.4 to 0.7 in the literature to indicate strong correlation [2]. We choose $\delta = 0.5$ as it indicates a moderate to strong correlation and our initial experiments showed good results. For instance, on the datasets used in the experiments (see Tab. 4), at most two attributes are deemed to have a strong correlation with the protected attributes. Attributes that are not removed keep their original values. A sensitivity analysis for the threshold is left to future work.

Irrespective of the chosen option, the output is an updated validation dataset D'_{val} that is input to clustering. While these strategies act as pre-processing for the clustering algorithm, in the scope of our whole framework they are happening during in-processing. The difference is that the models themselves are trained on the original datasets and that we also let the attributes untouched, when classifying new samples.

Example 3.2. We illustrate the second option that removes proxy attributes, considering the data in Tab. 2 showcasing an excerpt of D_{val}. The correlation of the non-protected attributes (*sickLeave*, *mgt*, *dpt*) to the sensitive attribute (*gender*) are computed. Assuming the remaining data is similar to the sample tuples shown, the attribute *sickLeave* will be flagged as proxy attribute and removed from the updated validation dataset D'_{val} due to its high correlation to *gender*.

3.5 Clustering

After processing the dataset and training classifiers, we now want to define local regions. Hereby, the component clusters the (previously updated) validation dataset D'_{val}, the rationale being to have similar samples within the same cluster, such that clusters are suited stand-ins for local regions. Independently of the proxy discrimination reduction strategy chosen, we do not want the clustering to be dependent on sensitive attributes. Thus,

eid	gender	sickLeave	mgt	dpt	$\cdots$	label	cluster	Pr_{m_1}	Pr_{m_2}	Pr_{m_3}
0	1	0.45	0	033	$\cdots$	?	2	-	-	-
1	1	0.8	1	066	$\cdots$	1	1	0	1	1
2	1	0.75	0	04	$\cdots$	1	2	1	1	0
3	0	0.1	1	07	$\cdots$	1	1	1	0	1
4	0	0.2	0	05	$\cdots$	0	2	0	0	0
5	1	0.9	0	04	$\cdots$	0	2	0	1	0
6	0	0.45	0	095	$\cdots$	0	1	0	0	1
7	0	0	1	01	$\cdots$	1	2	0	0	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 2: D_{val} of our running example. Instances from the favored group g_f are denoted by the protected attribute value 0 and discriminated group g_d are denoted by the value 1.

we project these out and perform clustering on $\Pi_{R\setminus Sens}(D'_{val})$, where R is the set of all attributes in D'_{val}.

Any clustering algorithm can be used in our framework. Our current implementation relies on the well known k-means clustering algorithm [37] with automatic selection of its parameter k. Examples of parameter estimation algorithms to set k are LOGMeans [30], the Elbow method [71], or XMeans [63]. We opt for the LOGMeans algorithm in our implementation as it is both runtime-efficient and does not compromise cluster quality [30]. To faithfully implement global fairness, our implementation also allows to simply set k manually, e.g., to 1.

Once the (automatically) configured clustering is complete, one potential issue that can arise is that a cluster does not contain samples of *all* sensitive groups, thus lack coverage. For example, with our two sensitive groups g_d and g_f and a cluster c_i that only contains samples from g_f, we can only reasonably associate a best model to g_f in the scope of that cluster. However, if a new sample of g_d comes in during the online phase and is assigned to the corresponding cluster c_i, the best choice of a model is unclear. To overcome this issue, our algorithm checks for each cluster c_i if samples of all sensitive groups are present. If not, a nearest neighbors algorithm [7] is used to find reasonably close representatives of any missing group to "fill in the gaps" for a cluster c_i [55]. Unlike in fair ranking research [75, 76], where the representation of the top ranked results should be proportional to the overall data distribution, we only require *any* representation from each group. This allows us to calculate the best model combination within the cluster, considering all groups, while not losing much of the locality. The number of nearest neighbors chosen is fixed in our implementation, estimating this parameter is left to future work.

Example 3.3. Assume that the parameter estimation algorithm for k-means returns $k = 2$. The clusters $C = \{C_1, C_2\}$ subsequently determined on the sample data of D'_{val} shown in Tab. 2 (ignoring the protected attribute *gender* and the proxy attribute *sickLeave*) are identified in the column labeled *cluster*. Note that both clusters include tuples from each sensitive group.

3.6 Model assessment

The last component of the offline phase assesses all model combination candidates MC_{cand} (see Sec. 3) for each local region, as determined by the clusters in C. The result is a dictionary MC that maps each cluster $C_i \in C$ to the best model combination in MC_{cand} for that cluster. This requires a metric to assess the quality of models with respect to some validation data. In general, the assessment should be able to consider both accuracy and fairness. A template for measures considering both is given in Eq. 2. The template essentially is a λ-weighted sum of two parts, one part quantifying *inaccuracy* and the other one *unfairness.* The weight $\lambda \in [0, 1]$ balances the relevance of inaccuracy and unfairness. The inaccuracy part corresponds to the L_1 loss, which calculates the percentage of tuples that have been predicted wrong. The unfairness part (marked with $\ldots$) can be exchanged to reflect different fairness definitions. In general, for each fairness definition, we use a mean difference metric in which we compare each group of a cluster with the average of that cluster. Notation-wise, we denote the number of tuples in a labeled dataset D as $|D|$. Each tuple $i \in D$ has an actual label y_i and predicted label z_i.

$$\hat{L} = \underbrace{\frac{\lambda}{|D|} \sum_{i \in D} |y_i - z_i|}_{\text{Inaccuracy}} + \underbrace{(1 - \lambda) \cdots}_{\text{Unfairness}} \tag{2}$$

Tab. 3 summarizes different metrics known for measuring global bias to define the unfairness part of the equation template. $|G|$ denotes the number of sensitive groups, remaining notation has been introduced before. All listed metrics are integrated in our implementation.

The FALCC framework also allows to accommodate individual fairness metrics, such as consistency [77]. High consistency indicates that kNNs have the same prediction outcome. To exactly model this within our framework actually requires adapting the implementation of the clustering component in addition to considering prediction outcomes during model assessment. Even so, finding the kNN of a new sample t can be quite expensive in terms of runtime. A more efficient, yet approximate solution can leverage clusters either as overestimates or substitutes for kNN. Future avenue is to explore other metrics and how applying them in our setting affect the results.

Once model assessment has computed the quality of each model combination for each cluster, FALCC retains only the best combination for each cluster, i.e., the model combination that minimizes $\hat{L}$.

Example 3.4. For both clusters in C, we assess the nine model combinations in MC_{cand} (see Example 3.1). We opt for demographic parity as fairness metric underlying $\hat{L}$ and set $\lambda = 0.5$. Predictions made by the individual models for the sample tuples of the validation dataset are illustrated in the last three columns with grey background in Tab. 2. Based on the sample data and predictions shown, the algorithm determines that for cluster C_1, m_3 is best suited for both sensitive groups (inaccuracy of $\frac{1}{3}$ and bias of 0) while for cluster C_2, model m_1 (m_3) fits g_d (g_f) best (inaccuracy of 0 and bias of 0). This results in the final model dictionary $MC = \{C_1 : \{(m_3, g_d), (m_3, g_f)\}, C_2 : \{(m_1, g_d), (m_3, g_f)\}\}$.

3.7 Online Phase

As outlined in Sec. 3.1, the online phase consists of three steps: (1) processing of the new sample; (2) assigning the new sample to a cluster; (3) looking up the model to use for classification. It takes as input a new sample t, the clustered validation dataset, the sensitive attributes *Sens*, and MC as computed offline. It outputs a classification result for sample t. Due to a rather straightforward processing, we discuss the three steps directly based on the running example.

Example 3.5. The goal is to classify new employee t (eid=0 in Tab. 2), which belongs to the discriminated group g_d. In Step (1),

Demographic parity [24]	$dp = \frac{1}{\lvert G\rvert} \sum_{j \in G} \left\lvert P(z{=}1 \vert G{=}j) - P(z{=}1) \right\rvert$	Protected and unprotected groups have an equal probability of a positive outcome.
Equalized odds [36]	$eq\_od = \frac{1}{2} \sum_{k=0}^{1} \left(\frac{1}{\lvert G\rvert} \sum_{j \in G} \left\lvert P(z{=}1 \vert G{=}j, y{=}k) - P(z{=}1 \vert y{=}k) \right\rvert \right)$	The probability of a positive outcome for tuples with a real positive label should be equal among all groups, as well as the probability of a positive outcome for tuples with a real negative label.
Equal opportunity [36]	$eq\_op = \frac{1}{\lvert G\rvert} \sum_{j \in G} \left\lvert P(z{=}1 \vert G{=}j, y{=}1) - P(z{=}1 \vert y{=}1) \right\rvert$	Similar to the equalized odds definition, but only takes into account the probability of a positive outcome for tuples with a real positive label.
Treatment equality [5]	$tr\_eq = \frac{1}{\lvert G\rvert} \sum_{j \in G} \left\lvert \frac{FP_{G=j}}{FP_{G=j} + FN_{G=j}} - \frac{FP_{total}}{FP_{total} + FN_{total}} \right\rvert$	Ratio of false positives (FP) to false negatives (FN) is equal among all groups.

Table 3: Metrics (or normalized adaptations thereof) traditionally used for global fairness, integrated in FALCC

we need to process the new sample analogously to the proxy discrimination mitigation technique of the offline phase, e.g., we generate t' by removing the attribute $leaveSick$ of t. In Step (2), we determine the local region t' falls into, i.e., we match t' to the cluster in C whose center is closest to t'. To not misguide the cluster matching based on sensitive attributes, it ignores all attributes of $Sens$, i.e., $gender$. We assume t' matches cluster C_2. Therefore, in Step (3), we retrieve the model combination for C_2 from MC, i.e., $\left\{(m_1, g_d), \left(m_3, g_f\right)\right\}$. Given that t belongs to g_d, the final choice of model to classify this sample is the one associated with g_d in the retrieved model combination, i.e., m_1. The final output is the classification result.

4 EVALUATION

4.1 Setup

4.1.1 Datasets. We evaluate the algorithms both on real world and synthetic datasets. As real-world data, we use several datasets commonly used in experiments of fairness-aware machine learning approaches [56]. The main features of these benchmark datasets are summarized in Tab. 4.

The Communities and Crime Data Set (short: *Communities*) [22, 67] contains data about violent crimes per population within several communities. We use 0.2 per 100k population as threshold for the label. This data has a rich set of features, which are numerical, apart from some information regarding the place of the communities. Few attributes tend to have lots of NULL values. We removed such attributes. Like in other papers [41, 42], we add a binary sensitive attribute Race for which we choose 6% of black population within a community as threshold.

We also use the *Adult Data Set* [21] about persons working in the United States and their salary. The salary is a binary label with the threshold of 50k per year. Due to the data not being numeric, it required some efforts in pre-processing. We use the pre-processed datasets according to the rules presented in [52]. As sensitive attributes, we choose the attributes sex and race. After pre-processing, both sensitive attributes are binary and we obtain 4 sensitive groups.

Another census dataset we use is the US Census Demographic Data [12] (also known as *ACS2017*) from 2017, which tracks the average salary and attribute values of people from different districts. We consider race being the protected attribute and use 6% of black population as threshold.

We also use a *Credit Card Clients* dataset [23], where sex is the protected attribute and payment is the label. Most of the data are already numerical values within the range [0, 1], since they represent several percentages of the district population. Other data, like the amount of males and females living in these districts also have been converted to a percentage number. Very few rows have missing data, which we remove from the dataset. In this dataset, there are several salary attributes. We choose the SalaryPerCapita attribute as label and used 30k as threshold. The other salary attributes are omitted. Like in the Communities dataset, we add the sensitive attribute Race and use 6% of black population as threshold again.

Finally, we also consider the COMPAS Recidivism Racial Bias (short: *COMPAS*) [66] dataset. We use the already pre-processed data which was designed for the FairML framework [1]. Race is the binary sensitive attribute, while 2y Recidivism is the binary label.

We further evaluate the algorithms on two synthetic datasets. Each exhibits one of two biases: *Social* bias (aka direct bias) is the bias resulting solely from the sensitive attribute. For *implicit* bias, the sensitive attribute itself has no direct influence on the overall prediction, but it correlates with several of the other features that do. For both cases, we generate a bias of 30% in mean difference, that is 35%/65% for unfavored/favored group. Both synthetic datasets contain around 14k tuples and 8 features. Since some algorithms do not apply on non-binary sensitive groups, we generate binary sensitive groups for most of the conducted experiments. All datasets are randomly split as follows: 50% for training, 35% for validation, and 15% for prediction.

4.1.2 Algorithms. We compare the performance of FALCC with components implemented as described in the previous section, with the *Decouple* algorithm [25] optimized for global fairness, the "Proposed Ensemble Fair Learning Method" [6] (we refer to it as *FairBoost*) developed to foster individual fairness, the "Learning Fair Representations" (short: *LFR*) [77], the *iFair* [50] and *Fair-SMOTE* [16] algorithms that focus on optimizing both, the *FaX* algorithm [34] that additionally focuses on reducing redlining, and the family of *FALCES* algorithms [54] proposed to obtain local fairness. For the LFR algorithm, we use the implementation provided by the AIF360 framework [3]. Implementations are also available for FaX [33], Fair-SMOTE [15], and iFair [49]. We implemented the other methods based on their description. As the Decouple and FALCES algorithms operate similarly and can be easily adapted to use other metrics, we implemented the metrics described in Sec. 3.6 to be used by these algorithms as well. For algorithms that are potentially affected by the models considered to form ensembles, namely FALCC, FALCES, and Decouple, we consider two alternative configurations. The first uses

dataset	sensitive attr.	# of samples	# of features	$Pr(y = 1 \| s = 1)$	$Pr(y = 1 \| s = 0)$	$Pr(s = 1)$
ACS2017 [12]	race	$72k$	23	49.6%	28.2%	58.8%
Adult Data Set [21]	sex	$46k$	21	31.3%	11.4%	67.6%
Adult Data Set [21]	race	$46k$	21	26.3%	16%	85.7%
Adult Data Set [21]	sex, race	$46k$	21	32.4%	12.3%, 22.6%, 7.6%	59.6%
Communities [22, 67]	race	$2k$	91	19.4%	62.6%	51.4%
COMPAS [66]	race	$6.1k$	7	38.5%	50.2%	40.1%
Credit Card Clients [23]	sex	$30k$	23	20.8%	24.2%	60.4%

Table 4: Metadata about real world datasets, including probabilities wrt group association.

the algorithms "off-the-shelf", i.e., FALCC uses the diversification algorithm for diverse model training (see Sec. 3.3), while Decouple and FALCES train 5 standard classifiers (analogously to the evaluation reported in [54, 55]). The second configuration provides the algorithms with models optimized for fairness to begin with (i.e., LFR, Fair-SMOTE, and FAX in our experiments). When reporting results for the second variant, we add an asterisk to the original algorithm names (i.e., FALCC*, FALCES*, and Decouple*). All classifiers are trained on the whole datasets.

Some of the algorithms require additional parameters. We set the k-value for the k-nearest neighbor algorithm for FALCES to $k = 15$, which is the suggested configuration in [55]. FALCC uses the same value when we have to apply the nearest neighbor algorithm during the clustering step due to missing representatives of a group within a cluster (see Sec. 3.5). Since FairBoost does not consider the k-nearest neighbors per sensitive group, we set $k = 30$ for this approach. This way, the number of kNN considered overall is equal. Due to excessive running time (>24h), we omit the results on the larger datasets for iFair. FALCES comes in four variants and we only report the best result of these algorithms (denoted as *FALCES-BEST*). That is, we choose the result of the variant that exhibits the least local bias. Thus, we consistently compare to the "best" FALCES algorithm, which varies across experiments.

4.1.3 Metrics. We use the metrics described in Sec. 3.6. Accuracy matches the respective term of Eq. 2 (with $accuracy = 1 - inaccuracy$). When reporting global fairness, we take the unfairness part of Eq. 2 and define the (local) region as comprising the full dataset. The local bias directly uses Eq. 2, with $\lambda = 0.5$, thus weighing the accuracy and fairness equally. We report the average local bias over all clusters (=regions), weighted by the sample ratio within the clusters. We choose consistency to assess individual fairness.

4.1.4 Evaluation goals. (EG1) First, we aim at evaluating the overall quality of FALCC. To this end, we perform an extensive comparative evaluation in terms of accuracy, global bias, local bias, and individual bias. (EG2) Our second goal is to validate our hypothesis that a diverse set of classifiers can help improve the overall results in terms of fairness, thereby justifying diverse model training. (EG3) We further study the effect of our redlining compensating strategy. (EG4) Finally, we evaluate the runtime of FALCC. In all experiments, we conduct four runs on different dataset splits (we use the same four randomstates for each algorithm) and report the averages of the runs.

4.2 EG1: Comparative evaluation of result quality

For the set of experiments concerning EG1, we perform a comparative evaluation of all algorithms mentioned above, on all considered datasets. The comparison focuses on accuracy, global fairness, local fairness, and individual fairness. For global and local fairness, we further consider varying fairness metrics, i.e., *demographic parity*, *equalized odds*, and *treatment equality*. We omit experiments with the *equal opportunity* fairness definition, as it is similar to *equalized odds*, and thus similar results are expected (confirmed by initial experiments). For individual fairness, we consider consistency only. Each combination of algorithm, dataset, and fairness-definition is run four times on different dataset splits.

Fig. 3 exemplifies our results on the COMPAS dataset using demographic parity for all "off-the-shelf" algorithms (i.e., without Decouple*, FALCES*, and FALCC*). For global fairness vs. accuracy, the Pareto-optimal solutions are found by LFR (∗), Fair-SMOTE(⊗), Decouple(■), FALCES-BEST(▲), and FaX(+). Yet, these may not all be "good compromises", especially when fairness should not come at the price of degrading accuracy. For instance, LFR exhibits the lowest global bias, but stands far apart from most (slightly) less fair algorithms in terms of accuracy. The same can be seen for the iFair algorithm in the local and individual bias results. Hence, Pareto-optimal solutions are not always the "optimal" choices and therefore we evaluate models using different strategies. Also, depending on which fairness-definition we employ, the algorithms yielding Pareto-optimal solutions vary. For instance, for global fairness, FALCC (●) is not part of the Pareto-optimal solutions, while it is for local and individual bias. To incorporate these two facets in our evaluation, we consider both the membership of an algorithm in the Pareto-optimal solutions and the rank of an algorithm's solution. This ranking relies on the $\hat{L}$ metric in Eq. 2, weighing accuracy and bias equally.

Tab. 5 summarizes the results for all tested configurations. It reports the percentage of configurations in which an algorithm is a Pareto-optimal solution and how often it appears in the top-3 according to the $\hat{L}$-based ranking. We opt for top-3, as we believe that consistently performing well is important. These percentages are reported separately for each bias notion (global, local, individual) and across all tested configurations (All dims). On the left, we report results for all algorithms with their standard configuration. On the right (with grey background), we show results when incorporating models designed for fairness to ensemble-based algorithms.

4.2.1 Default setup. The results show that FALCC performs best when it comes to improving local fairness without compromising accuracy. FALCC is within the top-3 in 89% of the

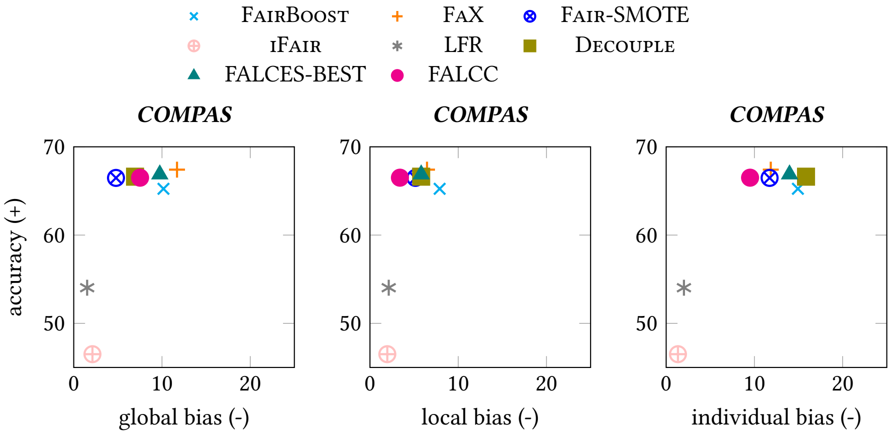


Figure 3: Accuracy-fairness tradeoffs on COMPAS dataset and using demographic parity. Values shown in %.

experiments (this observation still holds when considering the top-1 solution only, albeit with a smaller gap to the runner-up algorithm). This is expected, given that FALCC is designed to improve local bias. Interestingly, FALCC also performs well for global and individual fairness, with FALCC appearing the most frequent in the top-3 for global fairness (tied with Decouple and FaX), and individual fairness (tied with FaX). The Decouple algorithm performs well globally (as expected), but struggles in terms of local and individual results. The FaX algorithm tends to be especially effective in improving the individual fairness-accuracy tradeoff. This can be explained by the SHAP [59] values it uses, as they use *consistency* as one of their properties. Across all conducted experiments (see All dims columns), FALCC appears in ever Pareto-optimal solution. The biggest gap between being part of the Pareto-set and performing well against the $\hat{L}$ metric can be seen by the LFR algorithm. While it is part of most Pareto-sets due to exhibiting low bias, it rarely ends up in the top-3. The drawback is caused by the low accuracy as exemplified in Fig. 3.

The main takeaway so far is that FALCC is the (consistently) best algorithm to improve local fairness without compromising accuracy. Furthermore, it is also effective in improving global fairness and individual fairness. However, keep in mind that so far, the input classifiers of the Decouple, FALCES-BEST, and FALCC algorithms were solely trained for optimizing accuracy.

4.2.2 Using fair classifier as model input. We now also run Decouple*, FALCES-BEST*, and FALCC* that rely on fair classifiers. In the right part of Tab. 5, we see that this can enhance the overall quality of these algorithms' results.

Considering local and individual results, FALCC and FALCC* perform similarly well, with FALCC appearing in more Pareto-optimal solutions. The biggest difference can be seen globally. FALCC* appears in twice as many top-3 solutions compared to FALCC. This is expected, as all fair classifiers used as input are used to induce global fairness. Over all dimensions, both FALCC and FALCC* rank first and second in the Pareto-optimal set appearances and third and first in top-3 ranks, respectively.

The gap between Decouple and FALCC is small in the experiments, when using the fair classifier set as input. This can be explained, as we only take three different models as input, ending up with 9 model combinations (with binary protected groups). A higher variety of input models could highlight the differences between FALCC and the other approaches.

Overall, we can conclude that FALCC consistently outperforms state-of-the-art solutions in terms of local fairness, when accuracy should not be compromised. It also shows good performance for global and individual fairness. The results also show that a non-fairness-induced diverse model ensemble set can be nearly as effective as having fair classifiers as input.

4.3 EG2: Diversification of model ensembles

We introduced the diversification of model ensembles into FALCC to improve the quality of the results, which was confirmed in the experiments reported above. We now study the effect of model diversification in more detail. To this end, we first generate sets of models with a varying degree of diversity. That is, we train several AdaBoost and Random Forest models using different parameter settings.

Fig. 4 plots quality results (accuracy, local bias) of FALCC for each run with varying parameter settings during the model training step. To measure diversity, we use non-pairwise entropy [18], which returns a value between 0 and 1. A higher entropy score indicates a higher diversity within the model ensemble set. The different colors of the scatter plot points depict the density of the plot. Furthermore, a linear regression of the scatter plot is depicted. The figure reports results on accuracy and local bias for three datasets, those not shown exhibit similar trends.

On most datasets, we observe that the bias tends to be lower with higher entropy, i.e., higher diversity in the set of trained models. There are exceptions, e.g. the *social30* dataset, where the bias is generally low and quite steady. While accuracy also degrades with reduced bias, the overall accuracy-fairness tradeoff is getting better with more diverse model ensembles. A high entropy seems to be especially important for datasets where the bias is caused indirectly. In summary, we conclude that our hypothesis holds and that diversity in the set of classifiers can have a positive effect on fair classifications using model ensembles (see Sec. 4.2.1).

4.4 EG3: Effect of proxy discrimination mitigation

In line with EG3, we now apply FALCC approach and vary the strategies designed to mitigate proxy discrimination. More precisely, we consider the two techniques discussed in Sec. 3.4 and

	Global		Local		Individual		All dims.		Global		Local		Individual		All dims.	
algorithm	Pareto	$\hat{L}$	Pareto	$\hat{L}$	Pareto	$\hat{L}$	Pareto	$\hat{L}_{avg}$	Pareto	$\hat{L}$	Pareto	$\hat{L}$	Pareto	$\hat{L}$	Pareto	$\hat{L}_{avg}$
FairBoost	3.7	11.1	3.7	3.7	0.0	0.0	7.4	0.0	3.7	7.4	3.7	0.0	0.0	0.0	7.4	0.0
LFR	**74.1**	11.1	63.0	22.2	**85.2**	18.5	85.2	11.1	**74.1**	3.7	59.3	7.4	**85.2**	18.5	85.2	7.4
iFair	7.4	0.0	7.4	0.0	29.6	11.1	33.3	0.0	3.7	0.0	7.4	0.0	29.6	0.0	33.3	0.0
FaX	**74.1**	**59.3**	85.2	63.0	74.1	**81.5**	85.2	**88.9**	66.7	29.6	77.8	33.3	70.4	44.4	77.8	25.9
Fair-SMOTE	48.1	44.5	44.4	51.9	18.5	40.7	48.1	51.9	29.6	37.0	29.6	33.3	7.4	14.8	40.7	33.3
Decouple	40.7	**59.3**	33.3	25.9	29.6	29.6	44.4	22.2	37.0	37.0	29.6	11.1	29.6	22.2	40.7	14.8
FALCES-BEST	51.8	55.5	51.8	44.4	22.2	37.0	55.5	37.0	37.0	29.6	33.3	11.1	14.8	22.2	44.4	14.8
FALCC	37.0	**59.3**	**96.3**	**88.9**	74.1	**81.5**	**100.0**	**88.9**	29.6	25.9	**96.3**	66.7	74.1	**55.5**	**100.0**	55.5
Decouple-FAIR									66.7	**63.0**	59.3	48.1	40.7	33.3	81.5	59.3
FALCES-FAIR-BEST									55.6	29.6	40.7	33.3	29.6	51.9	70.4	29.6
FALCC-FAIR									40.7	51.9	51.9	**70.4**	33.3	48.1	85.2	**63.0**

Table 5: Summary of the first experiment. Values denote in how many percent of experiments the corresponding algorithm belonged to either the Pareto-optimal set, or to the top-3 when applying $\hat{L}$.

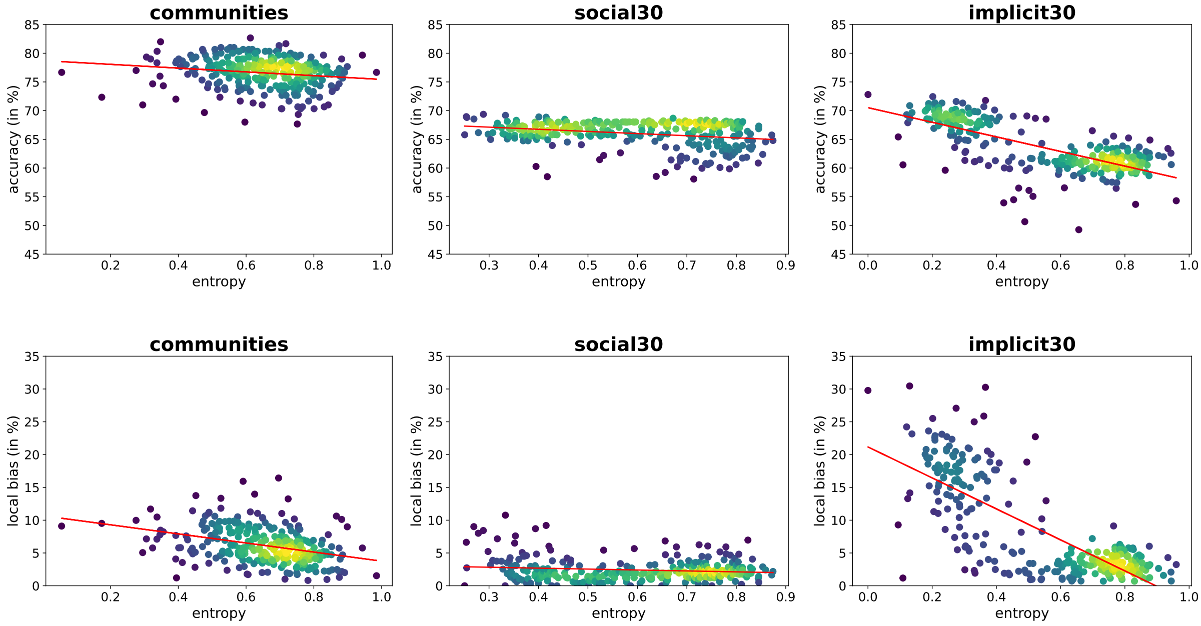
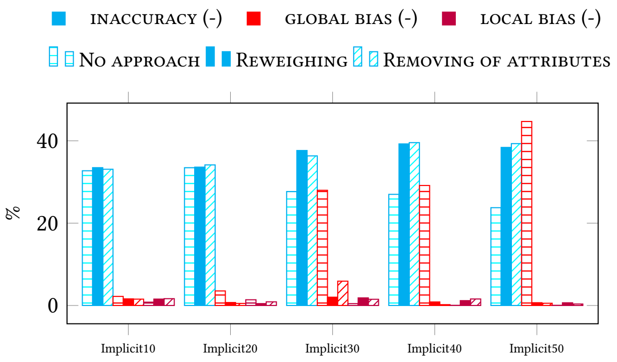


Figure 4: Results of the second set of experiments for varying model ensemble diversity (measured via entropy) with demographic parity chosen as fairness definition



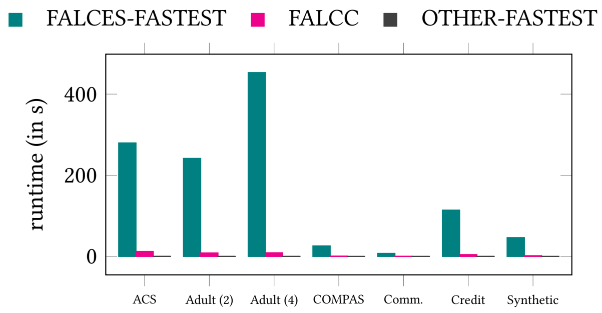


Figure 5: Experiment on varying redlining reduction strategies: (1) No approach; (2) Reweighing; (3) Removal of proxies

Figure 6: Runtime comparison for the online phase of FALCC, FALCES-FASTEST and OTHER-FASTEST.

compare results with running FALCC without any proxy discrimination reduction step. We expect the proxy discrimination mitigation strategies to create more "group-balanced" clusters, which exhibit a distribution similar to the full (global) dataset. This should result in improved global fairness. At the same time, we assume that local bias is less affected by the proxy discrimination mitigation strategies, as FALCC tries to optimize the chosen model combinations within each local region.

Fig. 5 reports the quality results for the different strategies on the *Implicit* dataset, i.e., the artificial dataset generated to include proxy-attributes, and using the *demographic parity* metric. For this experiment, we vary the degree of bias within the dataset (x-axis). The results show that both our proxy discrimination mitigation strategies reduce global bias in cases with moderate to high bias. They prove to be especially effective in settings where the datasets contain high indirect bias. As expected, these techniques do not affect local bias much; it remains quite stable among all strategies. However, applying one of the two proxy discrimination mitigation techniques might result in higher local bias, but to a very small degree. While the global bias overall is reduced, the inaccuracy is increased as well, but to a smaller degree.

On the real-world datasets, due to a less extreme proxy discrimination effect, the positive impact on global bias is less pronounced (but present), while we sometimes do observe a decrease in local bias using the *reweighing* technique. In general, both proxy discrimination mitigation strategies show similar results.

Overall, the hypothesis can be confirmed. While the global bias is improved on datasets containing proxy attributes, it might come at the cost of accuracy. However, the reduction in global bias is more significant than the decrease in accuracy, making the overall tradeoff feasible. A general conclusion cannot be made on *reweighing* compared to *removing*.

4.5 EG4: Runtime-efficiency

For the last set of experiments, we measure the runtime of FALCC and compare it to the fastest algorithm of the FALCES family, as this state-of-the-art approach can perform similarly well on local fairness when using the same classifiers as input (see EG2), and the fastest algorithm of the other algorithms. Note that the fastest algorithm often does not provide the best qualitative results. The runtime of the online phase is depicted in Fig. 6. The number behind "*Adult Data*" indicates the amount of sensitive groups. Clearly, FALCC significantly outperforms FALCES-FASTEST in terms of runtime of the online phase. We further note that while FALCES scales poorly with an increasing number of sensitive groups (e.g., on the *Adult Data* dataset), FALCC scales well. The main reason for the comparably bad runtime of FALCES is that *for each* new sample considered in the online phase, first the kNN [29] have to be computed. Then, all (retained) model combinations must be assessed on these kNN. This strategy has the big downside, that we cannot precompute the kNN in the offline phase for the instances we want to predict, if they are not available yet. In FALCC, local region determination and assessment are done once in the offline phase, only requiring cluster matching and model lookup during online processing. While FALCC is an efficient, fair dynamic model ensemble approach, other algorithms have better runtime efficiency. This is expected, as we must determine to which cluster we assign a new test sample t during the online phase, while the fastest compared algorithm only has to perform the classification.

Putting this result together with our previous results on quality, we conclude FALCC is the first efficient algorithm to tackle local bias effectively without compromising accuracy.

5 SUMMARY AND OUTLOOK

We presented FALCC, a system for efficient locally fair classifications. After an overview of the general framework, we discussed details of the implementation options of its different components. The evaluation validated that the novelties of the system architecture compared to the state-of-the-art make FALCC the new method of choice for locally fair classifications. Indeed, it is highly efficient while maintaining low local bias and high accuracy over a large range of experiments for a variety of fairness definitions. It is also a "one-size-fits-all" solution, since FALCC can often keep up with the performance of the varying best competitor, alleviating users from the tedious choice among multiple algorithms.

In the future, we plan to investigate how to simplify the configuration of FALCC using parameter estimation techniques [53]. This will increase the effectiveness and the usability of FALCC for non-expert users. Initial results when using fair classifiers as input were promising. Thus, we may also investigate several sets of fair classifiers that can be integrated into our framework, potentially in combination with our current training strategy.

ACKNOWLEDGEMENT

Supported by the DFG (German Research Foundation) under Germany's Excellence Strategy – EXC 2120/1 – 390831618.

REFERENCES

[1] Julius A Adebayo et al. 2016. *FairML: ToolBox for diagnosing bias in predictive modeling*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[2] Haldun Akoglu. 2018. User's guide to correlation coefficients. *Turkish journal of emergency medicine* 18, 3 (2018), 91–93.

[3] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. https://arxiv.org/abs/1810.01943

[4] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[5] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2021. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* 50, 1 (2021), 3–44.

[6] Dheeraj Bhaskaruni, Hui Hu, and Chao Lan. 2019. Improving prediction fairness via model ensemble. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 1810–1814.

[7] Nitin Bhatia. 2010. Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security (IJCSIS)* 8, 2 (2010), 4.

[8] Reuben Binns. 2020. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 514–524.

[9] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24 (1996), 123–140.

[10] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[11] Tim Brennan, William Dieterich, and Beate Ehret. 2009. Evaluating the predictive validity of the COMPAS risk and needs assessment system. *Criminal Justice and behavior* 36, 1 (2009), 21–40.

[12] US Census Bureau. 2019. US Census Demographic Data. https://www.kaggle.com.

[13] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

[14] Simon Caton and Christian Haas. 2020. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053* (2020).

[15] Joymallya Chakraborty. 2021. Fair-SMOTE. https://github.com/joymallyac/Fair-SMOTE.

[16] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in machine learning software: Why? how? what to do?. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 429–440.

[17] Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. 2018. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion* 41 (2018), 195–216.

[18] Padraig Cunningham and John Carney. 2000. Diversity versus quality in classification ensembles based on feature selection. In *Machine Learning: ECML 2000: 11th European Conference on Machine Learning Barcelona, Catalonia, Spain, May 31–June 2, 2000 Proceedings 11*. Springer, 109–116.

[19] Jeffrey Dastin. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. Reuters. *Reuters.com* (2018). https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-thatshowed-bias-against-women-idUSKCN1MK08G

[20] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science* 14 (2020), 241–258.

[21] Dheeru Dua and Casey Graff. 2017. Adult Data Set. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/adult

[22] Dheeru Dua and Casey Graff. 2017. Communities and Crime Data Set. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/communities+and+crimel

[23] Dheeru Dua and Casey Graff. 2017. default of credit card clients Data Set. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[24] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[25] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. 2018. Decoupled Classifiers for Group-Fair and Efficient Machine Learning. In *Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research)*, Vol. 81. 119–133.

[26] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. Association for Computing Machinery, New York, NY, USA, 259–268. https://doi.org/10.1145/2783258.2783311

[27] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[28] James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. 2020. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1918–1921.

[29] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3, 3 (1977), 209–226.

[30] Manuel Fritz, Michael Behringer, and Holger Schwarz. 2020. LOG-means: efficiently estimating the number of clusters in large datasets. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2118–2131.

[31] Sainyam Galhotra, Karthikeyan Shanmugam, Prasanna Sattigeri, and Kush R Varshney. 2022. Causal feature selection for algorithmic fairness. In *Proceedings of the 2022 International Conference on Management of Data*. 276–285.

[32] Aurélien Géron. 2019. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media.

[33] Przemyslaw Grabowicz and Nicholas Perello. 2022. Fair and Explainable AI (FaX-AI). https://github.com/social-info-lab/FaX-AI.

[34] Przemyslaw A Grabowicz, Nicholas Perello, and Aarshee Mishra. 2022. Marrying fairness and explainability in supervised learning. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 1905–1916.

[35] Sara Hajian and Josep Domingo-Ferrer. 2013. Direct and indirect discrimination prevention methods. In *Discrimination and privacy in the information society*. Springer, 241–254.

[36] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[37] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[38] Deborah Hellman. 1998. Two Types of Discrimination: The Familiar and the Forgotten. *Calif. L. Rev.* 86 (1998), 315.

[39] Vasileios Iosifidis and Eirini Ntoutsi. 2019. Adafair: Cumulative fairness adaptive boosting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 781–790.

[40] Faisal Kamiran and Toon Calders. 2010. Classification with no discrimination by preferential sampling. In *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*, Vol. 1. Citeseer.

[41] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.

[42] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. 2012. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 924–929.

[43] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 35–50.

[44] Amir E Khandani, Adlar J Kim, and Andrew W Lo. 2010. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance* 34, 11 (2010), 2767–2787.

[45] Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. 2008. From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition* 41, 5 (2008), 1718–1731.

[46] Nikita Kozodoi, Johannes Jacob, and Stefan Lessmann. 2022. Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research* 297, 3 (2022), 1083–1094.

[47] Ludmila I Kuncheva, Marina Skurichina, and Robert PW Duin. 2002. An experimental study on diversity for bagging and boosting with linear classifiers. *Information fusion* 3, 4 (2002), 245–258.

[48] Ludmila I Kuncheva and Christopher J Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning* 51, 2 (2003), 181–207.

[49] Preethi Lahoti. 2020. iFair: Learning Individually Fair Data Representations for Algorithmic Decision Making. https://github.com/plahoti-lgtm/iFair.

[50] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *2019 ieee 35th international conference on data engineering (icde)*. IEEE, 1334–1345.

[51] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439* (2019).

[52] Nico Lässig. 2020. *Entwicklung von fairen und personalisierten Machine Learning Modellen.* Master's thesis.

[53] Nico Lässig. 2023. Towards an AutoML System for Fair Classifications. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 3913–3917.

[54] Nico Lässig, Sarah Oppold, and Melanie Herschel. 2021. Using FALCES against bias in automated decisions by integrating fairness in dynamic model ensembles. *BTW 2021* (2021).

[55] Nico Lässig, Sarah Oppold, and Melanie Herschel. 2022. Metrics and Algorithms for Locally Fair and Accurate Classifications using Ensembles. *Datenbank-Spektrum* (2022), 1–21.

[56] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. 2022. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2022), e1452.

[57] Lan Li, Tina Lassiter, Joohee Oh, and Min Kyung Lee. 2021. Algorithmic hiring in practice: Recruiter and HR Professional's perspectives on AI use in hiring. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 166–176.

[58] Kristian Lum and William Isaac. 2016. To predict and serve? *Significance* 13, 5 (2016), 14–19.

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

[60] Timo Makkonen. 2002. Multiple, compoud and intersectional discrimination: bringing the experiences of the most marginalized to the fore. (2002).

[61] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[62] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[63] Dan Pelleg, Andrew W Moore, et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters.. In *Icml*, Vol. 1. 727–734.

[64] Dana Pessach and Erez Shmueli. 2022. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–44.

[65] Anya ER Prince and Daniel Schwarcz. 2019. Proxy discrimination in the age of artificial intelligence and big data. *Iowa L. Rev.* 105 (2019), 1257.

[66] ProPublica. 2017. COMPAS Recidivism Racial Bias. https://www.kaggle.com.

[67] Michael Redmond and Alok Baveja. 2002. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research* 141, 3 (2002), 660–678.

[68] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[69] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*. 793–810.

[70] Robert E Schapire. 2013. Explaining adaboost. In *Empirical inference*. Springer, 37–52.

[71] Robert L Thorndike. 1953. Who belongs in the family. In *Psychometrika*. Citeseer.

[72] Hao Wang, Berk Ustun, and Flavio Calmon. 2019. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*. PMLR, 6618–6627.

[73] Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. 1997. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 4 (1997), 405–410.

[74] Li Yang and Abdallah Shami. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* 415 (2020), 295–316.

[75] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in ranking, part i: Score-based ranking. *Comput. Surveys* 55, 6 (2022), 1–36.

[76] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2022. Fairness in ranking, part ii: Learning-to-rank and recommender systems. *Comput. Surveys* 55, 6 (2022), 1–41.

[77] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International conference on machine learning*. PMLR, 325–333.