Using FALCES against bias in automated decisions by integrating fairness in dynamic model ensembles

Nico Lässig[1] Sarah Oppold[2] Melanie Herschel[3]

Abstract: As regularly reported in the media, automated classifications and decisions based on machine learning models can cause unfair treatment of certain groups of a general population. Classically, the machine learning models are designed to make highly accurate decisions in general. When one machine learning model is not sufficient to define the possibly complex boundary between classes, multiple "specialized" models are used within a model ensemble to further boost accuracy. In particular, *dynamic model ensembles* pick the most accurate model for each query object, by applying the model that performed best on similar data. Given the labeled data on which models are trained, it is not surprising that any *bias* possibly present in the data will reflect in the classifiers using the models. To mitigate this, recent work has proposed *fair model ensembles*, that instead of focusing (solely) on accuracy also optimize *global fairness*, which is quantified using bias metrics. However, such global fairness that globally minimizes bias may exhibit imbalances in different regions of the data, e.g., caused by some local bias maxima leading to *local unfairness*. In this paper, we propose to bridge the gap between dynamic model ensembles and fair model ensembles and investigate the problem of devising locally fair and accurate dynamic model ensembles, which ultimately optimize for equal opportunity of similar subjects. Our evaluation shows that our approach outperforms the state-of-the-art for different types and degrees of bias present in training data in terms of both local and global fairness, while reaching comparable accuracy.

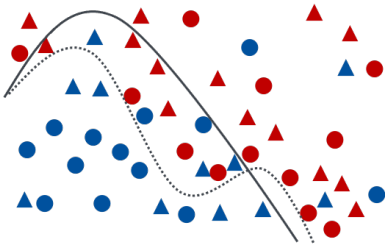
Keywords: Model fairness; bias in machine learning; model ensembles

1 Introduction

In decision support systems (DSS), machine learning models are frequently used to make recommendations or even decisions. While these unquestionably simplify many processes and tasks arising in modern life, critical situations arise in automatic classification scenarios such as credit scoring, or predictive policing applications. There, critical DSS automatically assign people to classes that have the possibility to deeply affect their lives in a positive or negative way. Recent real-life examples where the use of such DSS had to be revoked due to underlying biased classifiers include an algorithm that determined A-Level grades of British students who were unable to take their exams due to COVID-19 regulations [Co20]. Based on the teachers' assessment of the student's performance and the school's performance in subjects, each student was assigned A-Level grades. Using these features, about 40% of

[1] Universität Stuttgart, nico.laessig@ipvs.uni-stuttgart.de
[2] Universität Stuttgart, sarah.oppold@ipvs.uni-stuttgart.de
[3] Universität Stuttgart and University of Singapore, melanie.herschel@ipvs.uni-stuttgart.de

doi:10.18420/btw2021-08



(a) Decision boundary for a classifier using a single model (solid line) and a model ensemble (dotted line).

	male	female
positive class	13	11
negative class	12	14

(b) Training data statistics.

Approach	False positives	False negatives
Single model	7	4
Model ensemble	0	9

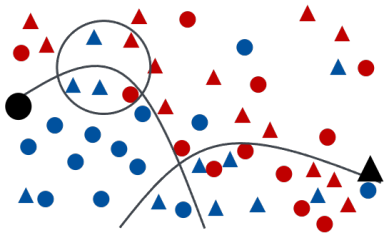
(c) Model performance statistics. Accuracy of single model reaches 0.78, model ensemble reaches 0.82.

Fig. 1: Example binary classification scenario deciding about employee raises.

British students received lower grades than recommended by their teachers, as the model indirectly favored students from private schools and wealthy areas. After a public outcry, the algorithmic decisions were revoked and replaced by the teachers' assessments. Another example is a recruitment tool developed by Amazon [Da18]. The tool was supposed to automatically assign scores to job applicants based on their application to support making hiring decisions. However, it exhibited discrimination against women, a problem that could not be resolved, leading to the project being discarded after several years of investment.

Classification tasks performed by DSS are, by themselves, not trivial to solve. For instance, consider Figure 1, which summarizes a simple classification problem when deciding on employee salary raises. We visualize the training data in Figure 1a, where we place similar employees close to each other and use different shapes to distinguish male (circle) and female (triangle) employees. The goal of the trained classifier is to divide in two classes, which we distinguish by color: employees in blue have a positive outcome and get a raise, while employees labeled in red are associated to the negative class (no raise). Opting for a simple classifier, let us assume we obtain the decision border shown as solid black line in Figure 1a. It classifies all employees below the line into category "blue" and all persons above the line into category "red". Using this simple classifier, a number of people are assigned to the wrong class (see Figure 1c). We see that the simple classifier yields an accuracy of 0.78, computed as the fraction of correctly classified points vs all data points. To obtain a classifier that more faithfully reflects the complex decision boundary in our example and thus improves accuracy, we can resort to *model ensembles*. There, different (simple) models are trained and then combined, e.g., to reach a classifier with the decision border shown as a dashed line in Figure 1a. This allows us to improve the accuracy from 0.78 to 0.82 in our example.

While the above example illustrates one means to boost the accuracy of classifiers, it leaves aside any consideration of *fairness*. The term fairness is often used in the literature to refer to non-discrimination. In the introductory examples, we see that not all students or job applicants were treated equally and some discrimination was unintentionally introduced to the classifiers. Such unfair behavior is commonly linked to some *bias*. There are many



(a) Decision boundary for a fair model ensemble that combines classifiers for male (left) and female employees (right) and illustration of a locally unfair situation (circle).

Approach	Accuracy	Fairness
Single model	0.78	0.76
Model ensemble	0.82	0.66
Fair model ensemble	0.78	0.81

(b) Model performance statistics.

Fig. 2: Example binary classification scenario deciding about employee raises (Figure 1 continued).

different kinds of bias that can be introduced through the data or human decisions. For instance, while it may seem reasonable to consider student's past performance as a feature, e.g., on mock-exams to assign a final grade, wealthy students who benefit from regular tutoring may be at an advantage over students that learn for exams on their own. In case of automatic resume analysis, having a training dataset with CVs predominantly from male applicants possibly causes models that favor terms more commonly used by men than women while penalizing terms associated to women. Returning to our fictional example, based on the numbers reported in Figure 1b, we see that the training data is reasonably balanced in terms of men and women being assigned a positive or negative label, which is a good starting point. To assess the classifiers in terms of fairness, we can use one of many available bias metrics. The American Title VII of the Civil Rights Act of 1964 prohibits employment discrimination and, for example, states that there is discrimination when the probability of a woman getting a positive outcome divided by the probability of a man getting a positive outcome is less than 0.8. In the case of the single classifier and model ensemble, the value is 0.76 and 0.66 respectively (see Figure 2b) and thus below the threshold. So using these classifiers would be against the law in the US.

With metrics quantifying bias being available, recent approaches have considered these to prevent bias. In particular, Dwork et al. [Dw18] have introduced *fair model ensembles*. Given a pre-defined set of sensitive groups (e.g., women), their approach learns classifiers dedicated to these groups and then calculates the best combination of classifiers according to a metric that combines both fairness and accuracy. By training classifiers specialized to different groups, the approach can better capture different patterns exhibited by different groups. Optimizing for both fairness and accuracy compromises between fair treatment of the different groups and model performance. Figure 2 illustrates the effect of using the method for fair model ensembles in our example. It combines two classifiers, for which we show the decision borders: one trained for male employees (left hand side) and one for women (right hand side). Instead of a fairness of 0.66, the positive classification of negatively labeled women by the dedicated classifier raises fairness to 0.81 and therefore (at least according to American law) no longer exhibits discrimination.

While the approach illustrated above comes closer to the goal of treating members of different predefined groups (e.g., male, female) equally, it does so from a global perspective. Thus, localized inequalities remain. For instance, looking again at Figure 2a, while the goal of non-discrimination between women and men is met, the subregion within the depicted circle exhibits local unfairness. As a reminder, we have placed persons in the 2-dimensional representation close to each other when they have similar features, e.g., all persons in the circled region may have similar age and number of sick days. Clearly, despite similar features, women in this region are significantly less likely to get a raise than men. This corresponds to a *local bias*. The approach presented in this paper aims at solving this issue.

The fact that optimization goals are only fulfilled globally and not locally is not a peculiarity of fairness. *Dynamic model ensembles* tackle this problem when optimizing accuracy. Intuitively, a model or model ensemble is dynamically selected for each new decision based on model performance in similar situations. This paper uses a similar rationale to optimize the overall local fairness of a model ensemble by combining ideas of both fair model ensembles and dynamic model ensembles. More precisely, our contributions are as follows:

- We present a framework for addressing the novel problem of mitigating locally unfair decisions. In an offline phase, it trains a diverse set of models to get accurate and fair models for different groups. In an online phase, it dynamically selects the model ensemble best suited for the different groups when focusing on group members similar to the subject to be classified. This combines ideas previously devised for (static) fair model ensembles and dynamic model ensembles specialized on accuracy.
- We present FALCES, which implements our framework using several algorithms. It comes in different variants, depending on whether the training data is further spilt before training classifiers or if trained classifiers are further pruned based on an initial assessment of their global combined performance in terms of fairness and accuracy. It also relies on a novel metric for a combined quantification of fairness and accuracy.
- We implement our algorithm variants and evaluate them on both synthetic and real data. Our results show that while we cannot fully eliminate bias, FALCES outperforms the state-of-the-art in both global group fairness and local group fairness, the latter quantified using a newly defined metric for local fairness. At the same time, our solution does not jeopardizse accuracy.

The remainder of this paper is structured as follows. Section 2 reviews related work. We present our framework in Section 3 and discuss algorithms implementing the framework in Section 4. Our implementation and experimental evaluation are presented in Section 5. The paper concludes with a summary and outlook on future research in Section 6.

2 Related work

Our proposed solution builds on previous work on model ensembles and fairness in machine learning, in particular fair model ensembles and dynamic model ensembles.

2.1 Model ensembles

The idea of model ensembles is to train multiple models and select or combine the best of these models [Po06]. Hereby, the optimization goal typically is the improvement of the accuracy of predictions [SHX19, DSM19, Zh19]. Combining the outputs of several models has proven to be preferable compared to single-model systems. By combining the results of several models, model ensembles can, for example, reduce the risk of choosing a model that performs poorly, which reduces the overall risk of a bad decision, or overcome complex decision borders that may not be able to be implemented by a chosen model because they lie outside the functional space of the model. The same rationale underlies fair model ensembles (discussed further below), which set an additional optimization focus on increasing fairness.

2.2 Fairness in machine learning

As already described in the introduction, the term fairness in machine learning commonly refers to the fact that models must not discriminate against people because of bias(es). Based on various laws, social definitions and understood meanings, different measures to quantify fairness have been defined [KC09, PRT08, Žl17]. They can be broadly classified into two categories. A group of metrics for *individual fairness* (or equality or equality of treatment) focuses on providing equal treatment to equal people [FSV16]. However, we will focus on the second notion of fairness: *group fairness*. It is also known as equality of outcome or equity. Here, groups with different preconditions are treated differently, so that in the end everyone, despite their differences, has the same opportunities. This is intended to overcome social inequalities and offer equal opportunity to different groups [FSV16].

Based on these fairness metrics, methods have been developed which allow the development of individual fair models using fair data, new machine learning algorithms, or techniques for modifying existing models [KC09, PRT08, Ga19].

2.3 Fair model ensembles

While there is now a visible body of research on measuring fairness and learning single fair models, only few works leverage multiple models in order to achieve fairness, thereby bringing the advantages of using model ensembles to the the realm of improving fairness.

Calders and Verwer [CV10] create fair naive Bayes model ensembles. To this end, they split the dataset according to the favored and discriminated groups and learn a naive Bayes model on each subset with the intention to classify independently of a given sensitive attribute. An overall naive Bayes model chooses the decision of either model depending on the group of incoming data tuples to be classified. While this approach yields fairer models, it is specialized to and does not extend beyond naive Bayes models.

Dwork et al. [Dw18] combine multiple machine learning classifiers to improve group fairness. They provide different versions of their algorithm, where the models are either learned on the different subgroups as in [CV10] or on larger data subsets in order to prevent accuracy loss. Their approach uses a joint loss metric that optimizes for both accuracy and fairness in order to assess which model should be used for which group of the dataset. While this approach does consider both accuracy and fairness at group level using any type of classifier, it may suffer from local unfairness.

2.4 Dynamic model ensembles

Dynamic classifier selection [CSC18] selects one classifier during runtime for each new sample which has to be classified. This is based on the rationale of model ensembles that not every classifier is an expert in all local regions of the feature space. Usually, for each new sample the local region is first identified, for example using k-nearest-neighbors (kNN). Then, the quality of available classifiers is determined and the best one(s) are selected. Dynamic ensemble selection is similar, it merely selects ensembles instead of classifiers. One example is the Dynamic Classifier Selection by Local Accuracy (DCS-LA) algorithm by Woods et al. [WKB97]. First, it uses kNN to identify the local region. Then, local accuracy of classifiers is evaluated as percentage of correctly classified samples in the local region. Alternatively, it uses local class accuracy, which is the accuracy of classifiers in the local regions with respect to the class the classifiers assign to the new sample. Only the most accurate classifier is then used to classify the unknown sample.

3 Framework for fair and dynamic model ensembles

As motivated in the introduction, our goal is to combine the benefits of fair model ensembles on the one hand and dynamic model ensembles on the other hand to devise a solution that resolves not only global bias among different groups, but also local bias, while not compromising overall accuracy. The rationale is that, while it is typically possible to define groups that should be treated fairly (and that are often defined by law), it is quite challenging to fully anticipate variations (sub-groups) within these groups that indirectly cause local bias. Techniques to counter local bias can potentially help in reaching equal opportunity among groups with similar features or similar trajectory. In this section, we first formalize our problem statement to counter locally unfair decisions. We then present a framework where we combine the ideas of fair and dynamic model ensembles to solve this problem.

3.1 Locally unfair decisions

As illustrated in Figure 2a, the problem with locally unjust decisions is that while existing solutions (reviewed in Section 2) are optimized to make globally fair and accurate decisions, there are still local regions where data points of different groups are treated unfairly. To address this problem, the decision should be optimized so that the optimal (i.e. fair and accurate) decision can be made at a local level. Our emphasis in this paper is on *group fairness*, i.e., equal opportunities between groups.

Formally, we consider as given a labeled data set D, a similarity metric s, a positive integer k, an optimization metric af combining fairness and accuracy, and a set of machine learning models (classifiers) M for the same classification task. Furthermore, D can be partitioned into groups G, for which equal opportunity is relevant. We further assume a new test sample t that belongs to one of the groups G. Then, we define our goal of *locally fair and accurate classification* as a classification task that classifies t using a model $m \in M$ with the best performance according to the af metric in the local region of D around t. This local region includes the k items in D most similar to t according to s.

3.2 Framework

To address the problem defined above, we combine the rationales underlying both fair and dynamic model ensembles described in Section 2 into a new framework for fair and dynamic model ensembles. The main components of this framework are visualized in Figure 3. We distinguish between an *offline phase* (bottom part), where suited model ensembles are trained and selected, and an *online phase* (upper part), where a previously unseen test sample t is classified by dynamically selecting the model ensemble most appropriate for t.

Offline phase. The first step of the offline phase, named *model training*, is a step common to model ensemble approaches. Here, using training data taken from the labeled dataset D, a diverse set of classifiers is trained. Given that we target both fair and accurate decisions, model training can benefit from using different subsets of data based on the different groups G present in D, which has for instance been proposed for fair model ensembles (see Section 2). We denote the set of classifiers resulting from model training considering different groups G as $M = \{(m_1, g_1), (m_2, g_2), \ldots, (m_n, g_n)\}$, where each m_i is a model and g_i identifies the group it is trained for. Among these classifiers, not all may be suited to make both fair and accurate decisions. Also, too many classifiers to be considered during the online phase (further discussed below) can be computationally prohibitive. Therefore, during *model pruning*, the framework assesses all model combinations or ensembles possible with the classifiers in M that use exactly one classifier per group g_i. Assessment is done with respect to the global performance metric af that considers both accuracy and fairness. Only the best classifier combinations are retained after model pruning, resulting in $MC = \{[((m_{11}, g_1) \ldots, (m_{1n}, g_n)], \ldots, [((m_{e1}, g_1) \ldots, (m_{en}, g_n)]\}$, a set of model ensembles (in square brackets) s.t. for each ensemble, $g_i \neq g_j$ when $i \neq j$ and $n = |G|$.

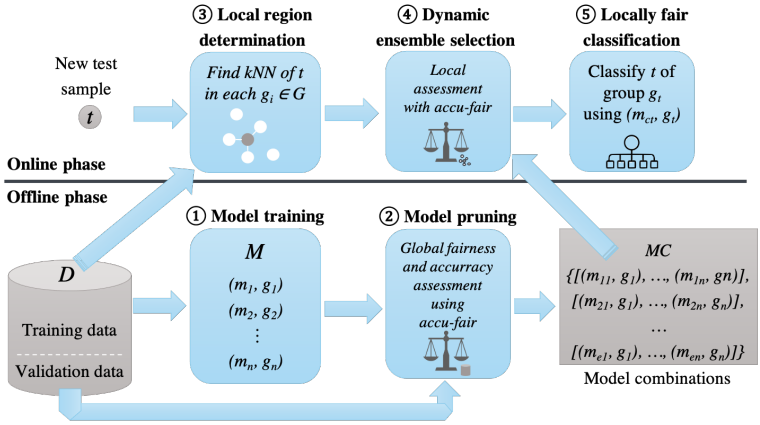


Fig. 3: Framework for locally fair classifications by combining fair and dynamic model ensembles

Online phase. When a new test sample t is to be classified, the framework determines the local region t belongs to as part of *local region determination*. To this end, it performs a kNN search of t on each g_i, where g_i is a group in $G = \{g_i, \ldots, g_n\}$. The framework specifically selects an equal number of members similar to t of each group, to have a locally balanced data region with respect to the different groups. Then, for this particular region, *dynamic ensemble selection* assesses which ensemble $E \in MC$ achieves the best local performance with respect to af. Intuitively, this dynamically selects the optimal model ensemble comprising a dedicated model for each group for the region most relevant to t. With this approach, our framework combines previous techniques separately devised for fair and dynamic model ensembles. The identification of the locally best model is performed according to dynamic model ensemble techniques using fair model ensemble metrics. Therefore the classifiers are tested on the local region of the test sample using an af metric. Finally, the best classifier m_{ct} such that $(m_{ct}, g_t) \in E$ and g_t corresponds to the group t belongs to is used in the final step of *locally fair classification* to classify t.

4 Algorithms implementing the framework

Section 3 discussed the general framework that we propose for locally fair and accurate classifications. There are a variety of techniques from both fair and dynamic model ensemble research, which can be applied or extended to implement its components. In the following, we discuss the algorithms we consider to implement the framework that stand behind our FALCES system (standing for Fair and Accurate Local Classifications using EnsembleS).

4.1 Model training

As mentioned before, the set of classifiers should be diverse in order to benefit from combining them to model ensembles. To this end, we vary both the set of machine learning techniques used to train classifiers as well as the data from D that is considered for training.

In principle, any machine learning technique suited for classification tasks can be considered as candidate technique. In our evaluation, we will resort to simple techniques, e.g., decision trees, logistic regression, or nonlinear support vector machines.

Concerning the data, following previous research on fair model ensembles [Dw18, CV10], we consider splitting the input dataset D on pre-defined sub-datasets that correspond to the different groups for which we aim to achieve group fairness (e.g., we divide by sex (male, female) and race (white, others) in our experiments which creates four subgroups). This effectively partitions D into $G = \{g_1, \ldots, g_n\}$, assuming n distinct groups. Then, models are trained separately for the different partitions. Different training datasets not only have the advantage of learning different models that exhibit their strengths in certain areas of the feature space. Indeed, as shown in Calders et al [CV10], the label does not depend directly on the sensitive group. In addition, complex decision borders between the two groups, which originate from different behavioral patterns, can be better modeled, thus increasing accuracy [Dw18].

As a result, similar to [Dw18, CV10], we obtain classifiers that are "specialized" on some group. More precisely, in this variant, we obtain $M = \{(m_1, g_1), \ldots, (m_k, g_n)\}$ where each (m_i, g_j) associates a classifier m_i to a group g_j. For any two $(m_i, g_j), (m_{i'}, g_{j'})$, it holds that $m_i \neq m_{i'}$, and $g_j, g_{j'} \in G$, but it is possible that $g_j = g_{j'}$.

Example 1. *As a simple example, consider we spilt a sample dataset following the gender of employees, which results in a group for each gender, e.g., g_F for female employees and g_M for male employees. Assuming m_1, m_2, m_3 are trained on g_M and m_4, m_5, m_6 on g_F, we obtain $M = \{(m_1, g_M), (m_2, g_M), (m_3, g_M), (m_4, g_F), (m_5, g_F), (m_6, g_F)\}$.*

On the downside, splitting the data as described above can lead to a too small dataset to train on, which often results in loss of accuracy for the classifiers. Hence, we further consider the option of training models on the complete dataset D rather than on individual partitions of G. In this setting, we have $M = \{(m_1, g_D), \ldots (m_n, g_D)\}$, where $g_D = D$.

To distinguish the two variants for implementing model training described above in FALCES, we will append a suffix SBT (for Split Before Training) for the first option, while absence of this suffix indicates training is performed on the full training data set.

4.2 Model pruning

In the offline phase, the number of classifiers can already be reduced based on their global performance in order to improve the efficiency of the online phase later. Indeed, the less classifiers need to be considered in the online phase, the faster the classification of a new test item is. As we shall see in the evaluation (Section 5.4), this has only little impact on making locally fair and accurate decisions, while runtime may improve significantly.

To assess the performance of a model when considering both accuracy and fairness, we rely on a metric that combines these two dimensions, denoted as af. To the best of our knowledge, the state-of-the art metric that accounts for both accuracy and fairness is the metric proposed by Dwork et al. [Dw18] for fair model ensembles, defined as follows.

$$\hat{L} = \underbrace{\frac{\lambda}{|D|} \sum_{t_i \in D} |y_i - z_i|}_{\text{Inaccuracy}} + \underbrace{\frac{1-\lambda}{|D|} \sum_{g_j \in G} \left| \sum_{\substack{t_i \in D: \\ g_{t_i} = g_j}} z_i - \frac{1}{|G|} \sum_{t_i \in D} z_i \right|}_{\text{Unfairness}} \tag{1}$$

Here, the number of tuples in a labeled dataset D is denoted as $|D|$, each tuple $t_i \in D$ has an actual and predicted label denoted as y_i and z_i respectively, $|G|$ represents the number of different groups in G, $g_{t_i} \in G$ represents the group a tuple t_i belongs to, and $0 \leq \lambda \leq 1$ balances the relative weight of the accuracy and the fairness part of the equation. Intuitively, in the first part of the metric, accuracy is measured by comparing the predicted and actual label for each data tuple (also known as L_1 loss). The second part of the metric determines the fairness of the classifier combination that associates a classifier to each group. It sums up the difference between the sum of predicted values of each group and the overall sum of labels divided by number of groups. Note that for both sides, higher values actually mean a poorer performance, we thus qualify them as inaccuracy and unfairness.

This metric combines both accuracy and fairness, however, the fairness-part is sensitive to differences in the relative size of groups. Assume for instance there is a larger group g_L and a smaller group g_S with equal sum of z_i, i.e. equal number of positively classified tuples. Indeed, the metric considers the situation to be fair among these two groups (unfairness-part drops to 0), even though the probability that a member of g_L is assigned a positive label is smaller than the probability of a member of g_S being assigned a positive label. While this may well serve minorities that are considered protected groups and are thus indirectly favored by being part of the smaller group, it does not accurately reflect equal opportunity.

We propose a variation of af that modifies the fairness-related part of Equation 1 to also consider the number of tuples $|g_j|$ in a group $g_j \in G$. This results in the following metric.

$$\hat{L}_{new} = \frac{\lambda}{|D|} \sum_{t_i \in D} |y_i - z_i| + \frac{1-\lambda}{|G|} \sum_{g_j \in G} \left| \sum_{\substack{t_i \in D: \\ g_{t_i} = g_j}} \frac{z_i}{|g_j|} - \frac{1}{|D|} \sum_{t_i \in D} z_i \right| \tag{2}$$

While the accuracy part still determines L_1 loss, the fairness part has slightly changed. For each group, its mean value is set in relation to its group size and compared to the overall mean value of positive predicted labels. These are then again summed up and divided by the number of groups to allow for an arbitrary number of groups.

Using the metric proposed in Equation 2, model pruning aims at retaining only a "good" selection of ensembles formed by models of M obtained during model training. Given that we are using model ensembles, this evaluation of model quality is performed by considering all possible combinations of classifiers in M when choosing one classifier per group, and keeping only the best ones. In our implementation, we keep ensembles up to a predefined rank. Another possibility would be to use a threshold for the maximally acceptable *af* score.

Example 2. *Continuing our previous example, given that we have three classifiers dedicated to g_F and g_M, respectively, we have a total of 9 combinations to test using af. Let us assume that the top-2 ensembles according to af are $(m_1, m_4), (m_2, m_5)$. Assuming FALCES is configured to the top-2 combinations, we obtain $MC = \{[(m_1, g_M), (m_4, g_F)], [(m_2, g_M), (m_5, g_F)]\}$.*

Similarly to model pruning, we consider running FALCES with or without model pruning enabled. When active, we append PFA to the algorithm name.

4.3 Local region determination

Moving on to the online phase, the task is to classify a new tuple t in a locally accurate and fair manner. Our framework defines locality relying on a similarity measure s and considers retrieving the k most similar tuples to t in D.

One way to determine the k most similar tuples are kNN algorithms [Bh10]. FALCES uses the kd-tree nearest neighbor approach [Be75] because it is simple and efficient. This method creates a k-dimensional tree that can be precomputed during the offline phase in which the tuples from D are arranged and stored according to the dimensions. During the online phase, when the tuple t is to be classified, the tree can then be searched in $O(log|D|)$ time.

While searching for the nearest neighbors, we need a similarity metric to identify tuples similar to t. To compare two tuples $t_1 = (x_1, ..., x_n)$ and $t_2 = (y_1, ...y_n)$, FALCES uses the Minkowski-Distance $md(t_1, t_2) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$. It is a generalization of both the Manhattan distance (p=1) and the Euclidean distance (p=2) and has already proven useful for similar problems such as K-Means algorithms [SYR13].

Using this distance measure, we identify the nearest neighbors of t. However, it must be ensured at this step already that the *af* metric used in the next step of FALCES receives the necessary information to calculate group fairness. For this, it needs to receive tuples from all groups to be able to produce meaningful results. Therefore, in FALCES, the kNN algorithm is applied to each group separately, which results in $|G|$ trees and $|G| * k$ nearest neighbors for $|G|$ groups.

4.4 Dynamic ensemble selection

Based on the $|G| * k$ tuples defining the local region for a given test sample t, dynamic ensemble selection identifies the ensemble $E = [(m_{c_1}, g_1), \ldots, (m_{c_p}, g_p)] \in MC$ that achieves the best local performance. To this end, FALCES follows previous research on dynamic model ensembles [WKB97] and combines these techniques with the *af* metric. More precisely, using as input MC, we evaluate all model combinations based on *af* when they classify the $|G| * k$ nearest neighbors of t. The combination E with the lowest *af*-score is retained.

Example 3. *Assume we want to classify a male employee t that is thus considered to be part of* g_M*. Assuming* $k = 20$*, kNN retrieves the 20 male and 20 female samples in D most similar to t. The two combinations possible with the classifiers retained after model pruning (see Example 2), i.e.,* $[(m_1, g_M), (m_4, g_F)]$ *and* $[(m_2, g_M), (m_5, g_F)]$*, are evaluated using the af metric and focusing on the 40 samples of D that form the local region. In our example, let this result in* $E = [(m_1, g_M), (m_4, g_F)]$ *as this combination reaches the lowest score.*

Note that through previous model pruning during the offline phase, the above example needed only to consider 2 instead of 25 classifier combinations. In addition to reducing the number of comparisons, we further reduce the complexity of an individual combination assessment, because the computation of classification predictions for all sets of classifiers and all local $|G| * k$ tuples can be quite time consuming. That is, FALCES precomputes all classification predictions for all tuples in D using all models in M. This allows dynamic ensemble selection to simply look up the necessary predictions instead of repeatedly computing them by applying the classifier for each test sample during the online phase.

4.5 Locally fair classification

Finally, the classifier of the previously identified model ensemble E that achieved best local performance with respect to the *af* metric and that is associated to the same group as t is used to classify t.

Example 4. *Continuing our running example with* $E = [(m_1, g_M), (m_4, g_F)]$*,* m_1 *is finally used to classify t, because t belongs to* g_M*. Considering a different* t' *of group* g_M *may*

result in a different local region, where for instance $E = [(m_2, g_M), (m_5, g_F)]$ *performs better, resulting in the use of* m_2 *instead.*

5 Evaluation

We have implemented the algorithms discussed in Section 4 and present their evaluation in this section. We first describe our experimental setup. We then present and discuss results on combined accuracy and fairness, differences observed for the two *af* metrics, as well as runtime results on the online phase.

5.1 Experimental setup

This section summarizes which different algorithm variants and baseline solutions we compare in our evaluation. We further discuss datasets and metrics we use for benchmarking.

Compared algorithms. We have presented different variants of FALCES, depending on whether or not the training data is split before training and whether or not model pruning is applied. In addition, we compare to the state-of-the-art algorithms. More precisely, we consider the algorithms summarized in Table 1. For the different FALCES variants as well as DCS-LA, which also relies on kNN search, we set $k = 15$.

Datasets and ML models. We use both synthetic and real benchmark data in our evaluation.

We developed a synthetic data generator in order to control different types and degrees of bias in order to study how the different algorithms are affected by these. It generates labeled data for two groups and a binary classification task and allows to control (i) the *group*

Algorithm	Description
FALCES	Our baseline algorithm without splitting before training and without model pruning.
FALCES-SBT	This variant of FALCES splits the dataset for training but does not apply model pruning.
FALCES-PFA	In this variant, model pruning is applied on models trained over the complete training dataset.
FALCES-SBT-PFA	The offline phase performs model pruning on models that have been trained on sub-sets of the training dataset, which has been split according to considered groups.
DCS-LA [WKB97]	A baseline algorithm for dynamic model ensembles that optimizes accuracy, which we extended for FALCES.
Decouple [Dw18]	State-of-the-art algorithm for fair model ensembles, when models are trained using the full training dataset.
Decouple-SBT [Dw18]	Variant of Decouple that trains models on a previously split training dataset.

Tab. 1: Overview of the algorithms compared in our evaluation

Bias type	Parameter settings
Group balance	0.1, 0.2, 0.3, 0.4, **0.5**
Social bias	**0**, 0.1, 0.2, 0.3, 0.4, 0.5
Implicit bias	**0**, 0.1, 0.2, 0.3, 0.4, 0.5

Tab. 2: Different configurations for synthetic data, default values in bold

balance, (ii) the degree of *social bias*, and (iii) *implicit bias*. Group balance describes the percentage group g_1 represents in the full dataset (g_2 implicitly making up the remainder of the dataset), which can be very unbalanced (e.g., only 10% of the training data belongs to g_1) or perfectly balanced at 50%. Social bias refers to bias directly related to the protected attribute defining a group (e.g., gender in our example), reflected by different probabilities for a positive label in the different groups (e.g., women have a lower probability for a positive label than men). Such bias is sometimes also called historical bias, because it reflects direct discrimination of a group in a dataset that commonly labels data based on historical decisions. In our experiments, a social bias of 0 means probabilities are equal for both groups (no discrimination), 0.1 if the probability for g_1 differs by 0.1, and so on. Implicit bias is present in a dataset when, even though groups are not directly discriminated, their label depends on an unfavorable attribute value that occurs more frequently in the protected group, i.e., that is correlated to the protected group. Note that both examples from the press mentioned in the introduction are likely linked to such indirect bias. Similarly to social bias, we vary indirect bias from 0 (none) in increments of 0.1. The generated data in all cases consists of approximately 13,000 tuples. Table 2 summarizes the configurations we used for testing. When not mentioned otherwise, the values are set to the default values highlighted in bold.

We chose the Adult Data Set [DG19], a census income dataset with data from 1994, which is a commonly used dataset in multiple machine learning experiments. This dataset consists of approximately 49,000 tuples and contains various variables, including a binary salary value of yearly income with the threshold of 50K$, which is our label in the experiments. We chose the attribute "sex" with values "male" and "female" as a sensitive attribute, as well as a combination of the attribute "sex" with the attribute "race" with values "white" and "others", where we grouped together all other races, because all other races make up $\approx$ 10% of the dataset.

Each dataset (synthetic and real) is split randomly such that 50% of the dataset serve as training data for model training, 35% for validation to determine emsembles, and 15% for testing the quality of predictions in the online phase.

To get a diverse set of classifiers, we train five different classifiers on our datasets: (i) Decision Tree, (ii) Logistic Regression, (iii) Softmax Regression, (iv) Linear Support Vector Machine, and (v) Nonlinear Support Vector Machine. Given that we have two groups, this results in ten classifiers when we split before training, and five when training on the full dataset.

Metrics. Given that we aim for a good compromise of accuracy and fairness, we use metrics to assess the different algorithms in these two dimensions. We use the well known accuracy-metric commonly used to evaluate machine learning techniques. For fairness, we distinguish between global and local fairness. To study global fairness, we use the "unfairness part" of the metric given by Equation 2 (setting $\lambda = 0$), to which we refer to as global bias (lower values are better). To measure local bias, we define a local region bias metric, which we call *local region discrimination (LRD)*:

$$LRD = \frac{1}{|G| \cdot |D|} \sum_{t_i \in D} \sum_{g_j \in G} \left| \frac{1}{k} \sum_{z_l \in R_{t_i,g_j}} z_l - \frac{1}{k|G|} \sum_{g_m \in G} \sum_{z_l \in R_{t_i,g_m}} z_l \right| \tag{3}$$

where R_{t_i,g_j} is the local data region of t_i comprising the kNN of t_i in group g_j. In this metric the probability of a positive predicted label of each group in the local region is measured against the average probability of a positive predicted label amongst all points in the local region. In this way, the metric reflects the average local fairness.

Using the experimental setup described in this section, we now discuss results we obtained.

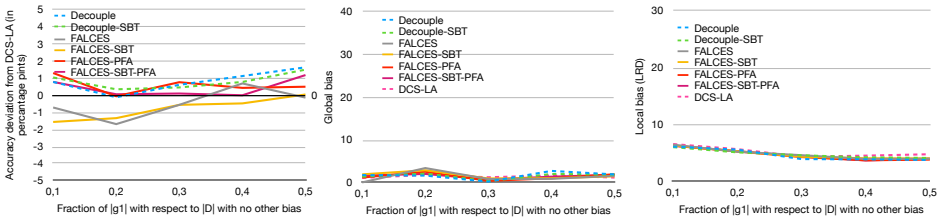
5.2 Comparative evaluation in terms of accuracy and fairness

We first present results we obtained when using different algorithms on our synthetic dataset in terms of accuracy, global bias, and local bias.

As a first baseline, we start with a "clean" dataset with no social or implicit bias, and see if changes in group balance have an impact on our three metrics. Essentially, we expect only a marginal effect on accuracy and a low global and local bias, because the input data is a priori unbiased. This is confirmed by the results depicted in Figure 4. Note that instead of plotting absolute accuracy for all methods, we plot the deviation algorithms have in accuracy from the accuracy reached by DCS-LA, reported as percentage points. DCS-LA is not considering bias and optimizes solely for accuracy, which is between 0.76 and 0.79 for DCS-LA over the whole range of considered group balance. The ordinate reporting percentage points, a deviation of -1 means that an algorithm reaches for instance 0.77 instead of 0.78 reached by DCS-LA.

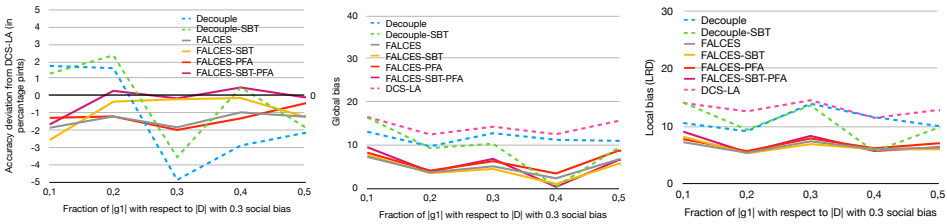
In Figure 4, we observe that all algorithms perform similarly, i.e., for all algorithms, there is some very small fluctuation in accuracy and global bias remains low. For local bias, while being generally low as well, we observe that it steadily increases with increasing imbalance, reaching a relative increase of up to 64% from the balanced case (0.5) to the highest imbalance (at 0.1, where only 10% of the dataset concern one group).

Next, we perform the same analysis again, but this time with an additional social bias of 0.3 introduced to g_1. The results are summarized in Figure 5. With the introduction of social bias, we observe that deviations in accuracy become more pronounced, in particular for the two variants of the Decouple algorithm. Least affected in terms of accuracy is FALCES-SBT-PFA, actually having comparable or better accuracy than DCS-LA. For both local and global bias, we see that all FALCES variants consistently outperform both Decouple variants and DCS-LA. Also, compared to the previous experiment without social bias, the field has overall shifted upwards. This shows that we cannot fully counter bias originally present in the dataset, but FALCES is best in reducing it while maintaining high accuracy.



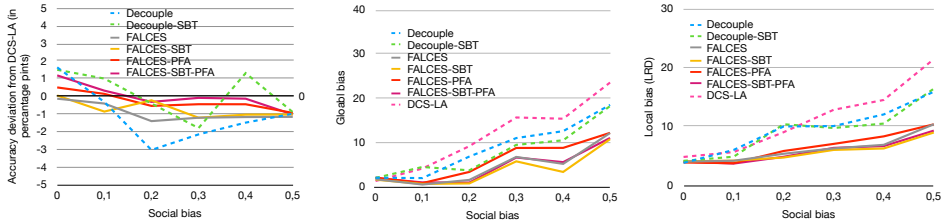
(a) Accuracy deviation from DCS-LA (b) Global bias (c) Local bias

Fig. 4: Results on synthetic data with varying group balance, no social bias, and no implicit bias.



(a) Accuracy deviation from DCS-LA (b) Global bias (c) Local bias

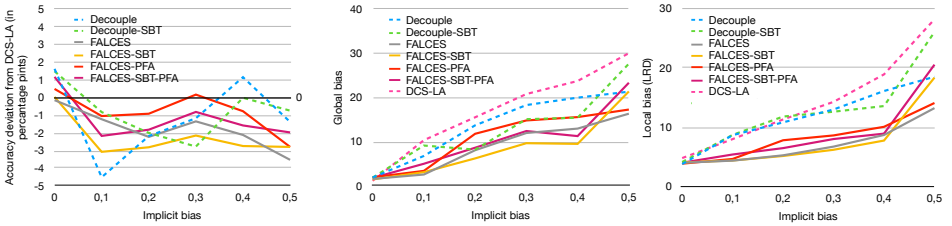
Fig. 5: Results on synthetic data with varying group balance, 0.3 social bias, and no implicit bias.



(a) Accuracy deviation from DCS-LA (b) Global bias (c) Local bias

Fig. 6: Results on synthetic data with varying social bias, group balance of 0.5, and no implicit bias.

Our next analysis focuses on the impact different degrees of social bias have on the overall performance, assuming balanced groups without additional implicit bias. Figure 6 reports our results. For accuracy, we observe that all methods fluctuate, but the degradation in accuracy (typically less than 2 percentage points) is tolerable. Our approaches are more robust to social bias than the state-of-the-art Decouple variants, the PFA variants generally being closest to the accuracy reached by DCS-LA. For both local and global bias, a clear upward trend is visible with increasing social bias, showing that the more bias in the input data, the more bias the ensemble generates. However, the gradient of our approaches is less steep and consistently below the baseline methods. This means that the more social bias in the data, the more effective our approaches are in countering the bias to optimize (local) group fairness compared to the state-of-the-art. FALCES-SBT is best in terms of global



(a) Accuracy deviation from DCS-LA (b) Global bias (c) Local bias

Fig. 7: Results on synthetic data with varying implicit bias, group balance of 0.5, and no social bias.

and local bias, but FALCES-SBT-PFA has similar performance and thus presents a good compromise in datasets with mainly social bias.

We perform a similar analysis for implicit bias in the source data, again assuming a balance of groups (balance = 0.5) and setting social bias to 0. Figure 7 visualizes the results of this set of experiments. Our first observation is that implicit bias impacts all metrics more than the previously considered social bias. As before, in terms of variations in accuracy, these are strongest for the Decouple variants, whereas the PFA variants of our algorithm outperform FALCES and FALCES-SBT. However, looking at both local and global bias, our algorithms without model pruning typically perform better than their counterpart with PFA. The reason for this is that model pruning during the offline phase can prune classifiers that would, during the online phase, be better compared to those retained after model pruning. Nevertheless, in general, our methods outperform the state of the art for a wide range of implicit bias configurations.

We validate our findings on artificial data on the real-world dataset as well. Given that it includes two sensitive attributes (sex and race), we study accuracy, global bias, and local bias when just one attribute is used to form groups (resulting in two groups) and when two attributes are used (resulting in 4 groups). Figure 8 shows results for global and local bias. Results on accuracy confirm that all algorithms perform similarly, it consistently ranges between 0.790 (Decouple) and 0.799 (DCS-LA). As before, we observe that FALCES variants typically are comparable or outperform the three baseline algorithms, both in terms of global and local fairness. With the increasing number of sensitive attributes, we observe that the bias increases for all methods.

In conclusion, we see that our methods improve on the state of the art by offering a better accuracy-fairness compromise than the state of the art Decouple approach (considering global fairness) and the difference in accuracy compared to DCS-LA is typically tolerable. Our methods are also the most robust to different types and degrees of bias (we studied group balance, social bias, and implicit bias). An added benefit is that our methods inherently consider local fairness as well, and our evaluation of local fairness shows that the classifications performed using our algorithms get us closer to equal opportunity for different predefined groups.

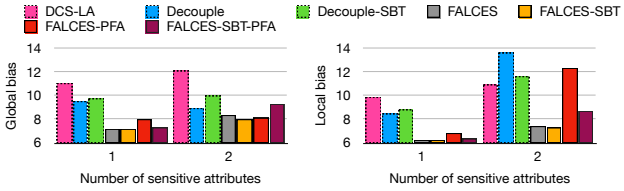
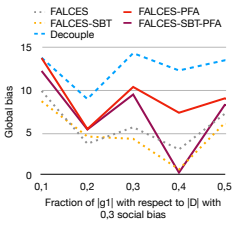


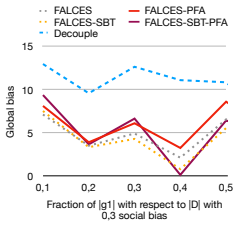
Fig. 8: Global bias (left) and local bias (right) on real-world dataset.

|G|	FALCES	FALCES-PFA
2	0.58	0.48
4	7.98	1.63

Fig. 9: Average classification time (s) on real-world dataset

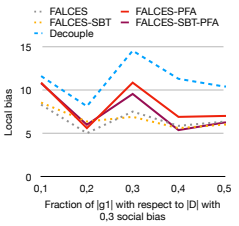


(a) SOA

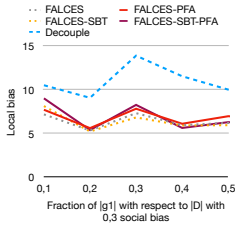


(b) NEW

Fig. 10: Global bias for varying group balance, 0.3 social bias, and using alternative *af*-metrics



(a) SOA



(b) NEW

Fig. 11: Local bias for varying group balance, 0.3 social bias, and alternative *af*-metrics

5.3 Impact of different accuracy-fairness metrics

Section 4.2 has discussed two alternative metrics for *af*, used both during model pruning in the offline phase and dynamic classifier selection in the online phase. All experiments so far have used our extended metric (Equation 2). The next series of experiments investigates how the two options potentially impact the result. We refer to the state-of-the-art metric of Equation 1 as SOA, while our extended metric is labeled NEW. As a reminder, our extension aims at countering the effect on fairness in presence of unbalanced groups. Therefore, we focus our study on evaluating both the global and local bias for different configurations of group balance. As before, accuracy is comparable across all approaches, whether we use SOA or NEW. Figure 10 reports our results on global bias, whereas Figure 11 focuses on local bias. For better readability, we omit the results of Decouple-SBT and DCS-LA, their relative performance to the other approaches being analogous to our previous discussion.

For both global bias and local bias, we see that FALCES variants without model pruning (dotted lines) are comparable when using SOA or NEW. The effect of using a different metric only becomes apparent when model pruning is active. Overall, we see that NEW closes the "bias gap" between FALCES variants with model pruning (solid lines) and those without. This allows our methods to consistently exhibit low bias, especially in comparison to state-of-the-art algorithms like Decouple. This behavior can be explained by the fact the *af* is used by model pruning where group imbalance can cause the pruning of otherwise good classifier combinations. Note that the use of *af* during the online-phase is not sensitive to the choice of the two metrics, because it ensures class balance in the local

region by selecting k members of each group to form a region. Consequently, FALCES and FALCES-SBT are not significantly affected by the choice of metric.

5.4 Runtime evaluation

We also evaluate the efficiency of our approach in its online phase. In particular, we study the effect of model pruning in the offline phase on the online performance. To this end, we run the four variants of FALCES and measure the average runtime to perform online classification for all tuples for which we want a prediction. We report results in Figure 9 on our real-world dataset, on which we can vary the number of groups (either 2 or 4), given two sensitive attributes. In any configuration, we see that model pruning during the offline phase improves the average runtime to classify a test tuple during the online phase. While this improvement is moderate when limiting to two groups, the difference increases as the number of groups increases. This can be explained based on the fact that for two groups and five models trained per group, we have 25 combinations to consider during the online phase when none are previously pruned. This exponentially increases with the number of groups, e.g., for 4 groups, 5^4 combinations need to be tested. Combined with the performance in terms of accuracy and fairness (see Section 5.2), FALCES-SBT-PFA is the method of choice when the number of groups increases.

6 Conclusion

This paper studied the novel problem of making locally fair and accurate classifications to foster equal opportunity decisions. We have presented a general framework to address the problem, as well as FALCES, an implementation of the framework that combines and extends techniques of dynamic model ensembles and fair model ensembles. Our experimental evaluation demonstrated that FALCES generally outperforms the state of the art when it comes to balancing accuracy and fairness for several types and degrees of bias present in the training dataset. Possible avenues for future research include methods that diversify the set of trained models in a controlled way or dynamic and adaptive setting of the parameter k of the kNN search, depending on the density of the data region.

Acknowledgements. Partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2120/1 - 390831618.

Bibliography

[Be75] Bentley, Jon Louis: Multidimensional binary search trees used for associative searching. Communications of the ACM, 18(9):509–517, 1975.

[Bh10] Bhatia, Nitin: Survey of nearest neighbor techniques. International Journal of Computer Science and Information Security (IJCSIS), 8(2):4, 2010.

[Co20] Coughlan, Sean: Why did the A-level algorithm say no? BBC.com, 2020.

[CSC18] Cruz, Rafael M.O.; Sabourin, Robert; Cavalcanti, George D.C.: Dynamic classifier selection: Recent advances and perspectives. Information Fusion, 41:195–216, 2018.

[CV10] Calders, Toon; Verwer, Sicco: Three naive Bayes approaches for discrimination-free classification. Data Mining and Knowledge Discovery, 21(2):277–292, 2010.

[Da18] Dastin, Jeffrey: Amazon scraps secret AI recruiting tool that showed bias against women. Reuters.com, 2018.

[DG19] Dua, Dheeru; Graff, Casey: UCI Machine Learning Repository. 2019.

[DSM19] Dvornik, Nikita; Schmid, Cordelia; Mairal, Julien: Diversity With Cooperation: Ensemble Methods for Few-Shot Classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3723 – 3731, 2019.

[Dw18] Dwork, Cynthia; Immorlica, Nicole; Kalai, Adam Tauman; Leiserson, Max: Decoupled Classifiers for Group-Fair and Efficient Machine Learning. In: Conference on Fairness, Accountability and Transparency. volume 81 of Proceedings of Machine Learning Research, pp. 119–133, 2018.

[FSV16] Friedler, Sorelle A.; Scheidegger, Carlos; Venkatasubramanian, Suresh: On the (im)possibility of fairness. arXiv preprint arXiv:1609.07236, p. 16, 2016.

[Ga19] Garg, Sahaj; Perot, Vincent; Limtiaco, Nicole; Taly, Ankur; Chi, Ed H.; Beutel, Alex: Counterfactual Fairness in Text Classification through Robustness. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. AIES '19. ACM, p. 219–226, 2019.

[KC09] Kamiran, Faisal; Calders, Toon: Classifying without discriminating. In: 2009 2nd International Conference on Computer, Control and Communication. IEEE, pp. 1–6, 2009.

[Po06] Polikar, Robi: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, 6(3):21–45, 2006.

[PRT08] Pedreschi, Dino; Ruggieri, Salvatore; Turini, Franco: Discrimination-aware Data Mining. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '08. ACM Press, pp. 560–568, 2008.

[SHX19] Shen, Zhiqiang; He, Zhankui; Xue, Xiangyang: MEAL: Multi-Model Ensemble via Adversarial Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 33:4886–4893, 2019.

[SYR13] Singh, Archana; Yadav, Avantika; Rana, Ajay: K-means with Three different Distance Metrics. International Journal of Computer Applications, 67(10):13–17, 2013.

[WKB97] Woods, Kevin; Kegelmeyer, W. Philip; Bowyer, Kevin: Combination of multiple classifiers using local accuracy estimates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):405–410, 1997.

[Zh19] Zheng, Hao; Zhang, Yizhe; Yang, Lin; Liang, Peixian; Zhao, Zhuo; Wang, Chaoli; Chen, Danny Z.: A New Ensemble Learning Framework for 3D Biomedical Image Segmentation. Proceedings of the AAAI Conference on Artificial Intelligence, 33:5909–5916, 2019.

[Žl17] Žliobaitė, Indrė: Measuring discrimination in algorithmic decision making. Data Mining and Knowledge Discovery, 31(4):1060–1089, 2017.