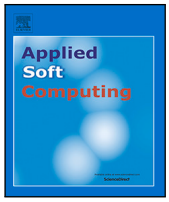
Applied Soft Computing Journal 167 (2024) 112313

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



Multiple fairness criteria in decision tree learning

Meryem Bagriacik *, Fernando E.B. Otero

School of Computing, University of Kent, Canterbury, CT2 7PE, United Kingdom

ARTICLE INFO

Keywords:
Fairness
Decision tree
Classification
Interpretability

ABSTRACT

The use of algorithmic decision-making systems based on machine learning models has led to a need for fair (unbiased) and explainable classification outcomes. In particular, machine learning algorithms can encode biases, which might result in discriminatory decisions for certain groups such as gender, race, or age. Although a number of works on decision tree learning have been proposed to decrease the chance of discrimination, they usually focus on the use of a single fairness metric. In general, creating a model based on a single fairness metric is not a sufficient way to mitigate discrimination since bias can originate from various sources—e.g., the data itself or the optimization process. In this paper, we propose a novel decision tree learning process that utilizes multiple fairness metrics to address both group and individual discrimination. This is achieved by extending the attribute selection procedure to consider not only information gain but also gain in fairness. Computational experiments on fourteen different datasets with various sensitive features demonstrate that the proposed FAIR-C4.5 models improve fairness without a loss in predictive accuracy when compared to the well-known C4.5 and the fairness-aware FFTree algorithms.

1. Introduction

It is difficult to identify an application area where Artificial Intelligence (AI) does not have a role to play—AI methods have become integral to many real-world applications. While AI has been very successful in recent years through the application of the supervised learning paradigm, it is becoming apparent that many of the popular methods create black-box (opaque) models that are prone to generating biased predictions, which could be racist and sexist [1–3]. This also limits their applicability in life-related situations, such as medical diagnoses, where understanding how AI methods reach a specific decision is crucial.

Supervised learning involves AI methods that *learn* based on data gathered from past experiences. In supervised learning, the goal is to build a model by finding patterns in the data that accurately represent the relationships between a set of predictors and an outcome of interest. The vast majority of learning methods rely solely on finding repeated patterns or correlations occurring in the data. This approach leads to several problems. First, in many cases, relationships supported by domain knowledge do not appear frequently enough in the data and therefore are not represented in the learned model. Second, the use of unrepresentative or biased data leads to the detection of accidental correlations, which are not useful to domain experts. Additionally, training data is often biased concerning potentially discriminatory attributes (e.g., gender, religion, race), which could lead to the creation of discriminatory models. In such cases, discriminatory bias can occur in different *flavours*: direct discrimination, where the model's predictions are based on discriminatory attributes, and indirect discrimination, where the model's predictions are based on attributes that are correlated with discriminatory attributes.

There are different aspects of fairness discussed in the literature: (i) fairness should consider specific individual pairs instead of average groups; (ii) similar individual groups should be treated in a similar way; and (iii) less-featured individuals should not be favoured over individuals with more features [4–6]. Fairness is defined in the literature as the absence of any prejudice or favouritism towards an individual or a group based on their inherent or acquired characteristics [7]. While a number of different fairness metrics have been previously proposed in the literature [8–10], it is becoming apparent that no single metric can be considered the best overall, as metrics capture fairness in different ways. Therefore, there is a clear motivation to employ multiple metrics in the creation of a model to achieve a more robust outcome—this is the focus of this paper.

Decision trees are widely used classification models. One of their main advantages is that they represent a comprehensible (white-box) model that can be interpreted by experts and users. A common approach to creating decision trees automatically from data is known as the *divide-and-conquer* approach [11], which consists of an iterative top-down procedure of selecting the best attribute to label an internal node of the tree. It starts by selecting an attribute to represent the

* Corresponding author.
E-mail address: mb2076@kent.ac.uk (M. Bagriacik).

https://doi.org/10.1016/j.asoc.2024.112313
Received 27 February 2024; Received in revised form 20 July 2024; Accepted 23 September 2024
Available online 2 October 2024

root of the tree. After selecting the first attribute, a branch for each possible (set of) value(s) of the attribute is created, and the data set is divided into subsets according to the examples' values of the selected attribute. The selection procedure is then recursively applied to each branch of the node using the corresponding subset of examples—i.e., the subset with examples that have the attribute's value associated with the branch. It stops for a given branch when all examples from the subset have the same class label or when another stopping criterion is satisfied, creating a leaf node to represent a class label to be predicted.

Current research aimed at reducing discrimination in decision tree learning mainly focuses on addressing bias through a single, specific fairness metric [12–16]. However, this approach has limitations, as various fairness metrics capture different dimensions of fairness and may not always align. In other words, a model deemed fair according to one metric might be considered unfair by another. Moreover, while efforts to improve fairness in these models have been made, such approaches often result in diminished accuracy. They generally overlook the potential for balancing fairness with accuracy, neglecting the opportunity to develop models that aim to meet fairness criteria while minimizing the impact on accuracy [12,14,15,17]. To overcome these limitations in the fairness literature, we propose extending the way attributes are selected during the decision tree learning process to consider their impact on the fairness of the model being created. This is achieved by using multiple fairness criteria for selecting attributes during tree creation. Attributes are then selected not only based on their impact on predictive performance but also on the fairness of the model. Computational results using fourteen datasets with different combinations of sensitive attributes (a total of twenty-three variants) show that the proposed Fair-C4.5 improves the fairness of the created models without negatively impacting their predictive accuracy.

The rest of the paper is organized as follows. Section 2 presents related work on fairness metrics and fairness-aware decision tree approaches. In Section 3, we discuss fairness-aware splitting procedures and algorithms for attribute selection to build fair and accurate decision tree models. Section 4 presents the datasets and experimental results. Finally, Section 5 presents the conclusion and direction for future research.

2. Related work

One of the initial studies on fairness in decision tree learning addressed discrimination within historical data and introduced a Discrimination-Aware Decision Tree model [13]. This model employed measures of accuracy and discrimination to guide the creation process for node splitting during the tree's construction, aiming to balance both accuracy and fairness. The objective was to develop a decision tree characterized by high accuracy and minimal discrimination concerning a sensitive attribute—i.e., the likelihood of a positive outcome remains constant irrespective of the sensitive attribute's value.

In another study based on adversarial training of decision trees, [14] developed a FATT (Fairness-Aware Tree Training) approach extended from the Meta-Silvae decision tree ensemble method according to the genetic algorithm as defined in [18]. Their work is based on an abstract interpretation that focuses on robust decision tree training by including similar individuals in the input space. Inspired by the individual fairness description in [4], they developed a fairness metric to find similarity relations. The new approach provides maximized fairness and accuracy for decision tree training and individual fairness verification of the decision tree. Experiments on CART and Random Forest show that this approach increased fairness by around 40% with a low decrease in accuracy, approximately 3%, while producing more interpretable and compact tree models.

Many studies focusing on decision tree learning for online streaming data employ statistical parity (or discrimination score) as a fairness metric. One of the first works addressing both fairness and concept drift is FEAT, which extends the Hoeffding Tree (HT) model [19]. FEAT is specifically designed to adapt to concept drift, taking into account non-stationary streaming data that may exhibit discrimination. This is achieved by reformulating the information gain metric to incorporate a fairness metric, resulting in fair-enhancing information gain (FEIG) for fair-splitting criteria. The first online Fair Random Forest (FARF) was proposed in [20] as a fair and adaptive random forest designed to manage fairness in online stream classification through fair statistical parity measurement. This is particularly important as online streaming data is subject to change over time. They designed an accumulative statistical parity as a fairness metric to assess online fairness, incorporating a single hyper-parameter to balance the trade-off between fairness and accuracy. In another study [12], a fairness-aware online decision tree was proposed. This tree is capable of processing data in a streaming environment without bias towards sensitive attributes, achieved by employing a fair information gain splitting procedure for tree construction.

In the method described in [21], Fairness-Aware Decision Tree Editing (FADE) employs Mixed-Integer Linear (MIL) optimization to minimize the discrimination score as part of a post-processing procedure. This involves modifying the decision tree by either deleting branches or altering labels on leaf nodes to adhere to fairness constraints. In a more recent study [22], Mixed-Integer Optimization (MIO) is utilized to derive optimal decision tree models. This approach integrates fairness constraints with MIO formulation to develop FairOCT. The study emphasizes five distinct group fairness metrics to ensure the learned tree meets fairness criteria. These metrics include statistical and conditional statistical parity, predictive equality (False Positive Rate), equalized opportunity (True Positive Rate), and equalized odds (a combination of True Positive and False Positive Rates).

Dynamic Programming (DP) is generally applied to design efficient optimal decision tree models, leveraging the separability or independence rule between left and right subtrees to achieve optimal accuracy. However, when considering fairness in decision tree models, those redesigned with DP fail to meet fairness criteria. This failure arises because fairness constraints cannot treat the left and right subtrees independently. Addressing this issue, [16] introduced DP Fair, which incorporates a global fairness constraint into Dynamic Programming. This is achieved by calculating the upper and lower bounds of the last fairness value in the partial solution, enabling early pruning of the tree to enhance fairness in the optimal decision tree. In this proposed approach, demographic parity, defined as equal positive prediction outcomes across all sensitive groups, is employed as the group fairness metric within the constraint.

The work in [23] combines demographic parity with ROC-AUC to develop the Splitting Criterion AUC for Fairness (SCAFF). This approach establishes a fair splitting criterion for fairness-aware learning from biased training datasets, ensuring that prediction outcomes are not influenced by sensitive attributes such as age or race. Moreover, their work demonstrates strong performance in both fairness and prediction accuracy by optimizing demographic parity across multiple sensitive attributes.

FairRepair, proposed in [24], aims to rectify unfairness in decision tree algorithms by modifying specific parts of the tree model in alignment with fairness constraints and semantic difference considerations. The method involves flipping leaves and refining paths after identifying unfair paths within the model using a MaxSMT-based tool. This approach employs a group fairness metric similar to the impact ratio for the flip and refine phases. Another method, known as Enforcing Fairness in Forests by Flipping Leaves (EiFFFeL) [25], focuses on flipping leaves in selected decision trees within a random forest, either by flipping all leaves or specifically those exhibiting the highest discrimination. The goal is to achieve group fairness while minimizing accuracy loss.

In a recent work [26], the authors introduced FFTree, a decision tree framework characterized by its transparency, flexibility, and sensitivity to fairness. FFTree incorporates multiple fairness metrics to filter

attributes before selection, with the final selection criterion being the maximization of information gain. For an attribute to be considered for selection, it must meet a fairness threshold, with all its fairness values falling below a predefined discrimination level denoted by δ. In instances where no attribute meets this criterion, the framework opts to create a leaf node instead. Thus, fairness indirectly influences attribute selection by acting as a filter for eligible attributes, although it does not directly determine the final choice of attribute.

Current research primarily focuses on mitigating discrimination through the application of single fairness metrics, which may not sufficiently address both direct and indirect forms of discrimination. Additionally, previous works often concentrated on enhancing the fairness of binary decision trees, accepting a certain degree of accuracy loss as a trade-off. In contrast, our methodology incorporates a variety of fairness metrics, including both group and individual fairness, to tackle different occurrences of discrimination more effectively. It employs a fairness-aware attribute selection procedure designed to balance accuracy and fairness, providing equal consideration to both aspects simultaneously. Finally, the proposed model extends the C4.5 decision tree algorithm to handle multi-valued attributes and missing values.

3. Learning fair decision trees

In this section, we introduce our proposed FAIR-C4.5 decision tree algorithm. Since bias or discrimination can occur at different degrees in data, and each fairness criterion might address discrimination differently, the proposed algorithm is designed to use a combination of multiple fairness criteria to guide the construction process. The rationale is to find an optimal balance between the different fairness measurements and predictive performance.

FAIR-C4.5 uses the concept of *sensitive* attributes to assess the fairness of a model. The goal is to generate a model that does not show bias in relation to the value of a sensitive attribute—i.e., the likelihood of a particular classification should not change due to a particular value of the sensitive attribute.

3.1. Fairness metrics

Let us consider S our set of instances, and $|S_P|$ and $|S_U|$ the number of privileged and unprivileged instances for a sensitive attribute, respectively. The number of privileged instances in the sensitive attribute labelled with the positive class is given by $|S_P^+|$; similarly, $|S_U^+|$ is the number of positively labelled unprivileged instances in the sensitive attribute.

In order to cope with missing values, we first determine the total number of missing values $|S_M|$, and then positively labelled missing values $|S_M^+|$. These values are used to estimate the fraction of missing values that belong to the privileged and unprivileged sets as follows:

$$F_P^+ = |S_M^+| * \frac{|S_P^+|}{|S^+| - |S_M^+|} \tag{1}$$

$$F_U^+ = |S_M^+| * \frac{|S_U^+|}{|S^+| - |S_M^+|} \tag{2}$$

where F_P^+ is the estimated fraction of the missing values in the sensitive attribute that are labelled with the positive class and belonging to a privileged group, while F_U^+ is the fraction of the missing values in the sensitive attribute that are labelled with the positive class belonging to a unprivileged group.

To calculate whether the classification is unfair or not based on different fairness metrics, we will use the probability of an instance with a privileged value being associated with the positive label $Pr(S_P^+)$ and the probability of an instance with an unprivileged value being associated with the positive label $Pr(S_U^+)$, as follows:

$$Pr(S_P^+) = \frac{|S_P^+| + F_P^+}{|S_P| + |S_M|} \tag{3}$$

$$Pr(S_U^+) = \frac{|S_U^+| + F_U^+}{|S_U| + |S_M|} \tag{4}$$

The definitions above are used to calculate four fairness measurements:

1. **Disparate Impact Ratio (DI)**: This metric, which was defined in [9,27], focuses on the rate of positive classification results across the unprivileged and privileged groups of a sensitive attribute. It has an optional value of 1, when both privileged and unprivileged groups have the same number of positive classifications, which can be interpreted as the case where the sensitive attribute does not introduce bias into the outcome of the classification. In our proposed approach, it is calculated as:

$$DI(S) = \frac{Pr(S_U^+)}{Pr(S_P^+)} \tag{5}$$

2. **Discrimination Score (CV)**: As used in [28–31], this metric represents the differences between the number of positively labelled instances in each group of the sensitive attribute. In this case, as the result approaches 0, the discrimination reduces since both groups have a similar number of positive classifications. It is calculated as:

$$CV(S) = Pr(S_P^+) - Pr(S_U^+) \tag{6}$$

3. **Consistency**: This metric compares the prediction of each instance to their k-nearest neighbours (kNN). A kNN result of 1 indicates completely consistent predictions, i.e., similar instances are classified the same way, while 0 would be a maximally inconsistent model. Consistency has been used as a fairness metric in [32]. It is calculated as:

$$Consistency(S) = 1 - \frac{1}{Nk} \sum_{i}^{N} \sum_{j \in \text{kNN}(i)} |\hat{y}_n - \hat{y}_j| \tag{7}$$

where N is the total number of instances and k is the number of neighbours to compare.

4. **Disparate Treatment (DT)**: This metric measures whether there is a difference in the classifier output when the sensitive attribute is taken into consideration or not, while other features are the same or similar; if the probability of predictions does not change, then there is no disparate treatment:

$$\begin{aligned} DT(S) =& Pr(C^+ \mid A_i, S_P) \\ &+ Pr(C^+ \mid A_i, S_U) \end{aligned} \tag{8}$$

where $Pr(C^+ \mid A_i, S_P)$ is the probability of a positive classification for privileged instances on the partition of a non-sensitive attribute A_i; $Pr(C^+ \mid A_i, S_U)$ is the probability of a positive classification for unprivileged instances on the partition of a non-sensitive attribute A_i. The details of the metric can be found in [10].

3.2. Fairness-based attribute selection

In our proposed work, we use multiple criteria to split the data during the decision tree creation; we, therefore, modified the attribute selection procedure. Firstly, the values for each criteria are calculated for all available attributes. Then, attributes' values for each criteria are ranked according to their gain—a matrix $A \times C$, where A is the number of attributes and C is the number of criteria, is created. In order to eliminate attributes that have a low chance of being selected, we identify attributes that are dominated by other attributes. An attribute a_x is considered dominated by another attribute a_y if for every criteria value, their values are equal or lower to the values of a_y and there is at least one value lower. At the end of this procedure, only non-dominated attributes are available for selection. Table 1 presents an example of

Table 1
Example of dominated attribute a_4 after the splitting criteria are calculated.

Attribute	*Entropy*	*DI*	*CV*	*CS*	*DT*
a_1	4	1	3	1	4
a_2	1	2	2	2	1
a_3	1	4	1	4	2
Dominated by a_2					
a_4	**3**	**3**	**4**	**2**	**3**

the rank matrix, where attribute a_2 dominates attribute a_4. In this case, attribute a_4 will not be considered for selection.

We employ three different strategies to select the attribute to split the data using the information of all criteria:

Lexicographic. In the lexicographic strategy, the selection follows a pre-defined order of metrics preference given by the user—the order dictates how attributes are compared. For example, consider a lexicographic strategy with $\{Entropy \rightarrow CV \rightarrow DI \rightarrow CS \rightarrow DT\}$ and the rank values presented in Table 1. In this case, the attribute selected would be a_3, since it ranks (jointly) first on *Entropy* and first on CV; there is no need to continue the comparison since at this point it is clearly the highest ranked attribute given the pre-defined order.

Constraint. The constraint strategy combines the rankings of the entropy metric with all other fairness metrics, subject to a constraint: only attributes where the sum of fairness metrics ranks is below a predefined threshold θ are considered; the attribute selected is the one that has the highest ranking overall. The rank of an attribute a is given by:

$$\begin{aligned} &Rank(a) = Rank(a, entropy) \times \sum_{m=1}^{M} Rank(a, m) \\ &\text{s.t.} \sum_{m=1}^{M} Rank(a, m) \leq \theta \end{aligned} \tag{9}$$

where M is the set of fairness metrics. The rationale of this strategy is to only consider attributes that have a clear contribution in improving the fairness of the model, regardless of whether they have a high rank for the entropy metric.

GRXFR. In the GRXFR strategy, the selection uses the original information gain in combination with the fairness metrics by multiplying the entropy rank with the average ranks of the fairness metrics. The rank used in the selection is given by:

$$GRXFR(a) = Rank(a, entropy) \times \frac{\sum_{m=1}^{M} Rank(a, m)}{|M|} \tag{10}$$

where M is the set of fairness metrics. The rationale of this strategy is to maximize the information (entropy) gain while taking into consideration the impact on fairness. Once the GRXFR value is calculated, the attribute with the lowest value is selected, which corresponds to the attribute with the best ranking.

4. Computational experiments

To evaluate the proposed Fair-C4.5, we used fourteen datasets that are widely used in the fairness literature. The datasets are Adult (**Adu**) [33]; German (**Ger**) [33]; Propublica Recidivism (**Prop**) and Propublica Violent Recidivism (**ProV**) [34]; NYPD SQF CPW (**NYP**)—related to racially-biased policy, NYPD: New York Police Department, SQF: stop, question, and frisk, CPW: Criminal Possession of a Weapon [35]; Student Mathematics (**StuM**) and Portuguese Performances (**StuP**) [36]; Drug Consumption (**Dru**) [37]; Ricci (**Ric**) [27]; Wine taste data (**Win**) [38]; Bank (**Bank**) [39]; Dutch (**Dut**) [13]; Law School admission (**Law**) [40]; UFRGS (**UF**)—Federal University of Rio Grande do Sul entrance exam and GPA data in Brazil—[24]. Table 2 provides the details of each dataset used in our experiments: the number of instances (**Size**); number of features (#); target class attribute (**Attribute**) and the value representing the positive class label (**Value(+)**); sensitive attribute's name (**Name**), privileged and unprivileged values (**P** and **U**), respectively. Each combination of dataset and sensitive attribute defines a variant, therefore, the algorithms are evaluated over twenty-three different variants. We compared the performance of Fair-C4.5 using the three proposed attribute selection strategies against the original C4.5 algorithm and FFTree [26] to evaluate the impact of extending its attribute selection to take into consideration fairness metrics. All algorithms were evaluated using a 10-fold cross-validation process, which consists of splitting the dataset into 10 partitions. Then, each partition is used as a test set, while the remaining nine are used as the training set. The final performance is then the average of the 10 executions.[1] All experiments are implemented in Python and run on a Windows PC 1.70 GHz Intel i5 with 16 GB of RAM.

In the result Tables 3–6, each dataset variation is represented by the abbreviated dataset name and sensitive attribute in parenthesis. For example, "Adu(G)" indicated the dataset "Adult" and the "Gender" sensitive attribute. The last line on each table presents the average rank of the Friedman statistical test with Hommel's post-hoc test [41,42]. We present the statistical test results at the bottom of the tables, p and $Hommel$ control value. The best result for each dataset variant is highlighted in boldface.

4.1. Performance and fairness results

In this section, we present the evaluation of our different fairness-aware splitting algorithms that are generated based on fairness measures and accuracy on the fourteen different datasets. We compared the fairness and predictive performance of three Fair-C4.5 tree-splitting variations that combine multiple fairness metrics and gain ratio against the standard C4.5 splitting method that uses only gain ratio. Additionally, to compare our proposed algorithm against an existing fairness-aware decision tree from the literature, we added results of the FFTree [26]. FFTree has been developed as a flexible fair decision tree algorithm to handle multiple fairness criteria by maximizing information gain among features that satisfy permitted discrimination-level fairness constraints. If fairness constraints are not satisfied during the selection of a node, the node added to the tree is a leaf—i.e., a class prediction. While the original paper proposing FFTree presented experiments with different discrimination levels, such as 0, 0.05, 0.1, 0.15, and 0.20, in our experiments they either generated default prediction without building a tree, or the same results for all available thresholds. Therefore, we only present FFTree results for 0.2 discrimination level when a decision tree is created; otherwise, a dash (-) represents the case where no decision tree was created.

Considering the prediction accuracy as shown in Table 3, the Constrain-based splitting procedure outperforms all other algorithms, including standard C4.5 and FFTree, with an average rank of 2.48. Additionally, the Constrain-based algorithm performance is statistically significantly better than FFTree, according to the non-parametric Friedman test. All of Fair-C4.5 approaches achieved better accuracy than the C4.5 and FFTree baseline algorithms for most datasets. In terms of ROC results, Lexicographic search shows the highest performance with a 2.39 average rank, and it is statistically significant better than FFTree. The ROC results of all Fair-C4.5 approaches are better than the standard C4.5 tree and FFTree for most of the datasets.

Table 4 indicates that GRXFR (Gain Ratio-Fairness) approach outperformed for False Positive Rates (FPR) and False Negative Rates (FNR) compared to C4.5 and other discrimination-aware algorithms, with average ranks of 2.48 and 2.61, respectively. Regarding Disparate Impact and CV Score results in Table 5, the Constrain-based model gave the best results, achieving average ranks of 2.48 and 2.61, respectively.

[1] All datasets and algorithms used in the evaluation can be found at: https://github.com/meryem1030/Fair-C4.5.git.

Table 2
Summary of the data sets used in the experiments.

Data set	#	Class		Size	Sensitive Attribute		
Variant		Attribute	Value (+)		Name	P	U
Adu	15	Income	$> 50k$	48 842			
Adu(G)					Gender	Male	Female
Adu(R)					Race	White	Non-White
Adu(A)					Age	≥ 25	< 25
Ger	22	Credit Status	2	1000			
Ger(G)					Gender	Male	Female
Ger(A)					Age	≥ 25	< 25
Prop	51	Two Year Recid	0	7214			
Prop(G)					Gender	Male	Female
Prop(R)					Race	White	Non-White
ProV	54	Two Year Recid	0	4743			
ProV(G)					Gender	Male	Female
ProV(R)					Race	White	Non-White
NYP	25	Weapon Found Flag	N	9826			
NYP(G)					Gender	Male	Female
NYP(R)					Race	White	Non-White
StuM	33	G3-binary	Pass	395			
StuM(G)					Gender	Male	Female
StuM(A)					Age	≥ 17	< 17
StuP	33	G3-binary	Pass	649			
StuP(G)					Gender	Male	Female
StuP(A)					Age	≥ 17	< 17
Dru	32	Meth	1	1885			
Dru(G)					Gender	Male	Female
Dru(R)					Race	White	Non-White
Ric	5	Combine	≥ 70	118			
Ric(R)					Race	White	Non-White
Win	13	binned quality	good	6497			
Win(T)					Type	White	Red
Bank	17	y	yes	40 004			
Bank(A)					Age	$(33 <, 60 >)$	$(33 \geq, 60 \leq)$
Dut	12	occupation	1	60 420			
Dut(G)					Gender	Male	Female
Law	17	pass bar	1	22 407			
Law(G)					Race	White	Non-White
UF	11	Mean GPA	≥ 3	43 303			
UF(G)					Gender	Male	Female

Table 3
Predictive performance of C4.5, FFTree and FAIR-C4.5 variants. The best result for each metric is shown in bold. Average ranks and statistical test results are displayed at the bottom.

Variant	C4.5		Lexicographic		Constraint		GRXFX		FFTree	
	ACC	ROC	ACC	ROC	ACC	ROC	ACC	ROC	ACC	ROC
Adu(G)	0.850	0.756	**0.850**	**0.756**	0.823	0.691	0.831	0.694	0.803	0.584
Adu(R)	0.850	0.756	**0.850**	**0.756**	0.842	0.714	0.828	0.699	0.803	0.584
Adu(A)	0.850	0.756	0.850	0.755	0.851	**0.756**	**0.851**	0.755	0.803	0.584
Ger(G)	0.681	0.603	0.679	0.598	**0.692**	**0.617**	0.683	0.599	–	–
Ger(A)	0.681	**0.603**	0.677	0.598	**0.685**	0.602	0.677	0.598	–	–
Ric(R)	0.873	0.875	0.873	0.875	0.864	0.866	**0.874**	**0.878**	0.610	0.595
Win(T)	0.637	0.617	0.637	0.618	**0.652**	**0.632**	0.649	0.630	0.640	0.588
StuM(G)	0.896	0.886	0.896	0.887	0.896	0.884	0.902	0.893	**0.909**	**0.916**
StuM(A)	0.896	0.886	**0.907**	**0.901**	0.902	0.891	0.899	0.891	–	–
StuP(G)	0.898	0.821	0.900	0.826	0.894	0.814	0.901	0.831	**0.912**	**0.838**
StuP(A)	**0.898**	0.821	0.897	**0.824**	0.886	0.798	0.891	0.809	–	–
NYP(G)	0.720	0.541	0.721	**0.543**	**0.737**	0.519	0.709	0.541	–	–
NYP(R)	0.720	**0.541**	0.719	0.539	0.731	0.518	**0.737**	0.530	–	–
ProV(G)	0.809	0.559	0.811	0.561	0.805	0.550	0.818	**0.571**	**0.834**	0.507
ProV(R)	0.809	0.559	0.810	**0.560**	0.812	0.553	0.810	0.555	**0.837**	0.500
Bank(A)	0.653	0.494	0.653	0.494	**0.770**	**0.583**	0.685	0.509	–	–
Prop(G)	0.654	0.642	0.655	0.643	**0.661**	**0.649**	0.659	0.646	0.646	0.635
Prop(R)	0.654	0.642	0.654	0.642	0.655	0.643	0.654	0.641	**0.668**	**0.664**
Dru(R)	0.788	0.721	0.786	0.718	**0.791**	**0.728**	0.787	0.720	–	–
Dru(G)	0.788	0.721	**0.789**	0.721	0.786	**0.723**	0.782	0.716	–	–
Dut(G)	0.729	0.730	**0.829**	**0.828**	0.811	0.809	0.808	0.805	0.587	0.604
Law(G)	0.904	0.538	0.904	0.538	0.905	0.528	0.913	0.531	**0.945**	**0.554**
UF(G)	**0.644**	**0.638**	0.644	0.638	0.644	0.638	0.642	0.636	0.538	0.507
Avg. rank	3.15	2.65	2.80	**2.39**	**2.48**	2.78	2.70	2.87	3.87	4.30
p	0.148	0.576	0.484	–	–	0.401	0.641	0.305	0.002	4.1E−05
Hommel	0.017	0.0125	0.025	–	–	0.025	0.05	0.017	0.0125	0.0125

Table 4
False Positive and False Negative Rates (fairness) performances of C4.5, FFTree and Fair-C4.5 variants. The best result for each metric is shown in bold. Average ranks and statistical test results are displayed at the bottom.

Variant	C4.5		Lexicographic		Constraint		GRXFX		FFTree	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Adu(G)	0.071	0.174	0.071	0.175	0.049	0.144	0.011	0.033	**0.000**	**0.016**
Adu(R)	0.036	0.042	0.036	0.048	0.021	0.049	0.034	0.072	**0.000**	**0.012**
Adu(A)	0.084	0.298	0.084	0.297	0.083	0.265	0.081	0.295	**0.001**	**0.034**
Ger(G)	0.108	0.203	0.133	0.198	**0.089**	0.160	0.117	**0.087**	–	–
Ger(A)	0.160	**0.112**	0.170	0.136	0.160	0.144	0.167	0.172	–	–
Ric(R)	0.113	0.167	0.113	**0.142**	0.113	0.183	**0.080**	0.167	0.100	0.233
Win(T)	0.417	0.307	0.413	0.308	**0.405**	0.298	0.407	0.297	0.534	**0.213**
StuM(G)	0.132	0.082	**0.116**	0.093	0.167	0.096	0.130	**0.072**	0.124	0.087
StuM(A)	0.145	0.189	0.134	0.198	0.1372	**0.186**	**0.130**	0.194	–	–
StuP(G)	**0.185**	0.084	0.290	0.077	0.280	0.077	0.238	0.075	**0.140**	**0.062**
StuP(A)	**0.275**	**0.050**	0.284	0.070	0.284	0.118	0.323	0.115	–	–
NYP(G)	0.139	**0.042**	0.144	0.047	0.154	0.048	**0.118**	0.048	–	–
NYP(R)	0.081	0.058	0.084	0.058	**0.075**	**0.033**	0.105	0.072	–	–
ProV(G)	0.115	0.035	0.118	0.037	0.138	0.047	0.109	0.035	**0.015**	**0.006**
ProV(R)	0.201	0.047	0.798	0.076	0.188	0.037	0.127	0.048	**0.000**	**0.000**
Bank(A)	0.043	0.065	**0.039**	0.066	0.041	0.071	0.041	**0.052**	–	–
Prop(G)	0.148	0.099	0.154	0.098	0.122	0.101	0.120	**0.094**	**0.115**	0.107
Prop(R)	0.193	0.079	0.181	0.080	0.180	**0.078**	0.189	0.079	**0.128**	0.113
Dru(R)	0.325	0.132	0.306	0.103	0.307	**0.093**	**0.300**	0.120	–	–
Dru(G)	0.103	0.082	0.111	**0.077**	**0.100**	0.079	0.138	0.079	–	–
Dut(G)	0.289	0.423	0.202	0.070	**0.040**	0.048	0.043	0.048	0.049	**0.032**
Law(G)	0.108	0.064	0.103	0.063	0.113	0.055	**0.092**	0.059	0.178	**0.038**
UF(G)	0.439	0.368	0.439	0.368	0.446	0.364	0.446	**0.363**	**0.004**	0.984
Avg. rank	3.37	3.04	3.26	3.22	2.80	2.78	**2.48**	**2.61**	3.09	3.35
p	0.056	0.351	0.093	0.192	0.484	0.709	–	–	0.192	0.113
Hommel	0.0125	0.025	0.017	0.017	0.05	0.05	–	–	0.025	0.0125

Table 5
Disparate Impact and CV score (fairness) performances of C4.5, FFTree and Fair-C4.5 variants. The best result for each metric is shown in bold. Average ranks and statistical test results are displayed at the bottom.

Variant	C4.5		Lexicographic		Constraint		GRXFX		FFTree	
	DI	CV	DI	CV	DI	CV	DI	CV	DI	CV
Adu(G)	0.269	0.180	0.269	0.180	0.319	0.133	**0.474**	0.088	0.410	**0.029**
Adu(R)	2.157	−0.105	2.171	−0.106	2.159	−0.082	2.364	−0.092	**2.002**	**−0.021**
Adu(A)	0.016	0.223	0.016	0.221	0.017	0.220	0.016	0.219	**0.073**	**0.045**
Ger(G)	1.280	−0.052	1.370	−0.067	1.270	−0.062	**1.262**	**−0.026**	–	–
Ger(A)	**1.192**	−0.025	1.217	**−0.024**	1.546	−0.103	1.288	−0.049	–	–
Ric(R)	0.461	0.400	0.503	0.365	0.461	0.387	**0.518**	0.356	0.238	**0.031**
Win(T)	0.259	0.532	0.261	**0.531**	0.229	0.558	**0.266**	0.532	0.227	0.682
StuM(G)	0.927	0.079	0.959	0.066	0.958	**0.060**	0.921	0.084	**0.959**	0.065
StuM(A)	0.469	−0.008	0.467	**0.004**	**0.473**	−0.015	**0.473**	−0.007	–	–
StuP(G)	1.104	−0.063	1.110	−0.063	1.098	−0.055	1.099	−0.058	**1.059**	**−0.021**
StuP(A)	1.300	**0.026**	**1.276**	0.046	1.284	0.040	1.279	0.041	–	–
NYP(G)	1.042	−0.034	1.040	−0.032	**1.015**	**−0.013**	1.022	−0.019	–	–
NYP(R)	0.961	0.035	0.971	0.026	**1.002**	**−0.001**	0.935	0.064	–	–
ProV(G)	1.058	−0.052	1.056	−0.050	1.055	−0.049	1.054	−0.049	**1.009**	**−0.008**
ProV(R)	0.993	0.010	1.057	−0.051	0.986	0.016	0.952	0.048	**1.000**	0.000
Bank(A)	1.545	−0.033	1.543	**−0.030**	1.150	−0.031	**1.401**	−0.035	–	–
Prop(G)	1.233	−0.142	1.234	−0.143	1.222	−0.135	**1.213**	**−0.132**	1.236	−0.139
Prop(R)	0.827	0.135	**0.833**	0.130	0.832	**0.129**	0.822	0.140	0.809	0.136
Dru(R)	1.036	−0.031	**1.032**	**−0.029**	1.048	−0.040	1.036	−0.031	–	–
Dru(G)	0.845	0.131	0.853	0.124	**0.860**	**0.118**	0.847	0.129	–	–
Dut(G)	0.378	0.459	0.483	0.323	0.646	0.183	0.644	0.182	**0.941**	**0.053**
Law(G)	0.917	0.078	0.918	0.078	0.926	0.069	0.926	0.070	**0.935**	**0.064**
UF(G)	0.480	0.288	**0.481**	0.287	0.475	0.292	0.480	0.291	0.086	**0.002**
Avg. rank	3.46	3.52	2.95	3.04	**2.48**	**2.61**	2.63	2.83	3.48	3
p	0.036	0.0502	0.305	0.3512	–	–	0.744	0.641	0.032	0.401
Hommel	0.0167	0.0125	0.025	0.0167	–	–	0.05	0.05	0.0125	0.025

Comparing consistency results as shown in Table 6, the FFTree baseline algorithm outperformed all other algorithms with an average rank of 2.57, although the differences are not statistically significant. However, FFTree did not build a decision tree for nine datasets and instead provided a leaf node according to the majority class value. This occurs when none of the attribute splits satisfy FFTree's fairness criteria. In the case of Disparate Treatment, an individual fairness metric, the results provided an interesting insight. FFTree's approach to handle disparate treatment is to omit the sensitive attribute from the information gain calculation, thereby not influencing attribute selection. The rationale is to provide *fairness through blindness*, also referred to as Fairness Through Unawareness [26,43], which explicitly avoids using the sensitive attribute in the decision tree. Our results show that this might not necessarily improve fairness, as it does not prevent non-sensitive attributes correlated to the sensitive attribute from being used. The Lexicographic variation obtained the best average rank of 2.65 for Disparate Treatment, as shown in Table 6, while FFTree had the worst average rank of 3.57—although the differences are not statistically significant.

Table 6
Consistency and Disparate Treatment (fairness) performances of C4.5, FFTree and Fair-C4.5 variants. The best result for each metric is shown in bold. Average ranks and statistical test results are displayed at the bottom.

Variant	C4.5		Lexicographic		Constraint		GRXFX		FFTree	
	CO	DT	CO	DT	CO	DT	CO	DT	CO	DT
Adu(G)	0.841	0.676	0.841	0.676	0.870	0.625	0.874	0.643	**0.993**	**0.214**
Adu(R)	0.842	0.640	0.842	0.634	0.889	0.512	0.886	0.535	**0.991**	**0.213**
Adu(A)	0.887	0.219	0.887	0.219	0.887	0.217	0.887	0.211	**0.996**	**0.122**
Ger(G)	0.696	0.765	0.696	0.769	**0.683**	0.803	0.691	**0.735**	–	–
Ger(A)	0.671	**0.702**	0.673	0.709	**0.683**	0.852	0.674	0.781	–	–
Ric(R)	0.738	0.651	0.730	0.636	0.739	0.645	0.722	0.698	**0.882**	**0.206**
Win(T)	0.537	0.390	0.536	0.387	0.540	**0.362**	0.541	0.395	**0.599**	0.420
StuM(G)	0.814	1.345	0.820	1.328	0.815	1.347	0.822	1.327	**0.825**	**1.286**
StuM(A)	0.703	0.694	**0.707**	**0.689**	0.703	0.695	0.670	0.695	–	–
StuP(G)	0.865	1.742	0.869	1.739	0.863	1.738	0.870	**1.737**	**0.898**	1.795
StuP(A)	0.798	1.373	**0.801**	**1.331**	0.796	1.360	0.795	1.364	–	–
NYP(G)	0.797	5.903	0.794	5.876	**0.844**	6.143	0.802	**5.646**	–	–
NYP(R)	0.807	5.273	0.799	**5.242**	0.833	5.378	**0.856**	5.564	–	–
ProV(G)	0.919	**4.490**	0.918	4.491	0.923	4.499	0.928	4.531	**0.997**	4.829
ProV(R)	0.907	**2.739**	0.918	4.497	0.918	2.748	0.931	2.863	**1.000**	3.082
Bank(A)	0.811	1.015	0.809	1.010	**0.856**	**0.720**	0.816	0.925	–	–
Prop(G)	0.862	1.676	0.864	1.685	0.863	1.662	0.872	1.667	**0.940**	**1.640**
Prop(R)	0.844	1.439	0.844	1.431	0.836	1.412	0.846	1.450	**0.895**	**1.216**
Dru(R)	**0.796**	2.317	0.790	**2.314**	0.787	2.330	0.787	2.315	–	–
Dru(G)	0.786	2.919	**0.787**	2.904	0.786	**2.866**	0.783	2.898	–	–
Dut(G)	0.917	1.719	0.930	1.617	0.930	1.543	0.927	**1.527**	**0.978**	3.166
Law(G)	0.914	2.519	0.914	**2.514**	0.913	2.556	0.925	2.569	**0.975**	2.672
UF(G)	0.849	1.038	0.849	1.038	0.849	1.042	0.845	1.044	**0.995**	**0.042**
Avg. rank	3.48	3.17	3.30	**2.65**	2.94	2.71	2.72	2.89	**2.57**	3.57
p	0.0502	0.263	0.113	–	0.428	0.8888	0.744	0.608	–	0.0502
Hommel	0.0125	0.0167	0.0167	–	0.025	0.05	0.05	0.025	–	0.0125

Table 7
Size of the decision trees created by each algorithm used in our experiments.

Variant	C4.5	Lexicographic	Constraint	GRXFR	FFTree (0.2)
$Adu(G)$	2138	2125	2908	1641	2
$Adu(R)$	2138	2125	935	2463	2
$Adu(A)$	2138	2125	2248	2259	2
$Ger(G)$	352.3	347.9	362.9	353.9	1
$Ger(A)$	352.3	348.9	358	342.3	1
$Ric(R)$	1	4	1	4	2
$Win(T)$	469.8	476.7	510	462.8	6.1
$StuM(G)$	31.2	31.7	31.4	32.2	7.2
$StuM(A)$	31.2	31.1	31	31.8	1
$StuP(G)$	41	42.8	41.9	41.7	11.6
$StuP(A)$	41	42.8	43	42.1	1.5
$NYP(G)$	63 470.3	54 691.8	44 336.9	31 708.3	1
$NYP(R)$	63 470.3	55 170.7	48 542.5	71149.4	1
$ProV(G)$	879.9	873.4	937.8	3361	2.2
$ProV(R)$	879.9	880.5	987.1	5878.6	2
$Bank(A)$	3579.5	3581.1	3328	4297.2	1.3
$Prop(G)$	554.9	555.8	519.3	524.9	2.4
$Prop(R)$	554.9	553.3	515.2	550.9	5.9
$Dru(R)$	626.5	642.8	644.7	639.2	1
$Dru(G)$	626.5	641.6	654.2	638.1	1
$Dut(G)$	383.7	3188.5	3213.9	4126.6	2
$Law(G)$	331.6	330.4	348.7	351	8.2
$UF(G)$	296.9	297.6	300.4	309.4	2

Looking at the size of the trees in Table 7, particularly for FFTree, we observe that setting a strict threshold to enhance fairness leads to a negative impact on tree construction due to the early stopping of the decision tree growth. It is clear for most of the datasets that our proposed fair tree-building approaches produced trees of similar size to the standard C4.5 algorithm, without significantly increasing the number of nodes.

4.2. Discussion

Our aim is not only to decrease discrimination but also to maintain the same level of predictive accuracy as possible. As seen in the results presented, the Lexicographic approach provides higher accuracy and lower discrimination scores compared to C4.5, although the Lexicographic approach prioritizes the gain ratio, making it slightly different from C4.5. This shows the advantage of combining fairness metrics with gain ratio during the selection of attributes. Based on the numerical results presented in Tables 3, 4, 5 and 6, the Constraint-based variation achieved the highest predictive accuracy and the lowest discrimination scores for most of the datasets. GRXFR is one of the proposed discrimination-aware splitting approaches generated based on the idea of fair information gain [12,15,19,20]. It shows better predictive accuracy and lower discrimination scores when compared to C4.5 and FFTree algorithms.

Overall, both models outperformed the standard C4.5 and FFTree algorithms for most of the datasets—the latter being a fairness-aware decision tree proposed in the literature. Considering the accuracy-fairness trade-off, existing works [13,15,26,44] reported a drop in

accuracy to achieve an improvement in fairness. In contrast, for most of the datasets—in particular the Adult, Propublica, and Bank datasets used by other works in the literature—our Fair-C4.5 models achieved a good balance between accuracy and fairness. This is particularly evident in the results achieved by the Constraint approach.

Considering the computational time complexity, both the proposed Fair-C4.5 and FFTree have a higher time complexity than the C4.5 algorithm. In general, decision tree algorithms following a divide-and-conquer approach have the following time complexity. To split the input data and select a node of the tree, the algorithm needs to determine the class distribution of n data inputs; the cost of this step is $O(n)$. Since the algorithm has m attributes to choose from, and for numeric attributes the values must be sorted, the cost of this step is $O(mn \log_2 n)$. Finally, the cost complexity to evaluate each partition of the data, given the recursive nature of the divide-and-conquer process, is $O(n \log_2 n)$. Therefore, the total time complexity for C4.5 algorithm can be defined as the combination of each step, $O(n) + O(mn \log_2 n) + O(n \log_2 n)$, where $O(mn \log_2 n)$ is the dominant factor for the time complexity. Since Fair-C4.5 and FFTree incorporate the use of fairness metrics into the evaluation of each split point, the second step the process described above is expanded to include t metrics, resulting in the cost being $O(tmn \log_2 n)$—t in most cases is smaller than m and n, and therefore n is still the variable with the highest influence in the time complexity of the algorithms.

As a result of the increase in time complexity, Fair-C4.5 training times are approximately four times higher than C4.5 and two times higher than FFTree—this is due to the factor t, which represents the multiple fairness metrics. It is important to note that training times have a relatively small importance in applications where fairness of the model is a requirement. Additionally, many data mining applications involve off-line execution and the time spent collection and preparing the data is usually much greater than the time required to train a model. There are also opportunities to improve the computation time of Fair-C4.5 by parallelizing the evaluation of each fairness metric for application where training time becomes a significant issue.

5. Conclusion and future work

In this paper, we proposed a novel fairness-aware decision tree algorithm aimed at creating discrimination-aware decision tree models to improve the overall fairness of their predictions. While various studies have employed the CART binary splitting approach in developing discrimination-aware decision trees, our algorithm uses non-binary splitting inspired by the C4.5 decision tree algorithm. Moreover, our approach utilizes a variety of fairness metrics, categorized into group and individual fairness measures, to improve both types of fairness, unlike previous works in the literature.

To address the trade-off between accuracy and fairness, we proposed attribute selection approaches that balance information gain and fairness. Our empirical evaluations across various datasets showed that our proposed algorithm improved the overall fairness of the models while maintaining the same level of accuracy. This is particularly evident with our Constraint-based approach, which demonstrates statistically significant improvements in accuracy compared to FFTree, a fairness-aware decision tree algorithm from the literature.

There are several potential avenues for future research. Currently, the fairness metrics only influence the selection of attributes during decision tree construction. A natural future research direction would be to extend their influence to the selection of continuous attribute thresholds, favouring threshold values that improve fairness, and to the pruning of the decision tree. Additionally, it would be interesting to investigate different ways to combine fairness metrics and information gain, beyond the approaches proposed. Exploring how to combine multiple fairness metrics in decision tree-based ensemble models is a research direction worth further exploration.

CRediT authorship contribution statement

Meryem Bagriacik: Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Fernando E.B. Otero:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link for data and codes are shared as a footnote in the article.

Acknowledgement

Meryem Bagriacik was supported by funding from the Ministry of National Education, Republic of Turkey, through the MoNE-YLSY scholarship programme.

References

[1] A. Caliskan, J.J. Bryson, A. Narayanan, Semantics derived automatically from language corpora necessarily contain human biases, Phys. Rep.-Rev. Sec. Phys. Lett. 356 (2017) 183–186.

[2] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, PMLR, 2018, pp. 77–91.

[3] J. Zou, L. Schiebinger, AI Can Be Sexist and Racist—It's Time to Make It Fair, Nature Publishing Group UK London, 2018.

[4] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 214–226.

[5] M. Joseph, M. Kearns, J. Morgenstern, A. Roth, Fairness in learning: Classic and contextual bandits, Adv. Neural Inf. Process. Syst. (2016) 325–333, http://dx.doi.org/10.48550/arxiv.1605.07139, URL https://arxiv.org/abs/1605.07139v2.

[6] A. Chouldechova, A. Roth, The frontiers of fairness in machine learning, 2018, http://dx.doi.org/10.48550/arxiv.1810.08810, URL http://arxiv.org/abs/1810.08810.

[7] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2019) http://dx.doi.org/10.1145/3457607, URL https://arxiv.org/abs/1908.09635v3.

[8] I. Žliobaitė, Measuring discrimination in algorithmic decision making, Data Min. Knowl. Discov. 31 (4) (2017) 1060–1089, http://dx.doi.org/10.1007/S10618-017-0506-1/FIGURES/3, URL https://link.springer.com/article/10.1007/s10618-017-0506-1.

[9] M.B. Zafar, I. Valera, M. Gomez Rodriguez, K.P. Gummadi, Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1171–1180.

[10] M.B. Zafar, Discrimination in Algorithmic Decision Making: From Principles to Measures and Mechanisms, Saarländische Universitäts-und Landesbibliothek, 2019.

[11] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[12] W. Zhang, L. Zhao, Online decision trees with fairness, 2020, arXiv preprint arXiv:2010.08146.

[13] F. Kamiran, T. Calders, M. Pechenizkiy, Discrimination aware decision tree learning, 2010, http://dx.doi.org/10.1109/ICDM.2010.50.

[14] F. Ranzato, C. Urban, M. Zanella, Fair training of decision tree classifiers, 2021, http://dx.doi.org/10.48550/arxiv.2101.00909, URL https://arxiv.org/abs/2101.00909v1.

[15] W. Zhang, E. Ntoutsi, Faht: An adaptive fairness-aware decision tree classifier, in: IJCAI International Joint Conference on Artificial Intelligence, Vol. 2019-August, 2019, http://dx.doi.org/10.24963/ijcai.2019/205.

[16] J.G.M. Van Der Linden, M.M. De Weerdt, D. Demirovi'c, Fair and optimal decision trees: A dynamic programming approach, in: Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 38899–38911.

[17] S. Aghaei, M.J. Azizi, P. Vayanos, Learning optimal and fair decision trees for non-discriminative decision-making, in: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, AAAI Press, 2019, pp. 1418–1426, http://dx.doi.org/10.1609/AAAI.V33I01.33011418, URL https://dl.acm.org/doi/10.1609/aaai.v33i01.33011418.

[18] F. Ranzato, M. Zanella, Genetic adversarial training of decision trees, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2021, pp. 358–367.

[19] W. Zhang, A. Bifet, Feat: A fairness-enhancing and concept-adapting decision tree classifier, in: Discovery Science: 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23, Springer, 2020, pp. 175–189.

[20] W. Zhang, A. Bifet, X. Zhang, J.C. Weiss, W. Nejdl, Farf: A fair and adaptive random forests classifier, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2021, pp. 245–256.

[21] K. Kanamori, H. Arimura, Fairness-aware decision tree editing based on mixed-integer linear optimization, Trans. Jpn. Soc. Artif. Intell. 36 (4) (2021) B–L13_1–B–L13_10, http://dx.doi.org/10.1527/TJSAI.36-4{_}B-L13.

[22] N. Jo, S. Aghaei, J. Benson, A. Gómez, P. Vayanos, Learning optimal fair classification trees, 2022, http://dx.doi.org/10.48550/arxiv.2201.09932, URL https://arxiv.org/abs/2201.09932v2.

[23] A. Pereira Barata, F.W. Takes, H.J. van den Herik, C.J. Veenman, Fair tree classifier using strong demographic parity, Mach. Learn. (2023) 1–20, http://dx.doi.org/10.1007/S10994-023-06376-Z/FIGURES/7.

[24] J. Zhang, I. Beschastnikh, S. Mechtaev, A. Roychoudhury, Fair decision making via automated repair of decision trees, in: Proceedings of the 2nd International Workshop on Equitable Data and Technology, 2022, pp. 9–16.

[25] S.A. Abebe, C. Lucchese, S. Orlando, Eifffel: enforcing fairness in forests by flipping leaves, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, 2022, pp. 429–436.

[26] A. Castelnovo, A. Cosentini, L. Malandri, F. Mercorio, M. Mezzanzanica, FFTree: A flexible tree to handle multiple fairness criteria, Inf. Process. Manage. 59 (6) (2022) http://dx.doi.org/10.1016/j.ipm.2022.103099.

[27] S.A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E.P. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 329–338.

[28] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, E. Ntoutsi, A survey on datasets for fairness-aware machine learning, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 12 (3) (2022) e1452, http://dx.doi.org/10.1002/WIDM.1452.

[29] Y. Li, L. Meng, L. Chen, L. Yu, D. Wu, Y. Zhou, B. Xu, Training data debugging for the fairness of machine learning software, in: Proceedings of the 44th International Conference on Software Engineering, 2022, pp. 2215–2227.

[30] Q. Zhang, J. Liu, Z. Zhang, J. Wen, B. Mao, X. Yao, Mitigating unfairness via evolutionary multi-objective ensemble learning, IEEE Trans. Evol. Comput. (2022) 1, http://dx.doi.org/10.1109/tevc.2022.3209544.

[31] D. Pessach, E. Shmueli, A review on fairness in machine learning, ACM Comput. Surv. 55 (3) (2023) 1–44, http://dx.doi.org/10.1145/3494672.

[32] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: International Conference on Machine Learning, PMLR, 2013, pp. 325–333.

[33] M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 259–268.

[34] A. Valdivia, J. Sánchez-Monedero, J. Casillas, How fair can we go in machine learning? Assessing the boundaries of accuracy and fairness, Int. J. Intell. Syst. 36 (4) (2021) 1619–1643.

[35] S. Goel, J.M. Rao, R. Shroff, Precinct or prejudice? Understanding racial disparities in New York City's stop-and-frisk policy 10(1), (ISSN: 1932-6157) 2016, pp. 365–394, http://dx.doi.org/10.1214/15-AOAS897.

[36] P. Cortez, A.M.G. Silva, Using data mining to predict secondary school student performance, 2008.

[37] E. Fehrman, V. Egan, A.N. Gorban, J. Levesley, E.M. Mirkes, A.K. Muhammad, Personality Traits and Drug Consumption, Springer, 2019.

[38] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (4) (2009) 547–553, http://dx.doi.org/10.1016/J.DSS.2009.05.016.

[39] V. Grari, B. Ruf, S. Lamprier, M. Detyniecki, Fair adversarial gradient tree boosting, in: 2019 IEEE International Conference on Data Mining, ICDM, 2019, pp. 1060–1065, http://dx.doi.org/10.1109/ICDM.2019.00124.

[40] Y. Bechavod, K. Ligett, Penalizing unfairness in binary classification, 2017, URL https://arxiv.org/abs/1707.00044v3.

[41] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Mach. Learn. Res. 9 (2008) 2677–2694.

[42] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Mach. Learn. Res. 7 (2006) 1–30.

[43] S. Verma, J. Rubin, Fairness definitions explained, in: Proceedings of the International Workshop on Software Fairness, FairWare '18, Association for Computing Machinery, 2018, pp. 1–7.

[44] S. Ravichandran, D. Khurana, A. Labs, A. Express Bangalore, K. Bharath Venkatesh, N. Unny Edakunni, B. Venkatesh, FairXGBoost: Fairness-aware classification in XGBoost, 2020, http://dx.doi.org/10.1145/1122445.1122456.