EiFFFeL: Enforcing Fairness in Forests by Flipping Leaves

Seyum Assefa Abebe
Ca' Foscari University of Venice
Venice, Italy
seyumassefa.abebe@unive.it

Claudio Lucchese
Ca' Foscari University of Venice
Venice, Italy
claudio.lucchese@unive.it

Salvatore Orlando
Ca' Foscari University of Venice
Venice, Italy
orlando@unive.it

ABSTRACT

Nowadays Machine Learning (ML) techniques are extensively adopted in many socially sensitive systems, thus requiring to carefully study the fairness of the decisions taken by such systems. Many approaches have been proposed to address and to make sure there is no bias against individuals or specific groups which might originally come from biased training datasets or algorithm design. In this regard, we propose a fairness enforcing approach called EiFFFeL –Enforcing Fairness in Forests by Flipping Leaves– which exploits tree-based or leaf-based post-processing strategies to relabel leaves of selected decision trees of a given forest. Experimental results show that our approach achieves a user-defined group fairness degree without losing a significant amount of accuracy.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**;
• **Human-centered computing**;

KEYWORDS

Fair machine learning, Random forest, Group Fairness

ACM Reference Format:
Seyum Assefa Abebe, Claudio Lucchese, and Salvatore Orlando. 2022. EiFFFeL: Enforcing Fairness in Forests by Flipping Leaves. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event,* . ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3477314.3507319

1 INTRODUCTION

Machine Learning is used in a wide range of systems, such as health care [9, 20], education [24–26], financial lending [4, 21], and social services [12, 23], to facilitate decision making and automate services which has a critical implications to individuals and communities. This extensive use of machine learning creates a growing concern, as algorithms might introduce far-reaching bias that treats individuals or groups unfairly, based on certain characteristics such as age, race, gender, or political affiliation. Thus, it is becoming very important to develop fairness aware algorithms.

In recent years many methods have been developed addressing both individual-based and group-based fairness. Most of the works tackles issues of discovering discrimination, and adding solutions to eliminate such discrimination to have fair and accurate decisions [31, 32]. The bias mitigation approaches on either the training data or the learned model can be categorized into three main groups [8]: (1) Pre-processing approaches such as reweighting [5], massaging [17], aiming to eliminate discrimination at the data level; (2) In-processing algorithms such as [3, 18] addressing discrimination by modifying algorithms; (3) post-processing methods such as [15, 18] take the learned model and change the predicted labels appropriately to meet fairness requirements.

Algorithms proposed recently in bias mitigation has focused on neural networks. However, the efficiency and explainability of tree ensembles for many applications makes them preferable to be implemented in many areas. Even though there are few works focused on studying fairness for trees and tree ensembles, notably [14, 18, 28, 30], most of them are focused on single decision tree classifiers and in-processing approaches. Our interest mainly lies in developing fair random forest classifiers with post-processing approaches designed to relabel leaves with accuracy and discrimination constraints. We take advantage of implementing a post-process approach, in which we do not require to know the training process.

Contributions. We focus on decision tree ensembles for binary classification tasks susceptible to group discrimination with respect to attributes sensitive classes such as age, gender, race, etc. We propose a post-processing approach named EiFFFeL –Enforcing Fairness in Forests by Flipping Leaves– that given a forest, however trained, selects a subset of its leaves and changes their predictions so as to reduce the discrimination degree of the forest.

We summarize the main contributions of our work as follows.

(1) We propose an iterative leaf flipping post-processing algorithm to ensure group fairness .
(2) We devise tree-based and leaf-based flipping methodologies on top of random forest classifier to enforce fairness.
(3) We report experimental evaluations of group fairness on three different datasets, aiming to empirically show the effectiveness of our method.

2 FAIRNESS IN MACHINE LEARNING

Without loss of generality, we consider a binary classifier $g : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an input feature vector $\boldsymbol{x} \in \mathcal{X}$ to a binary class label $y \in \mathcal{Y} = \{0, 1\}$. Among the attributes in the feature space $\mathcal{X}$, a binary attribute called *sensitive feature* $S \in \{0, 1\}$ identifies the aspects of data which are socio-culturally precarious for the application of machine learning. Specifically, given $x \in \mathcal{X}$ and $x.S$ the value of the sensitive attribute S for the given instance, if $x.S = 0$ then we say that x belongs to the *unprivileged group* that could possibly be discriminated.

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/10.1145/3477314.3507319

2.1 Fairness and Discrimination Definitions

To achieve non-discriminatory and fair machine learning model, it is essential to first define *fairness*. In a broad context, fairness can be seen from an individual or a group point of view. *Individual fairness* requires that similar individuals being treated similarly. *Group fairness* requires fairness of the classification model to apply on the two groups, defined through the binary sensitive feature S [11]. Our work focuses on group fairness, in which a group of individuals identified by S risks for experiencing discrimination.

We define the discrimination of a classifier measured by *group fairness* as follows. Recall that attribute $S = 0$ identifies the unprivileged group, while $S = 1$ corresponds to the privileged one, whose members are not discriminated but rather favoured by a learnt ML model. Moreover, we assume that the values 1 and 0 of class label Y represent *favorable* and *unfavorable* outcomes, respectively. For example, $Y = 1$ might correspond to the decision of granting a loan, thus favouring a bank customer.

A classifier g applied over $x \in \mathcal{X}$ is non-discriminatory if its prediction $g(\boldsymbol{x})$ is statistically independent of the sensitive attribute S. Hence, a classifier is fair if both groups have equal probability of being classified as belonging to the favorable class, which is the desirable outcome.

Using the problem formalization by [18], the discrimination of a model g with respect to a sensitive attribute S and a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ can be computed as follows:

$$disc_{\mathcal{D},S,g} := \frac{|\{(\boldsymbol{x}, y) \in \mathcal{D} \mid \boldsymbol{x}.S = 1 \wedge g(\boldsymbol{x}) = 1\}|}{|\{(\boldsymbol{x}, y) \in \mathcal{D} \mid \boldsymbol{x}.S = 1\}|} - \frac{|\{(\boldsymbol{x}, y) \in \mathcal{D} \mid \boldsymbol{x}.S = 0 \wedge g(\boldsymbol{x}) = 1\}|}{|\{(\boldsymbol{x}, y) \in \mathcal{D} \mid \boldsymbol{x}.S = 0\}|},$$

where $\boldsymbol{x}.S$ refers to the sensitive attribute of the instance $\boldsymbol{x}$. When S and $\mathcal{D}$ are clear from the context we simply use the notation $disc_g$.

To clarify the above definition, let's consider the case of a classifier g used by the HR staff of a company. The classifier g suggests hiring when $g(\boldsymbol{x}) = 1$ vs. not hiring when $g(\boldsymbol{x}) = 0$. We may wonder whether the classifier favours *men* ($S = 1$) over *women* ($S = 0$). The value of $disc_g$ is large if the ratio of men with a favorable hiring prediction is larger than the ratio of women with a favorable hiring prediction. By minimizing $disc_g$ we can provide a fairer classifier w.r.t. the gender attribute.

2.2 Related Works

Notably, in recent years works identifying and solving bias in machine learning algorithms have progressed. *Pre-processing*, *in-processing*, and *post-processing* approaches have been used to mitigate and quantify bias coming from training data, learning algorithms, or the interaction between the twos.

Algorithms which are identified in the *Pre-processing* category deal with discrimination at the dataset level by altering its distribution to ensure there is no bias against a specific group or individual. This can be achieved by removing the sensitive attribute, re-sampling the data, or changing class labels. One of the well known pre-processing method is massaging [17], which changes the class labels of a subset of carefully selected instances. Another work in this category is re-weighting [5], which assigns different weights to different groups of the dataset to reduce bias. A re-sampling approach in [7] limits the sample size to control discrimination.

In the *In-processing* bias mitigation algorithms, discrimination is accounted during the training phase of the learning algorithm. Strategies in this group take different approaches to discount discrimination by including fairness penalty into the loss function such as in [29], which integrates decision boundary covariance constraint for logistic regression. In [1] regularization terms are added to penalize discrimination in mixed-integer optimization framework of decision tree. Another interesting work is [6], which proposes three approaches for fairness-aware Naïve Bayes classifiers. The approaches are: altering the decision distribution until there is no more discrimination, building a separate model for each sensitive group to remove the correlation between sensitive attribute and class label, and adding latent variable representing unbiased label.

Kamiran *et al.* [18] included a discrimination factor into the information gain splitting criterion of a single decision tree classifier by considering the split of a node under the influence of a sensitive feature, i.e., before a node split happens not only the usual purity w.r.t. to the target label is calculated, but also the purity of the split w.r.t. the sensitive feature. Three alternative splitting criteria are given based on the way discrimination is accounted. The first option is subtracting discrimination gain from accuracy gain, which allows for a split if it is non-discriminatory, second option is an accuracy-discrimination trade-off split where the accuracy gain is divided by discrimination gain to have the final gain value. The third option is adding the accuracy and discrimination gain to decide the best feature to split a node. The authors claim the additive information gain criterion produces a lower discrimination. We also implement this method for the base trees of our forest and evaluate the impact of it to the overall forest discrimination value. Finally, authors propose an additional relabeling of some leaves of the tree so as to further reduce its discrimination degree.

A recent work, called Distributed Fair Random Forest (DFRF) [13] exploits randomly generated decision trees and filters them by their fairness before adding them to the forest. This is achieved through a hyper-parameter fairness constraint, which forces to accept only decision trees with statistical parity below the given threshold. The generation and fairness thresholding of each individual tree can be done in distributed framework that optimizes the trade-of between discrimination and accuracy of the tree before being added to the forest. Furthermore, this algorithm uses randomness constraint to train base trees in which one feature is randomly selected to split a node for building a randomized decision tree.

Post-processing mitigation approaches focus on adjusting the final output of the trained model rather than the underline loss function or training data. The algorithms discussed in [16, 27] aim at achieving same error rates between privileged and unprivileged groups,

[16] uses equalized odd and equalized opportunity to promote features which are more dependent on the target label than the sensitive attribute. While in [27] the proposed algorithm aims to achieve both privileged and unprivileged groups to have the same false negative rate and false positive rate by taking into account a calibrated probability estimates. Another post-processing algorithm called Reject Option based Classification (ROC) [19] takes in to

consideration the decision boundary of classifiers; in a region where uncertainty is high, it gives favorable outcomes to the unprivileged group and unfavorable outcomes to the privileged group to reduce discrimination.

Among the various works, the closest to our proposal is [18], which relabels leaves of a single tree classifier with a small effect on the model accuracy. We borrow from this approach and propose a novel algorithm for enforcing fairness in forests of decision trees.

Table 1: Notation Summary

Symbol	Meaning
$\mathcal{D}$	Dataset
S	Sensitive feature
λ	leaf
Λ	Set of Leaves to be flipped
$disc_{\mathcal{F}}$	Forest discrimination
$disc_{\mathcal{T}}$	Tree discrimination
$accu_{\mathcal{F}}$	Forest accuracy
$\Delta disc_\lambda$	change in discrimination after flipping
$\Delta accu_\lambda$	change on accuracy after flipping
δ	Ratio of change in accuracy and discrimination

3 THE EIFFFEL ALGORITHM

We propose a novel post-processing algorithm named EiFFFeL that, given a forest of decision trees for a binary classification task, modifies the prediction of a carefully chosen set of leaves so as to reduce the forest's discrimination degree. This process is named leaf relabeling, or, since we are focusing on a binary prediction task, *leaf flipping*.

The rationale is to flip the prediction of the leaves that contribute the most to the model discrimination degree so as to make them fair. Recall that the score $disc_{\mathcal{D},S,g}$ adopted to evaluate the model's discrimination depends on the number of privileged/unprivileged instances with a favorable prediction. Therefore, by flipping a leaf label we can increase or decrease the number of instances that contribute to the discrimination score. Note that, while leaf relabeling can be done judiciously so as to reduce discrimination, modifying the leaf predictions determined at training time may reduce the accuracy of the whole forest.

Therefore the goal of EiFFFeL is to find a sweet-spot in the accuracy vs. discrimination trade-off. While leaf relabeling was introduced by [18] for a single tree, we improve such strategy and extend it to a forest of decision trees.

In this work we focus on Random Forests ensembles, which, for their high accuracy and limited bias, are an optimal candidate for building a fair classifier. The approach is however general and we leave to future work the application to other tree ensembles, such as those obtained by bagging and boosting approaches.

The proposed EiFFFel algorithm accepts a user-defined maximum discrimination constraint ϵ and a minimum relative accuracy drop constraint α. Given a forest $\mathcal{F}$, it iteratively modifies the prediction associated with a subset of the leaves of $\mathcal{F}$, until either the desired discrimination ϵ is achieved, or the maximum required accuracy drop α is hit.

Below we first illustrate the *Leaf Scoring* strategy used to find the most discriminative leaves of a tree, and then we illustrate two variants of the EiFFFeL algorithm.

Algorithm 1 SCORE_LEAVES

Input: Decision Tree $\mathcal{T}$
Dataset $\mathcal{D}$
Sensitive feature S
Output: Candidate flipping leaves Λ

1: $\Lambda \leftarrow \emptyset$
2: **for all** $\lambda \in \mathcal{T} \mid \neg\lambda.flipped$ **do**
3: $\quad \Delta accu_\lambda \leftarrow -abs\left(\frac{|\mathcal{D}^\lambda_{y=1}| - |\mathcal{D}^\lambda_{y=0}|}{|\mathcal{D}|}\right)$
4: $\quad \Delta disc_\lambda \leftarrow sign\left(|\mathcal{D}^\lambda_{y=1}| - |\mathcal{D}^\lambda_{y=0}|\right) \cdot \left(\frac{|\mathcal{D}^\lambda_{S=1}|}{|\mathcal{D}_{S=1}|} - \frac{|\mathcal{D}^\lambda_{S=0}|}{|\mathcal{D}_{S=0}|}\right)$
5: $\quad \delta \leftarrow \frac{\Delta disc_\lambda}{\Delta accu_\lambda}$
6: $\quad$ **if** $\delta \geq 0$ **then**
7: $\quad\quad \lambda.score \leftarrow \delta$
8: $\quad\quad \Lambda \leftarrow \Lambda \cup \{\lambda\}$
9: $\quad$ **end if**
10: **end for**
11: **return** Λ

3.1 Leaf Scoring

EiFFFeL borrows from [18] a simple strategy for scoring leaves according to their impact $\Delta accu_\lambda$ and $\Delta disc_\lambda$ on accuracy and discrimination respectively. Then, the ratio δ between the two is used as a score to greedily select the best leaves to be *flipped*.

We proceed as described in Alg. 1. We consider only leaves of the tree that were not flipped during previous iteration of the EiFFFeL algorithm (see subsection below). For those leaves we compute the accuracy and discrimination variation in the case of flipping the leaf prediction. We illustrate shortly the computations below, please refer to [18] for a more detailed description.

The change in accuracy $\Delta accu_\lambda$ clearly depends on the number of instances of $\mathcal{D}$ that fall into the leaf λ denoted with $\mathcal{D}^\lambda$. The training process sets the leaf prediction to the majority class among such instances. Therefore, when flipping the leaf prediction the accuracy may only decrease depending on the instances with label 1 and 0, denoted by $\mathcal{D}^\lambda_{y=1}$ and $\mathcal{D}^\lambda_{y=0}$ respectively. The difference between the size of these two sets results in the accuracy loss as computed in line 3.

The change in discrimination $\Delta disc_\lambda$ depends on the number of privileged and unpriviledged instances that fall in the leaf λ respectively denoted by $\mathcal{D}^\lambda_{S=1}$ and $\mathcal{D}^\lambda_{S=0}$, and on their analogous on the whole dataset $\mathcal{D}_{S=1}$ and $\mathcal{D}_{S=0}$. If the leaf prediction equals 1 (favourable class), then increasing $\mathcal{D}^\lambda_{S=1}$ would increase the discrimination, while increasing $\mathcal{D}^\lambda_{S=0}$ would decrease it. The opposite holds if the prediction of the leaf equals 0 (unfavourable class). As the original leaf prediction depends on the majority of the instances between $\mathcal{D}^\lambda_{y=1}$ and $\mathcal{D}^\lambda_{y=0}$, the sign of their difference is used to correct the above contributions as computed in line 4.

The ratio $\delta = \Delta disc_\lambda / \Delta accu_\lambda$ is positive if the flipping generates a discrimination drop, and it is large if the benefit to discrimination is larger than the harm to accuracy. If the value of δ is positive, then this is stored with the leaf λ, and λ is recorded into the set of candidate leaves Λ. The set Λ is eventually returned and exploited during the iterations of EiFFFeL.

Algorithm 2 EiFFFeL-TF (Tree-based Flipping)

Input: Random Forest classifier $\mathcal{F}$
Discrimination Constraint $\epsilon \in [0, 1]$
Accuracy Constraint $\alpha \in [0, 1]$
Training Dataset $\mathcal{D}$
Sensitive feature S
Output: Fair Random Forest $\mathcal{F}$

for all $\mathcal{T} \in \mathcal{F}$ **do**
  $\mathcal{T}.flipped \leftarrow false$
  for all $\lambda \in \mathcal{T}$ **do**
    $\lambda.flipped \leftarrow false$
  end for
end for
$accu^*_{\mathcal{F}} \leftarrow \frac{|\mathcal{D}_{y=1 \wedge \mathcal{F}(\boldsymbol{x})=1}| + |\mathcal{D}_{y=0 \wedge \mathcal{F}(\boldsymbol{x})=0}|}{|\mathcal{D}|}$
$\Delta accu_{\mathcal{F}} \leftarrow 0$
while $|\{\mathcal{T} \in \mathcal{F} \mid \neg \mathcal{T}.flipped\}| > 0 \wedge$
    $disc_{\mathcal{D},S,\mathcal{F}} > \epsilon \ \wedge \ \Delta accu_{\mathcal{F}} < \alpha$ **do**
  $\mathcal{T}^\dagger \leftarrow \text{argmax}_{\mathcal{T} \in \mathcal{F}} \ disc_{\mathcal{D},S,\mathcal{T}}$
  $\Lambda \leftarrow$ Score_Leaves($\mathcal{T}^\dagger, \mathcal{D}, S$)
  if $\Lambda \neq \emptyset$ **then**
    for all $\lambda \in \Lambda$ **do**
      $\lambda.pred = 1 - \lambda.pred$
    end for
  end if
  $\mathcal{T}^\dagger.flipped = true$
  $accu_{\mathcal{F}} \leftarrow \frac{|\mathcal{D}_{y=1 \wedge \mathcal{F}(\boldsymbol{x})=1}| + |\mathcal{D}_{y=0 \wedge \mathcal{F}(\boldsymbol{x})=0}|}{|\mathcal{D}|}$
  $\Delta accu_{\mathcal{F}} \leftarrow accu^*_{\mathcal{F}} - accu_{\mathcal{F}}$
end while
return $\mathcal{F}$

Algorithm 3 EiFFFeL-LF (Leaf-based Flipping)

Input: Random Forest classifier $\mathcal{F}$
Discrimination Constraint $\epsilon \in [0, 1]$
Accuracy Constraint $\alpha \in [0, 1]$
Training Dataset $\mathcal{D}$
Sensitive feature S
Output: Fair Random Forest $\mathcal{F}$

for all $\mathcal{T} \in \mathcal{F}$ **do**
  $\mathcal{T}.flipped \leftarrow false$
  for all $\lambda \in \mathcal{T}$ **do**
    $\lambda.flipped \leftarrow false$
  end for
end for
$accu^*_{\mathcal{F}} \leftarrow \frac{|\mathcal{D}_{y=1 \wedge \mathcal{F}(\boldsymbol{x})=1}| + |\mathcal{D}_{y=0 \wedge \mathcal{F}(\boldsymbol{x})=0}|}{|\mathcal{D}|}$
$\Delta accu_{\mathcal{F}} \leftarrow 0$
while $|\{\mathcal{T} \in \mathcal{F} \mid \neg \mathcal{T}.flipped\}| > 0 \wedge$
    $disc_{\mathcal{D},S,\mathcal{F}} > \epsilon \ \wedge \ \Delta accu_{\mathcal{F}} < \alpha$ **do**
  $\mathcal{T}^\dagger \leftarrow \text{argmax}_{\mathcal{T} \in \mathcal{F}} \ disc_{\mathcal{D},S,\mathcal{T}}$
  $\Lambda \leftarrow$ Score_Leaves($\mathcal{T}^\dagger, \mathcal{D}, S$)
  if $\Lambda = \emptyset$ **then**
    $\mathcal{T}.flipped \leftarrow true$
  else
    $\lambda^\dagger \leftarrow \text{argmax}_{\lambda \in \Lambda} \ \lambda.score$
    $\lambda^\dagger.flipped = true$
    $\lambda^\dagger.pred = 1 - \lambda.pred$
    $accu_{\mathcal{F}} \leftarrow \frac{|\mathcal{D}_{y=1 \wedge \mathcal{F}(\boldsymbol{x})=1}| + |\mathcal{D}_{y=0 \wedge \mathcal{F}(\boldsymbol{x})=0}|}{|\mathcal{D}|}$
    $\Delta accu_{\mathcal{F}} \leftarrow accu^*_{\mathcal{F}} - accu_{\mathcal{F}}$
  end if
end while
return $\mathcal{F}$

3.2 EiFFFeL Leaf Flipping Strategies

By exploiting the scoring technique discussed before, we propose two strategies to choose which trees and which leaves in those trees to flip.

The first strategy, named *Tree-based Flipping*, is illustrated in Alg. 2. During each iteration of EiFFFeL, the tree $\mathcal{T}^\dagger$ with the largest discrimination degree is greedily selected: this is the best tree to be attacked in order to significantly reduce the discrimination of the full forest. Then, we use the previous scoring technique to find the set of leaves Λ in $\mathcal{T}^\dagger$ that should be relabeled. If Λ is not empty, the predictions $\lambda.pred$ of such leaves will be flipped. Then, the whole tree is marked as already flipped. The selection is repeated by considering only the remaining non-flipped trees. The algorithm ends when all trees have been flipped, or when the desired discrimination ϵ is achieved, or when tolerated accuracy drop α is met. Note that the accuracy drop is computed by comparing the accuracy of the original forest with the accuracy of the current forest after the flipping step.

Such tree-based strategy might be too aggressive, as it immediately flips all the candidate leaves of the selected tree. Indeed, only a few leaves may be sufficient to meet our discrimination and accuracy requirements. Therefore we propose a second strategy, named *Leaf-Based Flipping*, illustrated in Alg 3. As in the former strategy, we first select the tree $\mathcal{T}^\dagger$ with the largest discrimination. Then we use the leaf scoring technique to find a set of candidate leaves from $\mathcal{T}^\dagger$. If such set is empty, e.g., because they were already flipped or they cannot improve the discrimination, the full tree is marked as flipped and the procedure is repeated on the remaining non-flipped trees. Otherwise, the leaf with the largest score $\lambda^\dagger$ is selected, marked as flipped, while its prediction is inverted. The process is repeated until all trees have been flipped, or the desired discrimination ϵ is achieved, or the tolerated accuracy drop α is met.

We argued that the Leaf-based approach exploits a more fine-grained tuning of the given forest, and therefore it can achieve the desired accuracy with a smaller set of alterations. Indeed, reducing the flips applied to the forest provides a larger accuracy.

4 EXPERIMENTAL EVALUATION

4.1 Datasets.

We use datasets publicly available, widely used in fairness literature, concerning binary classification. We pre-process them using one-hot encoding for categorical features, binary encoding of sensitive feature, and removing of instances containing missing values. Moreover, we use an 80/20 training/test split.

- *Adult*: The Adult UCI income dataset [10] contains 14 demographic attributes of more than 45,000 individuals, together with class labels which states whether their income is higher than $50K or not. As sensitive attribute, we use the *gender* encoded as a binary attribute 1/0 for male/female respectively.

Table 2: Comparison of accuracy reduction and discrimination decrease on Adult dataset with respect to baseline accuracy of 0.85 and discrimination 0.2. Along with ΔAccu and ΔDisc, we also report (within parentheses) the final accuracy and discrimination values obtained.

			DFRF		EOP		EiFFFeL-TF		EiFFFeL-LF		EiFFFeL-TF*		EiFFFeL-LF*	
			ΔAccu↓	ΔDisc ↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑
Adult	ϵ	0.01	7(0.78)	18(0.02)	2(0.83)	7(0.13)	4(0.81)	19(0.01)	4(0.81)	**20(0)**	6(0.79)	15(0.05)	3(0.82)	17(0.03)
		0.05	3(0.82)	13(0.07)			3(0.82)	**16(0.04)**	2(0.83)	15(0.05)	6(0.79)	16(0.04)	3(0.82)	**16(0.04)**
		0.10	4(0.81)	15(0.05)			2(0.83)	12(0.08)	1(0.84)	**12(0.08)**	1(0.84)	**12(0.08)**	2(0.83)	10(0.1)
		0.15	2(0.83)	10(0.1)			0(0.85)	8(0.12)	0(0.85)	**9(0.11)**	0(0.85)	7(0.13)	0(0.85)	7(0.13)

Table 3: Comparison of accuracy reduction and discrimination decrease on Bank dataset with respect to baseline accuracy of 0.82 and discrimination 0.18. Along with ΔAccu and ΔDisc, we also report (within parentheses) the final accuracy and discrimination values obtained.

			DFRF		EOP		EiFFFeL-TF		EiFFFeL-LF		EiFFFeL-TF*		EiFFFeL-LF*	
			ΔAccu↓	ΔDisc ↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑
Bank	ϵ	0.01	9(0.73)	13(0.05)	0(0.82)	14(0.04)	7(0.75)	17(0.01)	10(0.72)	15(0.03)	8(0.74)	14(0.04)	5(0.77)	10(0.08)
		0.05	4(0.78)	11(0.07)			3(0.79)	13(0.05)	8(0.74)	14(0.04)	8(0.74)	13(0.05)	5(0.77)	13(0.05)
		0.10	4(0.78)	6(0.12)			2(0.80)	10(0.08)	1(0.81)	7(0.11)	7(0.75)	9(0.09)	4(0.78)	8(0.10)
		0.15	4(0.78)	9(0.09)			0(0.82)	4(0.14)	0(0.82)	4(0.14)	6(0.76)	5(0.13)	2(0.80)	5(0.13)

Table 4: Comparison of accuracy reduction and discrimination decrease on Compas dataset with respect to baseline accuracy of 0.69 and discrimination 0.3. Along with ΔAccu and ΔDisc, we also report (within parentheses) the final accuracy and discrimination values obtained.

			DFRF		EOP		EiFFFeL-TF		EiFFFeL-LF		EiFFFeL-TF*		EiFFFeL-LF*	
			ΔAccu↓	ΔDisc ↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑	ΔAccu↓	ΔDisc↑
COMPAS	ϵ	0.01	11(0.58)	28(0.02)	4(0.65)	5(0.25)	25(0.44)	29(0.01)	5(0.64)	26(0.04)	9(0.60)	29(0.01)	1(0.68)	7(0.23)
		0.05	5(0.64)	13(0.17)			12(0.57)	28(0.02)	5(0.64)	22(0.08)	9(0.60)	28(0.02)	1(0.68)	7(0.23)
		0.10	4(0.65)	7(0.23)			7(0.62)	21(0.09)	5(0.64)	21(0.09)	1(0.68)	21(0.09)	1(0.68)	7(0.23)
		0.15	2(0.67)	6(0.24)			1(0.68)	19(0.11)	2(0.67)	15(0.15)	0(0.69)	16(0.14)	1(0.68)	7(0.23)

- *COMPAS*: The COMPAS dataset [2] contains data collected on the use of the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) risk assessment tool. It contains 13 attributes of more than 7,000 convicted criminals, with class labels that state whether or not the individual reoffend within 2 years of her/his most recent crime. We use *race* as sensitive attribute encoded as a binary attribute 1/0 for Others/African-American, respectively.
- *Bank*: Bank marketing dataset [22] contains 16 features about 45,211 clients of direct marketing campaigns of a Portuguese banking institution. The goal is to predict whether the client will subscribe or not to a term deposit. We consider the *age* as sensitive attribute, encoded as a binary attribute 1/0, indicating whether the client's age is ≥25 or <25, respectively.

4.2 Experimental Setup.

We apply our proposed EiFFFeL algorithm over a Random Forest classifier with/without the fair splitting of nodes for individual base trees, and evaluate the performance of the algorithms in terms of model accuracy and discrimination over the three datasets mentioned above.

We compare our results against a DFRF classifier (*Distributed fair random forest*) [13], which only includes fair decision trees within the forest. The setting of hyper-parameters of DFRF are the same as the one described in the original work. We use fair split and sensitive feature as hyper-parameters, along with tree number and maximum tree depth. Additionally, we also compare our results against EOP (*Equalized Odds Post-processing*) [16, 27], a random forest classifier with the same number of base estimators and maximum depth as ours. After training and achieving the desired equalized odd we score the discrimination in the same approach we used for our experiments.

In conclusion, the comparisons of accuracy and discrimination values are among the following methods:

- DFRF [13],[1] which adds base trees to the forest only if they are fair;
- EOP [16, 27],[2] which adopts a post-processing method based on achieving equalized odds requiring the privileged and unprivileged groups to have the same false negative rate and same false positive rate;
- our implementations of EiFFFeL-TF and EiFFFeL-LF algorithms, whose post-processing is applied to a plain Random Forest of trees;

[1] https://github.com/pjlake98/Distributed-Fair-Random-Forest
[2] https://github.com/Trusted-AI/AIF360/blob/master/aif360/algorithms/postprocessing/calibrated_eq_odds_postprocessing.py

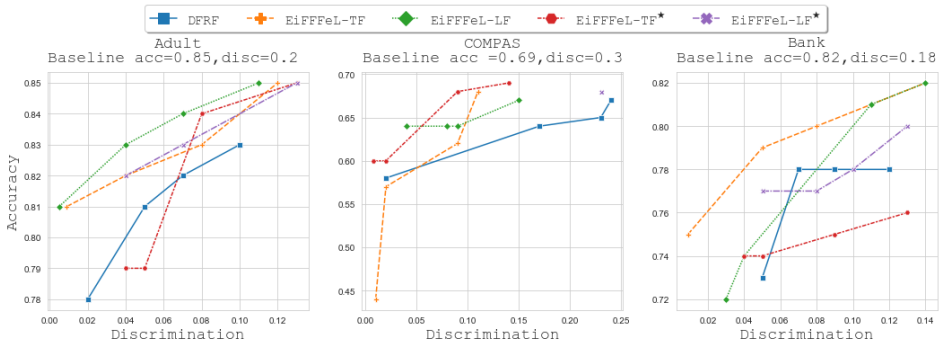

Figure 1: Accuracy vs. discrimination scores after relabeling for constraints $\epsilon = 0.01, 0.05, 0.1, 0.15$.

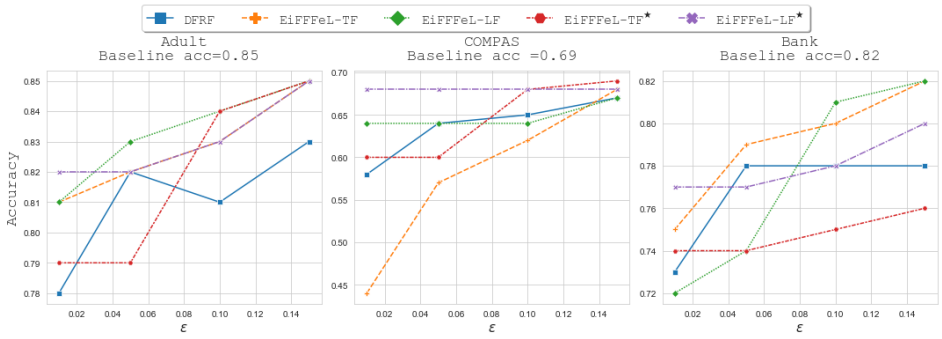

Figure 2: Accuracy of the model as a function the ϵ constraint.

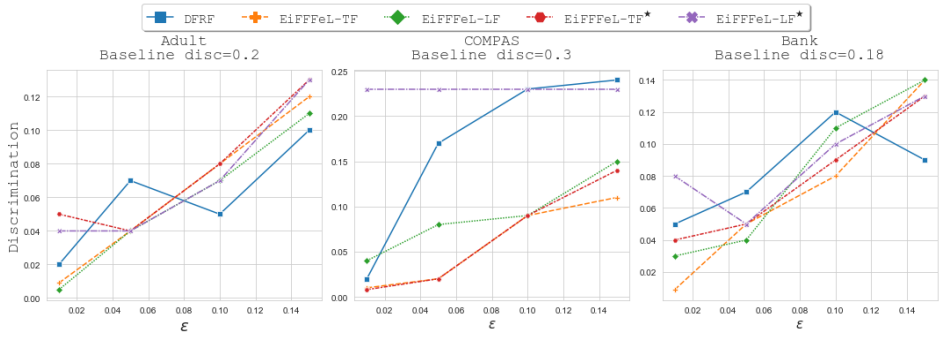

Figure 3: Discrimination scores as a function of the ϵ constraint.

- the same post-processing techniques of EiFFFeL-TF and EiFFFeL-LF applied on top of a random forest with discrimination aware base trees [18]. These versions are denoted by EiFFFeL-TF* and EiFFFeL-LF*.

Finally, the *baseline accuracy* and *discrimination* used to compare the various methods are the ones obtained by a plain Random Forest of trees, trained on the three datasets through the scikit-learn algorithm Random Forest Classifier[3]. The various EiFFFeL methods are applied to the same baseline Random Forest.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

4.3 Results.

Tables 2), 3), and 4) compare the decreases in accuracy and discrimination, obtained by the different algorithms, on the three datasets with respect to the baseline results obtained by plain Random Forest models.

Recall that increasing ϵ, we reduce the space for improving discrimination, and as a side effect, we preserves the baseline accuracy. Indeed, in these experiments the accuracy constraint α was set to 1, so that there are no limits in the possible accuracy reduction ΔAccu. This allows us to compare our methods against DFRF and EOP, which do not have this α constraint. Indeed, EOP is completely parameter free, and does not support neither α nor ϵ.

In more details, Tables 2), 3), and 4) report, for different values of ϵ in the set $\{0.01, 0.05, 0.10, 0.15\}$, the ΔAccu and ΔDisc values obtained by the different algorithms, where ΔAccu and ΔDisc indicate the *absolute difference* in accuracy and discrimination w.r.t. the baselines. Indeed, we express these Δ absolute differences in points/hundredths (each point corresponds to 1/100). Note that while greater values of ΔDisc are better, greater values of ΔAccu are worse, so a trade-off is needed. In addition, besides the absolute Δ values, we also report (within parentheses) the final values for accuracy and discrimination score obtained by the various techniques.

For example, for the *Adult* dataset (Table 2) and $\epsilon = 0.01$, EiFFFeL-TF can reach a very low discrimination score of 0.01, by only losing 4 points in accuracy (from 0.85 of the baseline to 0.81). In comparison, the best results we can obtain with DFRF in terms of discrimination is a score of 0.02, by losing 7 points in accuracy (from 0.85 of the baseline to 0.78). Overall, our algorithms are capable of reducing discrimination better than DFRF while maintaining the same accuracy. Also EOP does not work well, as the best discrimination score is only 0.13, by losing 2 points in accuracy. In addition, using $\epsilon = 0.15$ for EiFFFeL-TF and EiFFFeL-LF (also EiFFFeL-TF* and EiFFFeL-LF*), we can decrease the baseline discrimination of about $7 - 9$ points, by keeping the same accuracy of the baseline.

Results for the Bank dataset (Table 3) shows that EiFFFeL-TF can reach for $\epsilon = 0.01$ the desired discrimination score, but losing 7 points in accuracy (from 0.82 to 0.75), whereas DFRF has worse discrimination score of 0.05 and a worse accuracy of 0.73. EOP does not lose any accuracy for lowering the discrimination score by 14 points to 0.04.

Finally, considering the results obtained for the COMPAS dataset (Table 4), we observe in some cases DFRF works pretty well, but always one of our algorithms gets better results. For example, for $\epsilon = 0.01$, the best discrimination score of 0.01 is obtained by EiFFFeL-TF*, by only losing 9 points in accuracy, against the 11 points lost by DFRF with a discrimination score of 0.02.

Figures 1, 2, and 3 report the same data of the above tables, where we varied the discrimination constraint $\epsilon = \{0.01, 0.05, 0.1, 0.5\}$, with no constraints on accuracy. The results obtained by EOP are not plotted, as its results are always worse than the competitors and do not vary with ϵ.

Specifically, Figure 1 reports results for the three datasets, and aims at showing the tradeoff of accuracy vs. discrimination when we vary ϵ. Recall that we are interested in achieving low discrimination and high accuracy, and thus the best tradeoff corresponds to points of curves falling in the top-left quadrant.

First, we highlight that DRFR performs poorly on most settings compared to the proposed EIFFFeL variants. On the Adult dataset, EIFFFeL-LF dominates the other algorithms for all values of ϵ and achieves the desired or better discrimination with the largest accuracy. To appreciate the strict relationships between of the setting of ϵ and the discrimination/accuracy obtained, the reader can refer to the other two Figures 1 and 2.

Returning to Figure 1, the COMPAS EIFFFeL-LF provides the best performance together with EIFFFeL-TF*. This is the only dataset where EIFFFeL-TF* provides interesting performance, and thus the discrimination aware splitting at training time provides some benefits. We also highlight that when using $\epsilon = 0.15$ (see Figure 3) the algorithm DFRF only gets a discrimination score of 0.25. Note that EIFFFeL-LF* is not able to provide better performance when varying ϵ, thus resulting in a constant curve.

Finally, on the Bank Dataset, EIFFFeL-TF and EIFFFeL-LF achieve the best results, with an advantage for EIFFFeL-TF for smaller values of ϵ. Finally, the results show how we can obtain the desired discrimination degree with a limited drop in accuracy. Overall, the proposed EIFFFEL algorithm outperforms the competitor DFRF, and, on average, it is advisable to avoid the discrimination aware node splitting. We believe that working only at post-processing allows us to exploit a richer set of trees grown, by exploring a larger and unconstrained search space.

The effect of varying the discrimination constraint ϵ without constraining accuracy can be observed in Figure 2, where we discover that lower discrimination is achieved with large accuracy reduction. This is due to the fact that a small discrimination threshold allows our flipping strategies to force the change of many leaves, thus changing more the classification decision regions, with a final lower accuracy. However our approach of selecting potential leaves to relabel seems better than training random forest with only fair trees. In addition, training and then rejecting trees (because they are not fair) makes longer the training of the forest, particularly when we fail often in finding fair trees.

Finally, Figure 3 contrasts the discrimination measured on the test set against the desired discrimination constraint ϵ. Clearly, the twos do not always match. In particular, DFRF has an unstable behaviour, meaning that filtering the tree to be added to the forest is not the best option. Conversely, EiFFFeL-TF and EiFFFeL-LF provide a much more stable behaviour.

We also discuss the results of other experiments, aiming to evaluate the effects of different values for the α constraints. Note that only the EiFFFeL algorithms support the α parameter, so we cannot reports any results for the competitors DFRF and EOP. Specifically, Table 5 reports results relative to the Adult dataset, where, for a fixed $\epsilon = 0.01$, we vary the α constraint over the expected accuracy, with values ranging in the set {0.01,0.02,0.03,0.05}. For each α value, we show in bold the best results in terms of discrimination score. We observe that the accuracy constraint α has an indirect impact on the final discrimination score obtained. Using EiFFFeL-LF with $\alpha = 0.01$, the loss in accuracy is 1 point as expected, while the baseline discrimination score decreases by more than half (from 0.2 to 0.08). Furthermore, as the α value increases, discrimination score decreases further. With $\alpha = 0.05$, EiFFFeL-LF is able to reduce by 4 points the final accuracy, by also achieving a discrimination score of 0, thus showing the power of our method in achieving a very good trade-off between accuracy and discrimination.

5 CONCLUSION

In this work we deal with fairness in machine learning, and specifically in binary classifiers trained by a Random Forest algorithm. We are interested in group fairness, so as to mitigate the effect of bias against specific groups, which may comes from biased training datasets or algorithm design.

We develop EiFFFeL, a novel post-process approach, which maintains good predictive performance of the trained model with a low

Table 5: Accuracy and discrimination scores on the Adult dataset for $\epsilon = 0.01$ and $\alpha = 0.01, 0.02, 0.03, 0.05$. The baseline accuracy and discrimination score are 0.85 and 0.2, respectively.

				EiFFFeL-TF		EiFFFeL-LF		EiFFFeL-TF*		EiFFFeL-LF*	
				Accu	Disc	Accu	Disc	Accu	Disc	Accu	Disc
Adult	ϵ=0.01	α	0.01	0.83	0.09	0.84	**0.08**	0.84	0.10	0.84	0.11
			0.02	0.83	0.09	0.83	**0.06**	0.83	0.10	0.83	0.07
			0.03	0.82	**0.04**	0.82	0.05	0.82	0.07	0.82	**0.04**
			0.05	0.81	0.01	0.81	**0.00**	0.80	0.08	0.82	0.03

discrimination score. Our approaches flips the label of selected leaf (or leaves) of base trees in a random forest by using two algorithms: (i) an aggressive tree-based approach, which flips all candidate leaves of a tree, and (ii) a leaf-based strategy which only flips the label of the most discriminative leaf of a tree. Both strategies are implemented by considering accuracy and discrimination constraints. Indeed, the constraints are used to control the minimum accuracy decrease we can tolerate in order to achieve the desired discrimination value. In addition, we have tested the impact of incorporating discrimination aware node split strategies for base trees of the forest, by adding discrimination gain value in their node splitting criterion [18].

By using three publicly available datasets, our experimental results show that effective non-discriminative models can be obtained, while keeping a strict control over both accuracy and discrimination level. Compared to state-of-the-art methods, which adopt both in-process and post-process bias mitigation approaches, EiFFFeL resulted to produce the most accurate models that also exhibit the best levels of fairness.

As part of the future work, we plan to extend our methods by studying the effect of multiple sensitive features in relation to discrimination and accuracy, by also extending our work to other tree ensemble learning methods.

REFERENCES

[1] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. 2019. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, California, USA, 1418–1426.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. ProPublica (2016).

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

[4] Ajay Byanjankar, Markku Heikkilä, and Jozsef Mezei. 2015. Predicting credit risk in peer-to-peer lending: A neural network approach. In *2015 IEEE symposium series on computational intelligence*. IEEE, Cape Town, South Africa, 719–725.

[5] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE, FLorida ,USA, 13–18.

[6] Toon Calders and Sicco Verwer. 2010. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery* 21, 2 (2010), 277–292.

[7] Flavio P Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates, NY,USA, 3995–4004.

[8] Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. 2017. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data* 5, 2 (2017), 120–134.

[9] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine* 24, 9 (2018), 1342–1350.

[10] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

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

[12] Virginia Eubanks. 2018. *Automating inequality: How high-tech tools profile, police, and punish the poor*. St. Martin's Press, NY,USA.

[13] James Fantin. 2020. *A Distributed Fair Random Forest*. Ph.D. Dissertation. University of Wyoming.

[14] Vincent Grari, Boris Ruf, Sylvain Lamprier, and Marcin Detyniecki. 2020. Achieving fairness with decision trees: An adversarial approach. *Data Science and Engineering* 5 (2020), 99–110.

[15] Sara Hajian, Josep Domingo-Ferrer, Anna Monreale, Dino Pedreschi, and Fosca Giannotti. 2015. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery* 29, 6 (2015), 1733–1782.

[16] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016), 3315–3323.

[17] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *2009 2nd international conference on computer, control and communication*. IEEE, Karachi, Pakistan, 1–6.

[18] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*. IEEE, Australia, 869–874.

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

[20] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. 2015. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal* 13 (2015), 8–17.

[21] Milad Malekipirbazari and Vural Aksakalli. 2015. Risk assessment in social lending via random forests. *Expert Systems with Applications* 42, 10 (2015), 4621–4631.

[22] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.

[23] AI Now. 2018. Litigating algorithms: Challenging government use of algorithmic decision systems.

[24] Luca Oneto, Michele Donini, Andreas Maurer, and Massimiliano Pontil. 2020. Learning fair and transferable representations. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, Sydney,Australia, 30–39.

[25] Luca Oneto, Anna Siri, Gianvittorio Luria, and Davide Anguita. 2017. Dropout Prediction at University of Genoa: a Privacy Preserving Data Driven Approach.. In *ESANN 2017 proceedings*. ESANN, Bruges, //.

[26] Zacharoula K Papamitsiou and Anastasios A Economides. 2014. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *J. Educ. Technol. Soc.* 17, 4 (2014), 49–64.

[27] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On Fairness and Calibration. In *NIPS*, Vol. 30. Curran Associates, Inc., CA,USA.

[28] Edward Raff, Jared Sylvester, and Steven Mills. 2018. Fair forests: Regularized tree induction to minimize model bias. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, New Orleans, USA, 243–250.

[29] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*. PMLR, FL, USA, 962–970.

[30] Wenbin Zhang and Eirini Ntoutsi. 2019. FAHT: An Adaptive Fairness-aware Decision Tree Classifier. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, Macao, 1480–1486.

[31] Wenbin Zhang, Xuejiao Tang, and Jianwu Wang. 2019. On fairness-aware learning for non-discriminative decision-making. In *2019 International Conference on Data*

Mining Workshops (ICDMW). IEEE, Beijing, China, 1072–1079.
[32] Indre Zliobaite. 2015. A survey on measuring indirect discrimination in machine learning.