Fairness-Guided Pruning of Decision Trees

Meryem Bagriacik
University of Kent
School of Computing
Canterbury, United Kingdom
mb2076@kent.ac.uk

Fernando Otero
University of Kent
School of Computing
Canterbury, United Kingdom
f.e.b.otero@kent.ac.uk

Abstract

Decision tree learning is a popular machine learning technique, in particular for applications where the interpretability of the predictions is crucial—such as applications in health and financial domains. When dealing with a large dataset, decision tree algorithms potentially generate a complex model that overfits the training data. In such cases, it becomes challenging for decision trees to maintain an interpretable structure while also identifying potential biases in predictions, raising concerns about poor performance on unseen data. To address this issue, decision trees can be pruned to reduce their size, as a result enhancing interpretability and improving predictive accuracy on new instances through the use of simplified models. However, traditional pruning methods typically focus solely on predictive accuracy, which may inadvertently increase model bias and negatively affect fairness. Moreover, current post-processing fairness techniques often aim to reduce discrimination by modifying the tree's labels without considering the complexity inherent in large trees. To address these challenges, we propose a novel fairness-guided pruning strategy for decision trees that improves both fairness and interpretability. Computational experiments comparing our proposed strategy with existing methods demonstrate that our fairness-guided pruning achieves a good accuracy-fairness trade-off overall: small reductions in predictive accuracy are associated with improvements in fairness while simplifying the decision tree structure at the same time.

CCS Concepts

• **Computing methodologies** → *Classification and regression trees*;

Keywords

Fairness, Pruning, Decision Trees, Interpretability

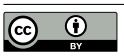
ACM Reference Format:
Meryem Bagriacik and Fernando Otero. 2025. Fairness-Guided Pruning of Decision Trees. In *The 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT '25), June 23–26, 2025, Athens, Greece.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3715275.3732117

1 INTRODUCTION

Machine learning algorithms have become integral components of decision-making systems, particularly in critical domains such as health, law, finance, and education. These algorithms make use of large datasets to create accurate models. However, in real-world applications, many widely used machine learning algorithms are known to create black-box (opaque) models that lack any explanation or interpretation regarding the rationale for their predictions [16]. Consequently, models like Deep Neural Networks (DNN) or, in some instances, decision tree models may produce results that are challenging to interpret. In crucial areas such as medical diagnoses, the complexity of these models makes it difficult to comprehend the logic behind the outcomes, leading to concerns about trustworthiness.

A typical decision tree is constructed through an iterative top-down process, selecting the best attribute to label an internal node of the tree. Although it is a widely used comprehensible classifier, particularly when dealing with large and imbalanced datasets, decision tree algorithms potentially generate a large and complex model that overfits the training data [36]— the depth and complex structure of a decision tree can render it incomprehensible, as highlighted in [35]. In such a case, the decision tree model performs well on training data but gives poor performance on unseen instances. To address the problem of overfitting, the simplification of decision trees by post-pruning has been proposed [37]. By replacing non-significant branches by leaves, post-pruning can efficiently reduce the depth of the tree, thereby potentially leading to improvements in the interpretability of decision trees [13]. Nevertheless, as [32] points out, decision trees may still lack essential features such as fairness and unbiased explainability, even after traditional pruning strategies are applied. Existing pruning approaches focus mainly on reducing tree complexity with the aim of improving predictive accuracy on new instances. However, simplifying models through pruning does not necessarily eliminate discriminatory predictions, as complex decision tree models may inherit biases due to factors like unrepresentative datasets, oversimplified features, or missing values.

Current research on post-processing strategies to address fairness in decision trees focuses primarily on flipping [1] or relabelling [22] of the tree leaves, without considering their complexity. Furthermore, most of the previous works fall under the in-processing category [3, 5, 10, 20]. An exception is the work presented in [18], where a post-pruning procedure has been included as a mutation operator in the multi-objective search-based post-processing method for binary decision trees. In this paper, we are particularly interested in the post-processing stage, proposing a new pruning strategy named *fairness-guided pruning*. The proposed approach investigates the optimization of decision tree models generated following the same strategy of the well-known C4.5 algorithm by combining fairness constraints and error-rate measurements during a post-pruning stage. The end result is to ensure an improvement in group



FAccT '25, Athens, Greece

ACM ISBN 979-8-4007-1482-5/25/06
https://doi.org/10.1145/3715275.3732117

fairness without negative effects on prediction performance, reducing the complexity of the decision tree to offer better explainability and without requiring significant computational resources. By setting up experiments on several datasets, our strategies achieve a balance between accuracy and fairness.

The rest of the paper is set up as follows. Section 2 discusses literature review on the strategies to cope with fairness, and the proposed method is presented in Section 3. Section 4 presents the datasets, experimental results and comparison with other methods. Finally, Section 5 presents the conclusion and future research directions.

2 LITERATURE REVIEW

The approaches to enhance fairness within machine learning applications fall under the three main strategies to inclusively mitigate unfairness and bias in machine learning. *Pre-processing* strategy involves modifying and transforming the dataset to eliminate existing biases, thereby creating unbiased data distribution for the classifier [49]. *In-processing* integrates fairness measures into the model's creation [41], rather than solely focusing on maximizing accuracy, thereby modifying the model itself. *Post-processing* makes adjustments or modifications on predictions to reduce discrimination against unprivileged groups within sensitive features after the model has been constructed using training data [15].

In the context of fair trees, most existing works fall under the in-processing approach to reduce or remove bias during the training procedure, while only a few studies adopt a post-processing approach [3, 20, 22]. One pioneering study [22] introduced discrimination-aware decision tree learning, fundamentally modifying the model based on fairness notions. Subsequently, many studies proposed modifications to the splitting procedure of decision tree learning—particularly in CART—to satisfy fairness constraints [33]. By incorporating MIO into the learning process, fairness constraints are integrated to address the accuracy–fairness trade-off [3, 20, 42]. Further works employed adversarial training of decision trees [39], dynamic programming formulation [40], and designed human-readable logical formula [9] to verify fairness constraints in decision tree classifiers. Rather than applying pre-processing methods for inadequate representation of target populations in the training data, [4] proposed the Domain Adaptive Decision Tree (DADT) to correct them during training. In addition to approaches based on static datasets, some methods aim to enforce fairness in decision trees for streaming data [46, 48, 49]. On fair ensembles of random forests, [19, 31, 38, 47] introduced various approaches for fair induction of CART binary trees in random forests. Finally, [5, 10] integrated multiple fairness metrics into the training mechanism of decision trees.

Post-processing is computationally effective, as it does not require any alterations to the original algorithm [15]. In decision trees, post-processing of fairness has been employed to enhance fairness without considering simplification of the complex trees, resulting in a lack of interpretability. One of the earliest works in this area, [22], introduced a relabelling technique in binary trees inspired by the Knapsack problem, with the aim of further decreasing discrimination with minimal effects on accuracy. Another post-processing strategy, proposed by [17], seeks to satisfy equalized odds and equal opportunity after a classifier is trained; it relies on sensitive attributes and (potentially) discriminatory predictions to build a labelled dataset for enforcing non-discriminatory outcomes while preserving accuracy. Similarly, [3] proposed improving interpretability by incorporating linear leafing and branching after creating a fair decision tree. A related post-processing approach, EiFFFeL [1], employs a leaf flipping technique to alter predictions based on group fairness measures. In this method, a ratio assesses the accuracy–fairness trade-off, guiding whether to flip a leaf if it yields a net improvement in fairness with minimal accuracy loss. To address fairness during post-processing, [25] developed Fairness-Aware Decision Tree Editing (FADE). FADE uses Mixed-Integer Linear Optimization (MILO) to minimize discrepancies between the edited and the original trees under demographic parity constraints. By enforcing threshold-based fairness requirements, FADE ensures that the final model remains close to the initial tree while meeting fairness objectives.

A recent study [29] introduced Fair Feature Importance Score in Decision Trees (FairTreeFIS), which measures the contribution of individual features to unfairness without compromising predictive performance. The approach specifically addresses demographic parity and equality of opportunity. To enhance group fairness in the AdaBoost classifier, [44] proposed incorporating group fairness constraints during the post-pruning stage of the boosting process, executed from the bottom up. When these constraints are satisfied, the final ensemble—originally constructed with a fairness penalty—is pruned to further improve group fairness and overall classifier performance. Unlike deterministic post-processing approaches that enhance fairness through methods such as flipping [1] and relabeling [22], [18] proposed a multi-objective search-based software repair strategy designed to simultaneously improve the fairness and accuracy of classification models. In this method, the binary classification model is mutated in the post-processing phase, thereby achieving concurrent gains in both group fairness and predictive performance. Specifically, pruning is employed as the mutation mechanism for binary decision trees to optimize these objectives.

In the context of post-processing fairness in decision trees, most existing methods that aim to improve fairness do not incorporate an actual pruning procedure; instead, they focus predominantly on transforming prediction outcomes in binary decision trees. Although [18] does include a pruning operation for binary decision trees, it increases the computational cost by running 2,500 iterations over 30 independent runs. In contrast, the concept of pruning—introduced by Quinlan [36] and subsequently refined (e.g., [37]) was designed to produce simpler trees, mitigate overfitting, and do so without incurring a high computational cost. Traditional pruning strategies rely solely on minimizing error, aiming to enhance accuracy while reducing tree size. To the best of our knowledge, there remains a gap in integrating and designing pruning strategies to simultaneously reduce discrimination and address overfitting by simplifying complex trees. Doing so could not only boost interpretability but also reduce the risk that a model's predictions depend unduly on potentially biased features in the training data. Our approach addresses this gap by incorporating fairness measures into the pruning steps while retaining the conventional, error-based pruning strategy and the standard building process

of C4.5, all without requiring pre-processing. This dual focus establishes a balance between fairness and accuracy in the resulting model. Additionally, our approach follows a deterministic approach, unlike the search-based one in [18], and has an ability to handle complex decision trees that have multi-valued attributes without significantly increasing the computational resources. Finally, we provide a systematic way to adjust the fairness threshold, offering users flexibility in determining how much pruning (edits) is performed.

3 PROPOSED METHOD

In this section, we introduce the proposed fairness-guided post-pruning strategies for C4.5 trees. Traditional pruning methods typically prioritize predictive accuracy while overlooking fairness considerations. Moreover, most existing fairness-oriented post-processing techniques for decision trees focus on fairness alone and do not incorporate pruning to simplify complex models. Our proposed strategies address this gap by integrating a fairness measure into the pruning process. Specifically, subtrees are pruned by substituting them with their most common branch or leaf, seeking to simultaneously improve both fairness and error rate. This approach balances predictive accuracy with fairness and can also reduce the overall complexity of the trees.

First, we introduce a fairness metric for evaluating individual (sub)trees. Next, we describe how this metric is employed during post-pruning to guide the decision of whether to prune a subtree.

3.1 Measuring Fairness

Let's assume $D = \{A_j, C\}_{j \in J}$, where J is the total number of predictor features, A_j represents the j-th feature and C is the class feature. For the purpose of this work, the domain of C is binary $\{+, -\}$. We define $S = \{S_P, S_U\}$ as a *sensitive feature*, where $|S_P^+|$ is the number of positively classified privileged values, and $|S_U^+|$ represents the number of positively classified unprivileged values. As an example, consider the feature S representing 'Race': the number of positively classified 'White' values is denoted by $|S_P^+|$; the number of positively unprivileged 'Non-White' values are denoted by $|S_U^+|$.

In our approach, we also handle missing values by fractionalising the missing values between both class values. Considering $|S_M|$ as the total number of missing values of the sensitive feature S and $|S_M^+|$ defines positively labeled missing values. Since the value for the feature is missing, we don't know whether to consider them as privileged or unprivileged. Therefore, we include a fraction of them into each group based on the proportion of known privileged or unprivileged values, given by:

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

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

where, F_P^+ is the fraction of the positively labeled missing values estimated to have privileged values; F_U^+ represents the fraction of the positively labeled missing values estimated to have unprivileged values. These definitions are used to define the proportion of instance on each group to determine whether a classification is fair or not, given by:

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

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

where $Pr(S_P^+)$ is the proportion of positively classified privileged instances and $Pr(S_U^+)$ is the proportion of positively classified unprivileged instances.

There are many fairness measures proposed in the literature for calculating a fairness (or discrimination) score. In this work, we adopt *statistical parity*—often called the CV (Calders–Verwer) score or simply the discrimination score [14, 23]. Statistical parity is one of the most widely used group-fairness metrics for quantifying discrimination [14, 27, 28, 30, 34, 45]. It measures the difference between the conditional probabilities of receiving the positive class for the unprivileged and privileged groups: the closer this difference is to zero, the lower the level of discrimination. As shown in Equation 5, for a sensitive attribute S (e.g., race), the discrimination score is obtained by subtracting the probability of a positive label for the "Not-White" group ($S \neq 1$) from that of the "White" group ($S = 1$).

$$CV = Pr(\hat{Y} = 1 \mid S = 1) - Pr(\hat{Y} = 1 \mid S \neq 1). \tag{5}$$

Given that the pruning procedure needs to determine whether to prune or not a node of the tree, a discrimination score is calculated using the set of instances at the current node. The discrimination score for a given set of instances S is given by:

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

where $Disc$ measures the discrimination level between the proportional number of positively classified privilege and unprivileged groups in relation to a sensitive feature. Depending on the value of $Disc$, we observe the following outcomes:

- $Disc = 0$: there is no discrimination between two groups;
- $Disc > 0$: there is a discrimination to *unprivileged* group;
- $Disc < 0$: there is a discrimination to *privileged* group.

3.2 Handling Group Fairness in Pruning

While C4.5 decision tree algorithm is able to create decision trees using an interpretable structure, there are many cases in which the trees are large and complex. Additionally, there is a risk that these trees overfit the training data and do not generalise well. Therefore, a pruning procedure is used to reduce its complexity and improve generalisation after constructing a decision tree on a training dataset. Quinlan [37] proposed an *error-based* pruning technique. This method calculates an estimated (pessimistic) error rate for each node and decides to prune the nodes that lead to an improvement of the estimated error, thereby creating a more comprehensible and accurate tree structure. It follows a bottom-up approach, calculating an error rate at each node. The error rate for a leaf, branch, and sub-tree are determined based on the confidence level for an adapted binomial distribution [37]. More precisely, the number of misclassified instances (E) and the total

number of instances (N) in a given leaf are used to calculate the binomial distribution and estimate an error rate. If the estimated error rate of a leaf is lower than that of the sub-tree and branch, a leaf pruning is applied—e.g., the sub-tree is replaced by a leaf. Conversely, if the estimated error rate of a branch is lower than the sub-tree's, the sub-tree is pruned in favor of the branch. The error-based pruning has demonstrated higher accuracy compared to other pruning techniques [7].

Despite improving accuracy of C4.5 tree by simplifying its complex structure, error-based pruning fails to generate fair classification results. Addressing this concern, we extended pruning of C4.5 trees that uses training data for pruning, either with an estimated error rate or without it, to include a group fairness measure to guide the decision to prune or not a sub-tree. The goal of this strategy is to enhance fairness, minimising a negative impact on the accuracy. While classical error-rate calculations are still used to estimate an error rate in some of our proposed strategies, discrimination levels of the leaf, sub-tree, and branch are assessed using Equation 6 before the pruning procedure. However, the decision to prune either heavily depends on the fairness constraint or equally looks at the fairness constraint and error rate. Thus, the strategy also examines the number of classified instances of privileged and unprivileged groups to establish group fairness. If there are insufficient instances to calculate the discrimination level, we apply a penalty to prevent pruning. Furthermore, our proposed fairness-guided post-pruning allows for a user-defined threshold, t, to reach the desired level of fairness while controlling the accuracy-fairness trade-off.

In essence, we proposed three different pruning strategies to process fairness during post-processing stage:

Fair-Only (FO) Pruning. The strategy prunes the subtree by replacing it by its most common branch or leaf based on the discrimination score *Disc* to enhance fairness without considering the error rate in each pruning iteration, after building a whole tree depending on the traditional gain ratio of C4.5. This is the way to enhance the fairness of an already given tree that aims to maximize accuracy. Additionally, we use thresholds of 0 (threshold-free), 0.5, and 0.8: pruning is permitted if the leaf provides less discrimination that is at least 50% or 80% of the discrimination level of the node that is a candidate for pruning (e.g., if the fairness level of the leaf is higher than at least 50% (for $t = 0.5$) or 80% (for $t = 0.8$) of the fairness level of the subtree); otherwise, a penalty is enforced to avoid pruning at the current iteration. The high-level pseudocode for fair-only (FO) pruning is presented in Algorithm 1.

FairXError (FxE) Pruning. We extended the traditional error-based pruning approach, originally introduced by Quinlan [37] to reduce overfitting and enhance generalization, by incorporating a fairness component into the pruning process. Drawing inspiration from GRXFR [5] and fair information gain [48], our FairXError method integrates fairness without sacrificing accuracy. Specifically, at each pruning iteration, we evaluate the product of a subtree's discrimination level and its error rate, and then compare this product with the corresponding product for each potential leaf or branch. Under the threshold-free setting ($t = 0$), a subtree is replaced by a leaf or branch if the product of that leaf's (or branch's) discrimination level and error rate is strictly lower than the subtree's. For a predefined threshold $t = 0.5$ (or 0.8), a subtree is pruned if the alternative leaf (or branch) yields at least a 50% (or 80%) reduction in the product of its discrimination level and error rate relative to the subtree. In this way, FairXError balances fairness considerations with error-based pruning to potentially reduce discrimination while preserving the advantages of conventional pruning for accuracy. The high-level pseudocode in Algorithm 2 outlines this strategy.

Algorithm 1 High-level pseudocode for the fair-only pruning procedure. The value with suffix D corresponds to discrimination score, measured by a group fairness criteria; t represents the pre-defined thresholds.

Require: training examples at the current node (sub-tree)
Ensure: pruned node
1: $\{subtree_D\} \leftarrow$ {discrimination score in subtree};
2: $\{branch_D\} \leftarrow$ {discrimination score in branch };
3: $\{leaf_D\} \leftarrow$ {discrimination score in leaf };
4: **if** $leaf_D \leq subtree_D$ - $t \times subtree_D$ **and** $leaf_D \leq branch_D$ **then**
5: **return** *leaf*;
6: **else if** $branch_D \leq subtree_D$ - $t \times subtree_D$ **then**
7: **return** *branch*;
8: **else**
9: **return** *subtree*;
10: **end if**

Algorithm 2 High-level pseudocode for the FairXError pruning procedure. The values with suffix E and D corresponds to estimated error rate and discrimination score, respectively. The discrimination score D is measured by a group fairness criteria; t represents the pre-defined thresholds.

Require: training examples at the current node (sub-tree)
Ensure: pruned node
1: $\{subtree_E, subtree_D\} \leftarrow$ {estimated error rate, discrimination score};
2: $\{branch_E, branch_D\} \leftarrow$ {estimated error rate, discrimination score};
3: $\{leaf_E, leaf_D\} \leftarrow$ {estimated error rate, discrimination score};
4: **if** $(leaf_E \times leaf_D) \leq (subtree_E \times subtree_D - t \times subtree_E \times subtree_D)$ **and**
$(leaf_E \times leaf_D) \leq (branch_E \times branch_D)$ **then**
5: **return** *leaf*;
6: **else if** $(branch_E \times branch_D) \leq (subtree_E \times subtree_D - t \times subtree_E \times subtree_D)$ **then**
7: **return** *branch*;
8: **else**
9: **return** *subtree*;
10: **end if**

Fair&Error (F&E) Pruning. This strategy also extends the traditional error-based pruning strategy by incorporating a fairness criterion. It retains the subtree's estimated error rate while adding a discrimination-level check at each pruning step. This strategy effectively balances accuracy and fairness by verifying both metrics iteratively, potentially improving final accuracy and fairness in a more streamlined tree structure. Although Fair&Error shares similarities with FairXError—in that both rely on discrimination and error rates for pruning—they differ in their approach. Whereas FairXError multiplies discrimination and error rates to guide pruning, Fair&Error requires both the error rate and the discrimination

level to meet specific conditions simultaneously. Fair&Error also allows users to specify thresholds values t: 0 (threshold-free), 0.5, or 0.8 to manage the trade-off between accuracy and fairness. For example, if $t = 0.8$, a subtree is pruned if (1) the leaf's error rate is lower than that of the subtree; and (2) the leaf's discrimination level is at most 80% of the subtree's discrimination level. In other words, the leaf must offer at least an 80% improvement in fairness relative to the subtree. If these conditions are not met, the pruning process retains the subtree and proceeds to the next iteration. The high-level pseudocode for this approach is provided in Algorithm 3.

Algorithm 3 High-level pseudocode for the Fair&Error pruning procedure
The values with suffix E and D corresponds to estimated error rate and discrimination score, respectively. The discrimination score D is measured by a group fairness criteria; t represents the pre-defined thresholds.

Require: training examples at the current node (sub-tree)
Ensure: pruned node
1: $\{subtree_E, subtree_D\} \leftarrow$ {estimated error rate, discrimination score};
2: $\{branch_E, branch_D\} \leftarrow$ {estimated error rate, discrimination score};
3: $\{leaf_E, leaf_D\} \leftarrow$ {estimated error rate, discrimination score};
4: **if** ($leaf_D \leq subtree_D$ - $t\times$ $subtree_D$) **and** ($leaf_E \leq subtree_E$) **and** ($leaf_D \leq branch_D$) **and** ($leaf_E \leq branch_E$) **then**
5: **return** *leaf*;
6: **else if** ($branch_D \leq subtree_D$ - $t\times$ $subtree_D$) **and** ($branch_E \leq subtree_E$) **then**
7: **return** *branch*;
8: **else**
9: **return** *subtree*;
10: **end if**

Table 1: Summary of the data sets used in the experiments.

Data set	#	Class		Size	Sensitive Attribute		
Variant		Attribute	Value (+)		Name	*P*	*U*
Adu	15	Income	$> 50k$	48842			
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
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
Dut	12	occupation	1	60420			
Dut(G)					Gender	Male	Female
Law	17	pass bar	1	22407			
Law(G)					Race	White	Non-White
UF	11	Mean GPA	≥ 3	43303			
UF(G)					Gender	Male	Female

4 COMPUTATIONAL EXPERIMENTS

In this section, we presented the experimental results of proposed fairness-guided pruning variants applied on eleven datasets that have been widely used in fairness literature. We first introduce the datasets, followed by the results of conducted experiments to evaluate the efficiency of proposed pruning strategies.

In the experiments, each dataset with its corresponding sensitive feature constitutes a variant, resulting in nineteen data-variants for the evaluation of the proposed work. The datasets used in the computational experiments are: Adult [24], German [21], Ricci [2], Student Mathematics and Student Portuguese Performances [11], Drug Consumption [12], Propublica Recidivism and Propublica Violent Recidivism [26], Dutch [22], Law School admission [43] and UFRGS—Federal University of Rio Grande do Sul extrance exam and GPA data in Brazil—[8]. The details of each nineteen dataset variants are presented in Table 1, where the number of instances (**Size**), number of features (**Features**), target class (**Class Attribute**), and the positive class label within the target class (**Value(+)**), the total number of instances on the dataset, sensitive attribute name (**Sensitive Attribute Name**), privileged and unprivileged values (**P** and **U**, respectively).

We implemented tenfold cross-validation for each dataset variant during both the building and pruning phases of the tree. In tenfold cross-validation, the data is divided into 10 equal or nearly equal parts, with each part containing a similar number of instances and class labels. For each of the ten iterations, one partition is used as unseen testing data while the remaining nine parts are utilized for training.

After constructing the original C4.5 decision tree, we apply our proposed fairness-guided pruning variants. Each variant defines a threshold t, where $t = 0$ corresponds to the threshold-free approach, and $t = 0.5$ or $t = 0.8$ imposes threshold-based fairness constraints. For example, setting $t = 0.5$ means that a leaf must provide at least a 50% improvement in fairness compared to the subtree before replacing it at each pruning step. By examining various thresholds in each fairness-guided variant, we aim to determine which threshold offers the most favourable balance between accuracy and fairness without excessively degrading predictive performance.

In addition to the conventional error-based prunning baseline (*Error*), we compare our strategies with *Relab*, the relabelling procedure introduced by Kamiran et al. [22]. *Relab* is a post-processing technique that revises the class labels at the leaves of a pre-trained binary decision tree to improve fairness. The method uses a parameter $\epsilon \in \{0.0, 0.01, 0.02, 0.03, 0.04\}$, which specifies the maximum discrimination to adjust the degree relabelling. To explore the trade-off between fairness and predictive performance, we set $\epsilon = 0.01$—the value that achieved the best overall rank across datasets when balancing accuracy and discrimination. We observed, however, that

for larger values—particularly at $\epsilon = 0.04$—*Relab* leaves the tree unchanged; because no relabelling occurs, accuracy reverts to that of the original model. Complete accuracy and discrimination results for $\epsilon \in \{0.0, 0.01, 0.02, 0.03, 0.04\}$ are provided in Appendix A.

4.1 Accuracy and Fairness Results

We evaluated three different variations of the fairness-guided pruning strategy, each incorporating a distinct threshold ($t = 0$, $t = 0.5$, and $t = 0.8$) for the discrimination level measure, comparing them to C4.5 traditional error-based pruning (error) and the relabelling (Relab) post-processing approach [22]. In our approach, the addition of these thresholds was intended to assess the strategy's effectiveness in reducing discrimination with minimal or no loss in accuracy during the pruning. Furthermore, we aim to observe the side-effect of enforcing fairness across different demographic groups by increasing pre-defined thresholds.

The accuracy results are presented in Table 2, while Table 3 compares AUROC results. In addition to accuracy and AUROC, which are classic performance metrics for evaluating classification models, we also evaluated our proposed approaches from a fairness perspective using statistical parity (*group* fairness) [14] and disparate treatment (*individual* fairness) [6] metrics; their results are provided in Table 4 and Table 5, respectively. Each value in these tables represents the average of tenfold cross-validation runs. For each pruning strategy, we have included its average rank, with the lowest average rank indicating the best performance regarding accuracy and fairness scores. When computing these ranks, we assign the worst possible rank to any variant whose mean tree size falls below 1.5, as shown in Table 7. This indicates that the approach mostly produced a single-node model, thereby failing to build a decision tree.

Considering the accuracy results in Table 2, the higher accuracy results are typically observed as the threshold t increases from 0.0 to 0.8 for each fairness-guided pruning variant. The Fair&Error (F&E) pruning variant provided the highest accuracy scores for the majority of datasets compared to error-based pruning and Relab. In particular, the threshold-free ($t = 0$) version of it was the best performer with the best average rank of 2.37. In several datasets, Relab does not achieve accuracy results comparable to those of our proposed pruning strategy. However, Relab is able to reach competitive AUROC results with an average rank of 2.16. Among our proposed approaches, only the Fair&Error variant with 0.5 did not lead to substantial degradation in AUROC across most dataset variants.

To assess the fairness impact of our pruning strategies, we firstly refer to Table 4, where lower discrimination scores indicate a reduction in bias. Fair-Only (F-O) showed a consistent decrease in discrimination scores across multiple datasets as the threshold was increased up to 0.8, ultimately achieving the best average rank of 3.84. In contrast, FairXError (FxE) and Fair&Error (F&E) did not follow a similar pattern: raising the threshold to 0.8 often adversely affected group fairness. Specifically, FxE achieved its best average rank of 3.9 at $t = 0.5$, but its performance deteriorated (average rank 4.32) when $t = 0.8$. Meanwhile, the threshold-free (i.e., $t = 0$) version of F&E led to better fairness outcomes than its threshold-based versions, with an average rank of 4.63. When compared with standard error-based pruning of C4.5, F-O with $t = 0.8$ and FxE with $t = 0.5$ offered a more substantial reduction in discrimination. In contrast, F&E did not yield a comparable improvement, as its pruning process requires both error rate and discrimination constraints to be satisfied simultaneously; if these are not met, further pruning and thus further fairness improvements are restricted. Beyond simple error-based pruning (*error*), F-O ($t = 0.8$) and FxE ($t = 0.5$) also reduced discrimination relative to Relab, although Relab provided stronger fairness outcomes at the expense of diminished predictive performance. Comparing disparate treatment results as an individual fairness metric in Table 5, FxE ($t = 0.8$) outperformed all other post-processing approaches, including standard error-based pruning of C4.5 and Relab, with an average rank of 3.21. Additionally, as the threshold t increased through 0.8, a reduction in discrimination was notably observed, particularly for Fair-Only (F-O) and FairXError (FxE), which performed better compared to Relab and error-based pruning.

To evaluate the interpretability of our proposed post-processing strategies, we report the size of the pruned trees in Table 7 along with the unpruned C4.5. We exclude Relab from this comparison because, by design, its relabelling method does not modify tree size but rather changes the labels at the leaves. Our findings show that the threshold-free versions of the fairness-guided pruning variants (except F&E) often produce a single-node tree that classifies all samples into the majority class. In contrast, as the threshold is raised to 0.8, the pruning strategies tend to simplify the trees rather than collapsing them into a single node. Particularly, F-O and FxE at a threshold of 0.8 simultaneously achieve promising accuracy, fairness, and interpretability by substantially reducing tree depth [13]. Moreover, the number of nodes is strongly influenced by the threshold level, which governs the strictness of the pruning constraints. As an extension of error-based pruning in C4.5, F&E yields trees whose sizes are similar to or slightly larger than those generated by traditional error-based pruning. However, F&E never collapses to a single node at $t = 0.8$, and in most cases involving other thresholds, it produces a non-trivial tree structure for the majority of the dataset variants.

Based on the overall ranks presented in Table 6, our novel pruning variants generally offer a favourable balance between accuracy and fairness as the threshold is increased. While threshold-free versions of the fairness-guided pruning strategies typically produce single-node trees, resulting in lower accuracy and fairness compared to their threshold-based counterparts—the trends in Table 6 indicate a simultaneous improvement in accuracy and fairness measures at higher thresholds. A notable exception arises with F&E, where more stringent thresholds restrict pruning and thus limit further fairness gains.

Among the fairness-guided pruning variants, F–O with threshold $t = 0.8$ achieved higher accuracy and superior group- and individual-fairness scores than Relab, although Relab maintained a better AUROC. The threshold-free F&E and the FxE variant at $t = 0.8$ also performed well, either reducing discrimination or preserving accuracy, as reflected by their respective ranks. Overall, F–O and FxE at $t = 0.8$ provide the best balance between accuracy and fairness when compared with error-based pruning and Relab. They also reduce model complexity, as evidenced by the smaller tree sizes relative to the unpruned C4.5 baseline in Table 7. While

Table 2: Accuracy results of fairness-guided pruning variants. For each variant, the numbers in parentheses, 0, 0.5 and 0.8, show the user-defined threshold, t. The best results (highest values) are shown in bold. The average ranks are displayed at the bottom of the table.

ACCURACY RESULTS											
	F-O			FxE			F&E			C4.5	Relab
Dataset	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(ϵ=0.01)
Adu(G)	0.7638	0.8031	0.8031	0.8031	0.8028	0.803	0.8566	0.857	**0.8573**	0.8565	0.7296
Adu(R)	0.8031	0.8031	0.8031	0.8031	0.803	0.803	0.8565	0.8565	**0.8574**	0.8565	0.7927
Adu(A)	0.7638	0.8031	0.8031	0.8031	0.8031	0.8031	0.8569	0.8569	**0.8573**	0.8565	0.3981
Ger(G)	0.7	0.7	0.701	0.7	0.7	0.7	**0.708**	0.705	0.707	**0.708**	0.6790
Ger(A)	0.7	0.7	0.7	0.7	0.7	0.7	0.707	0.705	**0.715**	0.708	0.6680
Ric(R)	0.7288	0.7955	0.8061	0.8068	0.8318	0.8311	**0.872**	**0.872**	**0.872**	**0.872**	0.4909
StuM(G)	0.8808	0.8907	0.9039	**0.9189**	**0.9189**	0.9164	0.9062	0.9037	0.8987	0.9012	0.8565
StuM(A)	0.7921	0.8503	**0.919**	0.9164	0.9164	0.9064	0.8987	0.8987	0.9012	0.9012	0.8760
StuP(G)	0.8628	0.8967	0.8967	0.889	0.9074	0.9105	0.926	0.926	0.9245	**0.9291**	0.7450
StuP(A)	0.849	0.869	0.8767	0.8705	0.872	0.8999	0.926	0.926	0.9199	**0.9291**	0.9198
Dru(G)	0.7581	0.7581	0.7719	0.7581	0.7618	0.7926	**0.8302**	0.8297	**0.8302**	**0.8302**	0.7947
Dru(R)	0.7581	0.7581	0.7581	0.7581	0.7581	0.7581	0.8297	0.8291	**0.8302**	**0.8302**	0.7915
Pro(G)	0.5493	0.6224	0.6435	0.5931	0.6224	0.6564	**0.6755**	0.6744	0.6719	**0.6755**	0.6665
Pro(R)	0.601	0.6422	0.6568	0.6241	0.6546	0.6573	**0.6756**	0.6751	0.6742	0.6755	0.6711
ProV(G)	0.8366	0.8366	0.8351	0.8366	0.8366	0.836	0.8366	0.8366	0.836	0.8366	**0.8377**
ProV(R)	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366	0.8366	**0.8377**
Dut(G)	0.5239	0.5774	0.6902	0.5239	0.7079	0.7085	0.7269	0.7276	0.7269	0.7269	**0.7947**
Law(R)	0.9478	0.9478	0.9436	0.9478	0.9478	**0.9482**	0.9478	0.9478	0.9478	0.9478	0.9478
UF(G)	0.5321	0.5772	0.6094	0.5321	0.6139	0.6087	0.6464	**0.6467**	0.6459	0.6464	0.6430
AvgRank	6.9	5.95	4.58	5.47	4.74	4.11	**2.37**	2.74	2.42	2.42	4.79

Table 3: AUROC results of fairness-guided pruning variants. For each variant, the numbers in parentheses, 0, 0.5 and 0.8, show the user-defined threshold, t. The best results (highest values) are shown in bold. The average ranks are displayed at the bottom of the table.

AUROC RESULTS											
	F-O			FxE			F&E			C4.5	Relab
Dataset	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(ϵ=0.01)
Adu(G)	0.5	0.5839	0.5839	0.5839	0.5833	0.5836	0.7584	0.76	0.7591	0.7583	**0.7978**
Adu(R)	0.5839	0.5839	0.5839	0.5839	0.5836	0.5836	0.7583	0.7583	0.7593	0.7583	**0.8021**
Adu(A)	0.5	0.5839	0.5839	0.5839	0.5839	0.5839	0.7576	0.7576	**0.7591**	0.7583	0.7199
Ger(G)	0.5	0.5	0.5017	0.5	0.5	0.5	0.5448	0.5521	0.5364	0.5457	**0.6886**
Ger(A)	0.5	0.5	0.5	0.5	0.5	0.5	0.5564	0.5483	0.584	0.5457	**0.6776**
Ric(R)	0.7281	0.7948	0.8081	0.8098	0.8348	0.8331	**0.8731**	**0.8731**	**0.8731**	**0.8731**	0.4844
StuM(G)	0.8644	0.8777	0.9108	**0.9239**	**0.9239**	0.92	0.8987	0.8949	0.8872	0.891	0.9011
StuM(R)	0.7043	0.7967	0.9219	0.92	0.92	0.9046	0.8872	0.8872	0.891	0.891	**0.9408**
StuP(G)	0.6285	0.7876	0.7794	0.7217	0.8226	0.8162	0.8213	**0.8254**	0.8204	0.8231	0.6976
StuP(R)	0.5918	0.6976	0.6532	0.6659	0.7403	0.765	0.8213	0.8213	0.8095	0.8231	**0.9196**
Dru(G)	0.5	0.5	0.5464	0.5	0.5162	0.6066	0.6966	0.6955	0.6951	0.6966	**0.7712**
Dru(R)	0.5	0.5	0.5	0.5	0.5	0.5	0.6955	0.6944	0.6951	0.6966	**0.7718**
Pro(G)	0.5	0.6051	0.635	0.5634	0.6051	0.649	0.6632	0.6622	0.6608	0.6632	**0.7087**
Pro(R)	0.5764	0.6339	0.6537	0.6072	0.6514	0.6563	0.6633	0.6638	0.664	0.6632	**0.7126**
ProV(G)	0.5	0.5	0.5074	0.5	0.5	0.5068	0.5	0.5	0.5089	0.5	**0.7069**
ProV(R)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.7069**
Dut(G)	0.5	0.5628	0.6943	0.5	0.7151	0.7143	0.7271	0.7276	0.7267	0.7271	**0.7713**
Law(R)	0.5	0.5	0.5111	0.5	0.5	0.5079	0.5	0.5	0.5	0.5	**0.8257**
UF(G)	0.5	0.557	0.5936	0.5	0.6035	0.5973	0.6406	0.641	0.64	0.6406	**0.6848**
AvgRank	7.47	6.47	5.26	6.11	5.11	4.90	3.47	3.26	3.53	3.42	**2.16**

Table 4: Discrimination scores of fairness-guided pruning variants. For each variant, the numbers in parentheses, 0, 0.5 and 0.8, show the user-defined threshold, t. The best results (values close to 0) are shown in bold. The average ranks are displayed at the bottom of the table.

DISCRIMINATION SCORE											
	F-O			FxE			F&E			C4.5	Relab
Dataset	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(ϵ=0.01)
Adu(G)	0.0	**0.0289**	**0.0289**	**0.0289**	0.0294	0.0296	0.1774	0.1784	0.1794	0.1773	-0.0682
Adu(R)	-0.0212	-0.0212	-0.0212	-0.0212	-0.0209	-0.0209	-0.1049	-0.1049	-0.1054	-0.1049	**0.008**
Adu(A)	0.0	**0.0446**	**0.0446**	**0.0446**	**0.0446**	**0.0446**	0.2079	0.2079	0.2093	0.4347	-0.4347
Ger(G)	0.0	0.0	**0.0016**	0.0	0.0	0.0	-0.006	-0.0084	-0.0092	-0.0114	-0.0167
Ger(A)	0.0	0.0	0.0	0.0	0.0	0.0	-0.0615	-0.0547	-0.1267	**-0.0408**	-0.0755
Ric(R)	0.2357	0.2729	0.2831	0.2888	0.3145	0.3356	0.4046	0.4046	0.4046	0.4046	**-0.0233**
StuM(G)	**0.0511**	0.0554	0.0902	0.0529	0.0529	0.0569	0.0711	0.0751	0.0754	0.0696	0.0428
StuM(A)	0.2133	0.142	**0.0236**	0.0307	0.0307	0.0428	0.0386	0.0386	0.0301	-0.0244	0.1205
StuP(G)	-0.0191	-0.0562	-0.0513	-0.0388	-0.0733	-0.0609	-0.0274	-0.0228	-0.019	-0.0239	**-0.0464**
StuP(A)	-0.0438	0.0461	0.052	0.0477	**0.0026**	-0.0569	-0.0436	-0.0436	-0.0423	0.0449	-0.0625
Dru(G)	0.0	0.0	0.0065	0.0	**-0.002**	0.0326	0.0867	0.0878	0.0867	0.0867	-0.0078
Dru(R)	0.0	0.0	0.0	0.0	0.0	0.0	0.003	0.0036	0.0048	**0.0025**	0.0422
Pro(G)	0.0	-0.1158	-0.1117	0.1157	0.167	0.1821	-0.1747	-0.1658	-0.1307	-0.1747	**-0.1071**
Pro(R)	0.0856	0.1424	0.1737	**0.0**	**0.0**	-0.0067	0.1357	0.137	0.1417	0.1355	**0.1793**
ProV(G)	0.0	0.0	-0.0082	0.0	0.0	**-0.0067**	0.0	0.0	-0.0092	0.0	-0.0357
ProV(R)	0.0	0.0	**0.0**	0.0	0.0	**0.0**	0.0	0.0	**0.0**	0.0	0.0622
Dut(G)	0.0	0.0357	0.116	0.0	0.1271	0.1855	0.4272	0.4381	0.4358	0.4272	**-0.0055**
Law(R)	0.0037	0.0037	0.0163	0.0037	0.0037	0.0105	0.0037	0.0037	0.0037	0.0037	**0.001**
UF(G)	0.0	**0.0533**	0.0585	0.0	0.081	0.1526	0.3082	0.3094	0.3168	0.3084	-0.5853
AvgRank	5.9	4.9	**3.84**	5.11	3.9	4.32	4.63	4.9	4.79	4.47	4.21

Table 5: Disparate Treatment results of fairness-guided pruning variants. For each variant, the numbers in parentheses, 0, 0.5 and 0.8, show the user-defined threshold, t. The best results (values close to 0) are shown in bold. The average ranks are displayed at the bottom of the table.

DISPARATE TREATMENT											
	F-O			FxE			F&E			C4.5	Relab
Dataset	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(ϵ=0.01)
Adu(G)	0.0	0.2201	0.2201	0.2201	0.218	0.2184	0.6104	0.6154	0.5828	0.6003	0.8378
Adu(R)	0.2137	0.2137	0.2137	0.2137	0.2130	0.2130	0.5844	0.5844	0.5446	0.5775	0.0854
Adu(A)	0.0	0.122	0.122	0.122	0.122	0.122	0.2241	0.2241	0.2253	0.2256	5.371
Ger(G)	0.0	0.0	0.0016	0.0	0.0	0.0	0.1856	0.2515	0.1643	0.1936	0.473
Ger(A)	0.0	0.0	0.0	0.0	0.0	0.0	0.3772	0.3378	0.5945	0.2486	0.6797
Ric(R)	0.5653	0.5024	0.684	0.6343	0.58	0.6275	0.6225	0.6225	0.6225	0.6225	0.9297
StuM(G)	1.356	1.377	1.257	1.2965	1.2965	1.2988	1.3511	1.3534	1.3609	1.3623	0.9693
StuM(R)	0.9706	0.8861	0.6408	0.6307	0.6307	0.6603	0.6967	0.6967	0.6796	0.6777	1.051
StuP(G)	1.9894	1.8073	1.82	1.8865	1.7784	1.8017	1.8555	1.8492	1.8572	1.864	1.2007
StuP(R)	1.8456	1.5598	1.6448	1.6042	1.4696	1.4589	1.423	1.423	1.4272	1.4321	1.41045
Dru(G)	4.3931	4.3931	4.0982	4.3931	4.2786	3.811	3.4432	3.4492	3.4602	3.4432	2.8917
Dru(R)	3.331	3.331	3.331	3.331	3.331	3.331	2.6147	2.6155	2.6174	2.6141	3.993
Pro(G)	2.52	1.7979	1.5006	2.047	1.7979	1.4829	1.6981	1.6771	1.568	1.6981	1.5736
Pro(R)	1.6755	1.3598	1.2687	1.5552	1.2581	1.237	1.3874	1.3696	1.3585	1.3874	1.3535
ProV(G)	4.8875	4.8875	4.8506	4.8875	4.8875	4.8687	4.8875	4.8875	4.8655	4.8875	3.8485
ProV(R)	3.0821	3.0821	3.0821	3.0821	3.0821	3.0821	3.0821	3.0821	3.0821	3.0821	4.6225
Dut(G)	0.0	0.653	1.9691	0.0	2.1864	2.084	1.6615	1.6392	1.6333	1.6615	2.8896
Law(R)	2.8376	2.8376	2.7964	2.8376	2.8376	2.8161	2.8376	2.8376	2.8376	2.8376	2.535
UF(G)	0.0	0.4753	0.704	0.0	0.8855	0.8248	1.0463	1.0508	1.0414	1.0465	0.9387
AvgRank	7.0	5.21	3.58	5.84	4.05	**3.21**	4.53	4.68	4.16	4.79	4.0

Table 6: General ranks of post-processing strategies according to their average ranks.

	F-O			FxE			F&E			C4.5	Relab
Metrics	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(ϵ=0.01)
Accuracy	10	9	5	8	6	4	**1**	3	2	2	7
AUROC	11	10	8	9	7	6	4	2	5	3	**1**
Discrimination Score	10	8	**1**	9	2	4	6	8	7	5	3
Disparate Treatment	11	9	2	10	4	**1**	6	7	5	8	3

Table 7: Tree sizes of the fairness-guided pruning variants measured as the number of nodes in each tree.

TREE SIZES											
	F-O			FxE			F&E			C4.5	C4.5
Dataset	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(t=0)	(t=0.5)	(t=0.8)	(error)	(unpruned)
Adu(G)	1.0	2.0	5.0	2.0	86.0	31.0	59.0	66.0	71.0	58	2138
Adu(R)	2.0	2.0	2.0	2.0	32.0	23.0	58.0	63.0	86.0	58	2138
Adu(A)	1.0	2.0	2.0	2.0	2.0	2.0	51.0	55.0	78.0	58	2138
Ger(G)	1.0	1.0	2.2	1.0	1.0	1.8	14.3	22.6	12.5	12.3	352.3
Ger(A)	1.0	1.0	1.0	1.0	1.0	1.0	14.2	13.1	37.0	12.3	352.3
Ric(R)	2.8	5.3	6.4	3.3	4.4	7.5	5.6	7.8	9.1	5.6	10.3
StuM(G)	3.5	4.2	5.4	3.1	4.0	5.1	12.0	12.7	17.4	11.3	31.2
StuM(A)	2.0	2.9	4.3	2.9	3.6	4.0	12.1	12.1	15.3	11.3	31.2
StuP(G)	1.6	2.9	4.0	1.9	3.1	4.6	7.2	8.2	8.8	6.4	41
StuP(A)	1.3	1.7	1.9	1.6	1.8	2.4	6.7	6.7	10.3	6.4	41
Dru(G)	1.0	1.0	3.8	1.0	1.8	6.9	20.4	23.1	29.8	19.1	626.5
Dru(R)	1.0	1.0	1.0	1.0	1.0	1.0	20.5	23.7	28.6	19.1	626.5
Pro(G)	1.0	2.4	5.6	1.5	2.1	7.4	101.5	9.5	11.1	14.1	554.9
Pro(R)	1.8	2.9	5.4	2.0	3.3	5.3	77.7	9.6	11.7	17.1	554.9
ProV(G)	1.0	1.1	2.3	1.0	1.0	2.2	1.0	1.1	2.7	1.0	879.9
ProV(R)	1.0	1.3	1.7	1.0	1.2	1.7	1.0	1.2	1.5	1.0	879.9
Dut(G)	1.0	1.8	6.5	1.0	4.9	8.3	13.6	18.4	33.2	13.6	383.7
Law(R)	1.0	1.9	5.0	1.0	1.7	4.3	1.0	1.3	3.3	1.0	331.6
UF(G)	1.0	1.5	4.6	1.0	2.5	3.5	97.2	117.3	142.2	94.8	296.9

the deterministic nature of our pruning strategies avoids the high computational overhead characteristic of search-based methods [18], implementing fairness in FxE and F&E does increase computational time compared to Relab and conventional error-based pruning, resulting in a potential limitation of our approach for applications sensitive to computational time.

5 CONCLUSION and FUTURE WORK

We proposed new fairness-guided pruning strategies for decision trees, focusing on incorporating a group fairness measure into the post-processing stage. Traditional deterministic pruning procedures have not been utilized to enhance fairness during post-processing, particularly for non-binary trees. Our experimental evaluation, conducted on eleven widely used fairness datasets in a total of nineteen dataset variations, demonstrates that our approach is able to improve group and individual fairness, measured by discrimination scores and disparate treatment, receptively. The improvement in fairness is achieved without significant impact on the accuracy. In comparison with another post-processing fairness approach in binary trees, our proposed method demonstrates improved accuracy and fairness for the majority of the dataset variants. Consequently, our fairness-guided pruning strategies strike a balance between maintaining predictive accuracy and improving fairness while also simplifying the tree structures, which might result in improving interpretability.

For future work, we plan to expand our methodology by incorporating different individual fairness metrics into the initial construction of decision trees. This will be the first step to extend our work and create a fairness-guided Random forest ensemble. The rationale is that by combining decision trees created from different fairness metrics, the ensemble can implicitly take into consideration different fairness metrics while maintaining a good predictive accuracy.

Acknowledgments

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

References

[1] Seyum Assefa Abebe, Claudio Lucchese, and Salvatore Orlando. 2022. EiFFFeL: Enforcing Fairness in Forests by Flipping Leaves. (2022). doi:10.1145/3477314.3507319

[2] Bryan L Adamson. 2011. Ricci v. DeStefano: Procedural Activism. *National Black Law Journal (University of California, Los Angeles)* 24 (2011), 11–01.

[3] Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. 2019. 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, 1418–1426. doi:10.1609/AAAI.V33I01.33011418

[4] Jose M Alvarez, Kristen M Scott, Ku Leuven, LeuvenAI Leuven, Belgium Bettina Berendt, Salvatore Ruggieri, and Bettina Berendt. 2023. Domain adaptive decision trees: Implications for accuracy and fairness. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. ACM, 11. doi:10.1145/3593013.3594008

[5] Meryem Bagriacik and Fernando E.B. Otero. 2024. Multiple fairness criteria in decision tree learning. *Applied Soft Computing* 167 (12 2024), 112313. doi:10.1016/j.asoc.2024.112313

[6] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *Proceedings of the 26th International Conference on World Wide Web* (2017). doi:10.1145/3038912.3052660

[7] Leonard A. Breslow and David W. Aha. 1997. Simplifying decision trees: A survey. *Knowledge Engineering Review* 12, 1 (1997), 1–40. doi:10.1017/S0269888997000015

[8] Bruno Castro da Silva. 2019. {UFRGS Entrance Exam and GPA Data}.

[9] Stefano Calzavara, Lorenzo Cazzaro, Claudio Lucchese, and Federico Marcuzzi. 2023. Explainable Global Fairness Verification of Tree-Based Classifiers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE Computer Society, 1–17. doi:10.1109/SATML54575.2023.00011

[10] Alessandro Castelnovo, Andrea Cosentini, Lorenzo Malandri, Fabio Mercorio, and Mario Mezzanzanica. 2022. FFTree: A flexible tree to handle multiple fairness criteria. *Information Processing and Management* 59, 6 (11 2022). doi:10.1016/j.ipm.2022.103099

[11] Paulo Cortez and Alice Silva. 2008. USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE. (2008).

[12] Elaine Fehrman, Vincent Egan, Alexander N. Gorban, Jeremy Levesley, Evgeny M. Mirkes, and Awaz K. Muhammad. 2019. *Personality Traits and Drug Consumption*. Springer International Publishing. doi:10.1007/978-3-030-10442-9

[13] Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* 15, 1 (2014).

[14] Sorelle A. Friedler, Sonam Choudhary, Carlos Scheidegger, Evan P. Hamilton, Suresh Venkatasubramanian, and Derek Roth. 2018. A comparative study of fairness-enhancing interventions in machine learning. In *FAT* 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, Inc, 329–338. doi:10.1145/3287560.3287589

[15] 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. doi:10.1007/s41019-020-00124-2

[16] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys (CSUR)* 51, 5 (8 2018). doi:10.1145/3236009

[17] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*. Neural information processing systems foundation, 3323–3331. https://arxiv.org/abs/1610.02413v1

[18] Max Hort, Jie M. Zhang, Federica Sarro, and Mark Harman. 2024. Search-based Automatic Repair for Fairness and Accuracy in Decision-making Software. *Empirical Software Engineering* 29, 1 (2 2024), 1–33. doi:10.1007/S10664-023-10419-3/TABLES/10

[19] Haewon Jeong, Hao Wang, and Flavio P Calmon. 2022. Fairness without imputation: A decision tree approach for fair prediction with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 9558–9566. https://ojs.aaai.org/index.php/AAAI/article/view/21189

[20] Nathanael Jo, Sina Aghaei, Jack Benson, Andres Gomez, and Phebe Vayanos. 2023. Learning Optimal Fair Decision Trees: Trade-offs Between Interpretability, Fairness, and Accuracy. In *AIES 2023 - Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, Inc, 181–192. doi:10.1145/3600211.3604664

[21] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication, IC4 2009*. doi:10.1109/IC4.2009.4909197

[22] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *Proceedings - IEEE International Conference on Data Mining, ICDM*. 869–874. doi:10.1109/ICDM.2010.50

[23] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Considerations on fairness-aware data mining. In *Proceedings - 12th IEEE International Conference on Data Mining Workshops, ICDMW 2012*. 378–385. doi:10.1109/ICDMW.2012.101

[24] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 7524 LNAI. doi:10.1007/978-3-642-33486-3{_}3

[25] Kentaro Kanamori and Hiroki Arimura. 2021. Fairness-Aware Decision Tree Editing Based on Mixed- Integer Linear Optimization. *Transactions of the Japanese Society for Artificial Intelligence* 36, 4 (2021), B–L13_1.

[26] Lauren Kirchner, Surya Mattu, Jeff Larson, and Julia Angwin. 2016. Machine Bias — ProPublica. *Propublica* (2016).

[27] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. 2022. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12, 3 (5 2022), e1452. doi:10.1002/WIDM.1452

[28] Yanhui Li, Linghan Meng, Lin Chen, Li Yu, Di Wu, Yuming Zhou, and Baowen Xu. 2022. Training Data Debugging for the Fairness of Machine Learning Software. In *Proceedings - International Conference on Software Engineering*, Vol. 2022-May. IEEE Computer Society, 2215–2227. doi:10.1145/3510003.3510091

[29] Camille Olivia Little, Debolina Halder Lina, and Genevera I. Allen. 2024. Fair Feature Importance Scores for Interpreting Decision Trees. *Transactions on Machine Learning Research* (2024).

[30] Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. 2021. *Exacerbating Algorithmic Bias through Fairness Attacks*. Technical Report. https://github.com/Ninarehm/attack

[31] Geraldin Nanfack, Valentin Delchevalerie, and Benoît Frénay. 2021. Boundary-based fairness constraints in decision trees and random forests. In *ESANN 2021 Proceedings-29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. i6doc.com publication, 375–380. doi:10.14428/esann/2021.ES2021-69

[32] Géraldin Nanfack, Paul Temple, and Benoît Frénay. 2022. 201 Constraint Enforcement on Decision Trees: A Survey. *ACM Comput. Surv* 54 (2022). doi:10.1145/3506734

[33] António Pereira Barata, Frank W. Takes, H. Jaap van den Herik, and Cor J. Veenman. 2023. Fair tree classifier using strong demographic parity. *Machine Learning* (8 2023), 1–20. doi:10.1007/S10994-023-06376-Z/FIGURES/7

[34] Dana Pessach and Erez Shmueli. 2023. A Review on Fairness in Machine Learning. *Comput. Surveys* 55, 3 (4 2023), 1–44. doi:10.1145/3494672

[35] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. 2016. What makes classification trees comprehensible? *Expert Systems with Applications* 62 (11 2016), 333–346. doi:10.1016/J.ESWA.2016.06.009

[36] J. R. Quinlan. 1987. Simplifying decision trees. *Int. J. Man Mach. Stud.* 51, 2 (1987), 497–510. doi:10.1006/IJHC.1987.0321

[37] J R Quinlan. 1993. *C4.5 Programs for Machine Learning*. Vol. 5. 302 pages. https://books.google.com/books/about/C4_5.html?hl=tr&id=HExncpjbYroC

[38] Edward Raff, Jared Sylvester, and Steven Mills. 2017. Fair Forests: Regularized Tree Induction to Minimize Model Bias. In *AIES 2018 - Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, Inc, 243–250. doi:10.1145/3278721.3278742

[39] Francesco Ranzato and Marco Zanella. 2020. Genetic Adversarial Training of Decision Trees. *GECCO 2021 - Proceedings of the 2021 Genetic and Evolutionary Computation Conference* (12 2020), 358–367. doi:10.48550/arxiv.2012.11352

[40] Jacobus G M Van Der Linden, Mathijs M De Weerdt, and Demirovi´c Demirovi´c. 2022. Fair and Optimal Decision Trees: A Dynamic Programming Approach. In *Advances in Neural Information Processing Systems*, Vol. 35. 38899–38911.

[41] Mingyang Wan. 2022. In-Processing Modeling Techniques for Machine Learning Fairness: A Survey. (2022). doi:10.1145/3551390

[42] Jingbo Wang, Yannan Li, and Chao Wang. 2022. Synthesizing Fair Decision Trees via Iterative Constraint Solving. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13372 LNCS (2022), 364–385. doi:10.1007/978-3-031-13188-2{_}18/FIGURES/4

[43] Linda F Wightman. 1998. LSAC National Longitudinal Bar Passage Study. LSAC Research Report Series. *ERIC* (1998).

[44] Zhiyu Xue. 2023. Group AdaBoost with Fairness Constraint. In *Proceedings*. Society for Industrial and Applied Mathematics, 865–873. doi:10.1137/1.9781611977653.CH97

[45] Qingquan Zhang, Jialin Liu, Zeqi Zhang, Junyi Wen, Bifei Mao, and Xin Yao. 2022. Mitigating Unfairness via Evolutionary Multi-objective Ensemble Learning. *IEEE Transactions on Evolutionary Computation* (9 2022), 1–1. doi:10.1109/tevc.2022.3209544

[46] Wenbin Zhang and Albert Bifet. 2020. FEAT: A Fairness-Enhancing and Concept-Adapting Decision Tree Classifier. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 12323 LNAI. doi:10.1007/978-3-030-61527-7{_}12

[47] Wenbin Zhang, Albert Bifet, Xiangliang Zhang, Jeremy C. Weiss, and Wolfgang Nejdl. 2021. FARF: A Fair and Adaptive Random Forests Classifier. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12713 LNAI (2021), 245–256. doi:10.1007/978-3-030-75765-6{_}20/TABLES/2

[48] Wenbin Zhang and Eirini Ntoutsi. 2019. FaHT: An adaptive fairness-aware decision tree classifier. In *IJCAI International Joint Conference on Artificial Intelligence*, Vol. 2019-August. doi:10.24963/ijcai.2019/205

[49] Wenbin Zhang and Liang Zhao. 2020. Online Decision Trees with Fairness. (10 2020). http://arxiv.org/abs/2010.08146

A Results of Relabelling (Relab)

This section presents results for the existing post-processing approach Relab [22] to identify the variant that most effectively improves group fairness. The best variant was used in Section 4.1 to observe how post-processing approaches, including the proposed ones, simultaneously affected fairness and accuracy. In Relab, class labels at the tree leaves are modified to optimise statistical parity (also known as the discrimination or CV score); hence we rank the Relab variants by both accuracy and discrimination.

As shown in the tables below, when the relabelling approaches $\epsilon = 0.04$, Relab typically reproduces the unmodified decision tree. Therefore, its results are identical to those of the original model for most dataset variants. More specifically, in 7 of the 11 datasets the run with $\epsilon = 0.04$ returns the original tree, indicating that this setting fails to mitigate discrimination. To make this behaviour explicit, the *OrgTree* column in Tables 8 and 9 reports the accuracy and fairness metrics of the unpruned tree; we excluded these values when computing the average ranks. According to Table 10, the configuration $\epsilon = 0.01$ achieves the lowest average rank (2) and the best discrimination score without reverting to the original tree, thereby producing the greatest reduction in discrimination.

Table 8: Accuracy comparisons of Relab variants. The best accuracy results, the highest values, are shown in bold. The average ranks are displayed at the bottom of the table.

ACCURACY RESULTS						
	Relabelling (Relab)					
Dataset	(ϵ=0)	(ϵ=0.01)	(ϵ=0.02)	(ϵ=0.03)	(ϵ=0.04)	*OrgTree*
Adu(G)	0.7296	0.7296	0.7296	0.7296	0.8068	0.8448
Adu(R)	0.3638	0.7927	0.8068	0.8068	0.8068	0.8448
Adu(A)	0.3981	0.3981	0.3981	0.3981	0.3981	0.8448
Ger(G)	0.6810	0.6790	**0.6840**	0.6820	0.6820	0.6820
Ger(A)	0.6580	0.6680	**0.6890**	0.6820	0.6820	0.6820
Ric(R)	0.4909	0.4909	0.5182	0.5182	0.6015	0.8826
StuM(G)	0.8538	0.8565	**0.9064**	**0.9064**	0.9038	0.9064
StuM(A)	0.8760	0.8760	0.8913	**0.9088**	**0.9088**	0.9064
StuP(G)	0.6726	0.7450	0.8705	0.8968	0.9029	0.9168
StuP(A)	0.9168	**0.9198**	**0.9198**	**0.9198**	0.9168	0.9168
Dru(G)	0.7878	0.7947	**0.7952**	0.7947	0.7947	0.7947
Dru(R)	0.7306	0.7915	**0.7947**	**0.7947**	**0.7947**	0.7947
Pro(G)	0.6589	0.6665	0.6704	**0.673**	**0.673**	0.673
Pro(R)	0.6626	0.6711	0.6730	0.6730	0.6730	0.673
ProV(G)	**0.8379**	0.8377	0.8377	0.8377	0.8377	0.8377
ProV(R)	**0.8396**	0.8377	0.8377	0.8377	0.8377	0.8377
Dut(G)	0.7923	0.7947	0.7947	0.7947	0.7947	0.8063
Law(R)	0.9465	**0.9478**	**0.9478**	**0.9478**	**0.9478**	0.9478
UF(G)	0.6247	**0.6430**	**0.6430**	**0.6430**	**0.6430**	0.6430
AvgRank	2.68	2.21	1.42	1.42	**1.37**	-

Table 9: Discrimination score comparisons of Relab variants. The best results, which are close to 0, are shown in bold. The average ranks are displayed at the bottom of the table.

DISCRIMINATION SCORES						
	Relabelling (Relab)					
Dataset	(ϵ=0)	(ϵ=0.01)	(ϵ=0.02)	(ϵ=0.03)	(ϵ=0.04)	*OrgTree*
Adu(G)	-0.0682	-0.0682	-0.0682	-0.0682	**0.0328**	0.1445
Adu(R)	-0.1463	**0.008**	0.0169	0.0169	0.0169	0.0586
Adu(A)	-0.4347	-0.4347	-0.4347	-0.4347	-0.4347	0.1792
Ger(G)	-0.0152	-0.0167	**-0.0136**	-0.0107	-0.0107	-0.0107
Ger(A)	-0.0868	**-0.0755**	-0.0845	-0.0942	-0.1035	-0.1035
Ric(R)	-0.0233	-0.0233	-0.0317	-0.0317	**-0.0067**	0.2894
StuM(G)	0.0559	**0.0428**	0.0522	0.0653	0.072	0.0694
StuM(A)	**0.1205**	**0.1205**	0.1256	0.1453	0.1453	0.1478
StuP(G)	-0.0388	-0.0464	-0.0393	-0.0134	**-0.0053**	-0.0190
StuP(A)	**-0.0523**	-0.0625	-0.0625	-0.0625	-0.0681	-0.0681
Dru(G)	-0.0237	-0.0078	-0.0072	**-0.0055**	**-0.0055**	**-0.0055**
Dru(R)	**0.0183**	0.0422	0.0491	0.0491	0.0477	0.0477
Pro(G)	**-0.093**	-0.1071	-0.1075	-0.1082	-0.1082	-0.1082
Pro(R)	0.1854	0.1793	**0.1787**	**0.1787**	**0.1787**	**0.1787**
ProV(G)	**-0.0352**	-0.0357	-0.0357	-0.0357	-0.0357	-0.0357
ProV(R)	**0.0485**	0.0622	0.0622	0.0622	0.0622	0.0622
Dut(G)	0.1164	**-0.0055**	**-0.0055**	**-0.0055**	**-0.0055**	0.1342
Law(R)	0.0149	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
UF(G)	**-0.4148**	-0.5853	-0.5853	-0.5853	-0.5853	-0.5853
AvgRank	2.0	**1.95**	2.11	2.21	2.11	-

Table 10: General ranks of Relab variants according to their average ranks.

	AVERAGE RANKS				
Metrics	(ϵ=0)	(ϵ=0.01)	(ϵ=0.02)	(ϵ=0.03)	(ϵ=0.04)
Accuracy	4	3	2	2	1
Discrimination Score	2	1	3	4	3
AvgRank	2.5	2	2.5	3	2