Fairness-Aware Training of Decision Trees by Abstract Interpretation

Francesco Ranzato, Caterina Urban, Marco Zanella

Fairness-Aware Training of Decision Trees by Abstract Interpretation

Francesco Ranzato, University of Padova, Italy
Caterina Urban, Inria & ENS | PSL, France
Marco Zanella, University of Padova, Italy

Abstract

We study the problem of formally verifying individual fairness of decision tree ensembles, as well as training tree models which maximize both accuracy and individual fairness. In our approach, fairness verification and fairness-aware training both rely on a notion of stability of a classifier, which is a generalization of the standard notion of robustness to input perturbations used in adversarial machine learning. Our verification and training methods leverage abstract interpretation, a well-established mathematical framework for designing computable, correct, and precise approximations of potentially infinite behaviors. We implemented our fairness-aware learning method by building on a tool for adversarial training of decision trees. We evaluated it in practice on the reference datasets in the literature on fairness in machine learning. The experimental results show that our approach is able to train tree models exhibiting a high degree of individual fairness with respect to the natural state-of-the-art CART trees and random forests. Moreover, as a by-product, these fairness-aware decision trees turn out to be significantly compact, which naturally enhances their interpretability.

1 Introduction

Nowadays, machine learning (ML) software assists or even automates decisions with far-reaching socio-economic impact, e.g., for credit scoring [29], recidivism prediction [14], or hiring tasks [41]. The widespread and ever-increasing adoption of ML-based decision-making tools raises concerns on their fairness properties [4, 31]. A number of recent cases have indeed shown that such tools may reproduce, or even reinforce, bias directly or indirectly present in the training data [9, 27, 33]. For this reason, the Artificial Intelligence Act [19] — a first legal framework on machine learning software proposed by the European Commission in April 2021 — imposes strict requirements to minimize the risk of discriminatory outcomes. Our work anticipates the need for methods and tools to facilitate the development of machine learning software that complies with this newly proposed regulation.

Several fairness verification and bias mitigation approaches for machine learning systems have been proposed in recent years, e.g., [1, 23, 39, 40, 43, 45, 46] among the others. However, most works focus on neural networks [39, 40, 43, 45] or on group-based notions of fairness [23, 46], e.g., demographic parity [18] or equalized odds [24]. These notions of group-based fairness require some form of statistical parity (e.g., between positive outcomes) for members of different protected groups (e.g., gender or race). On the other hand, they do not provide guarantees for individuals or other subgroups. By contrast, in this paper we focus on *individual fairness* [18], which intuitively requires that similar individuals in the population receive similar outcomes, and on decision tree ensembles [7, 20], which are commonly used for tabular datasets since they are easily interpretable ML models with high accuracy rates.

1.1 Contributions

We propose an approach for verifying individual fairness of decision tree ensembles, as well as for training tree models which maximize both accuracy and individual fairness.

Our methodology is based on the well-established framework of *abstract interpretation* [15, 16, 38] for designing computable, correct, and precise over-approximations of model computations. In this approach, model computations are over-approximated using an *abstract domain*, which defines a symbolic abstract representation of the concrete quantities being manipulated. Our technique over-approximates computations over input space regions containing similar individuals, and is fully parametric in the choice of the underlying abstract domain. More precisely, we use a product of two abstract domains:

(a) the well-known numerical abstract domain of hyper-rectangles (also known as boxes or intervals) [16], that represents exactly the standard notion of similarity between individuals based on the ℓ_∞ distance metric, and does not lose precision for the univariate split rules of type $x_i \leq k$, with x_i feature and k threshold, the most common split rules used in decision trees;

(b) a specific relational abstract domain which is able to represent precisely one-hot encoded categorical features, such as a feature $color \in \{white, black\}$ which is one-hot encoded by $color_{white}, color_{black} \in \{0, 1\}$.

Our Fairness-Aware Tree Training method, called FATT, is designed as an extension of Meta-Silvae [37], a learning methodology for ensembles of decision trees based on a genetic training algorithm, which is able to train a decision tree for maximizing both its accuracy and its robustness to adversarial perturbations. Meta-Silvae in turn leverages a verification tool for robustness properties of decision trees based on abstract interpretation [36]. We demonstrate the effectiveness of our method FATT in training accurate and fair models on the standard datasets used in the literature on fairness in machine learning. Overall, the experimental results show that our fairness-aware tree models are on average between 35% and 45% more fair than naturally trained decision tree

ensembles at an average cost of -3.6% of accuracy. Moreover, it turns out that our tree models are orders of magnitude more compact and thus naturally easier to interpret. Finally, we show how our fairness-aware models can be used as "hints" for setting the size and shape hyper-parameters (i.e., maximum tree depth and minimum number of samples per leaf) when training standard decision tree models. As a result, this hint-based strategy is capable to output models that are about 20% more fair and just about 1% less accurate than standard models.

1.2 Related Work

The most related work to ours is by Aghaei et al. [1], Raff et al. [34] and Ruoss et al. [40].

By relying on the mixed-integer optimization learning approach by Bertsimas and Dunn [5], Aghaei et al. [1] put forward a framework for training fair decision trees for classification and regression. The experimental evaluation shows that this approach mitigates unfairness as modeled by their notions of disparate impact and disparate treatment at the cost of a significantly higher training computational cost. Their notion of disparate treatment is distance-based and thus akin to individual fairness with respect to the nearest individuals *in a given dataset* (e.g., the k-nearest individuals). In contrast, we consider individual fairness with respect to the nearest individuals *in the input space*, thus including individuals that are not necessarily part of a given dataset.

Raff et al. [34] propose a regularization-based approach for training fair decision trees as well as fair random forests. They consider both group fairness as well as individual fairness with respect to the k-nearest individuals in a given dataset, similarly to Aghaei et al. [1]. In their experiments they use a subset of the datasets that we consider in our evaluation (i.e., the Adult, German, and Health datasets). Our fair models have higher accuracy than theirs (i.e., between 2% and 5.5%) for all but one of these datasets (Health dataset). Interestingly, their models (in particular those with worse accuracy than ours) often have accuracy on par with a constant classifier due to the highly unbalanced label distribution of the datasets. Unfortunately, their tool does not appear to be available for further experimental comparison.

Finally, Ruoss et al. [40] have proposed an approach for learning individually fair data representations and training neural networks (rather than decision tree ensembles as we do) that satisfy individual fairness with respect to a given similarity notion. We use the same notions of similarity in our experiments (cf. Section 6.1).

More broadly, our work fits into the research ecosystem that is nowadays forming around the use of formal methods for the verification of machine learning software. We refer to Liu et al. [30] and Urban et al. [44] for recent surveys of this field.

The rest of the paper is organized as follows. In Section 2 we recall some basic notions on decision tree ensembles and abstract interpretation, while Section 3 provides our setting of individual fairness. In Section 4 we describe our

formal verification method of the individual fairness of tree ensembles, which is then exploited in Section 5 by the fairness-aware tree training algorithm FATT. Section 6 describes our implementation of FATT and the results of our experimental evaluation. Section 7 concludes.

2 Background

2.1 Classifiers

Given an input space $X \subseteq \mathbb{R}^d$ of numerical vectors and a finite set of labels $\mathcal{L} = \{y_1, \ldots, y_m\}$, a *classifier* $C\colon X \to \wp_+(\mathcal{L})$, where $\wp_+(\mathcal{L})$ is the set of all nonempty subsets of $\mathcal{L}$, associates at least one label to every input in X. A *training algorithm* takes as input a dataset $D \subseteq X \times \mathcal{L}$ and outputs a classifier $C_D\colon X \to \wp_+(\mathcal{L})$ which optimizes a given objective function, e.g., the Gini index or the entropy-based information gain for decision trees.

Categorical features can be converted into numerical ones by *one-hot encoding*, where a single feature with $k \geq 2$ possible distinct categories $\{c_1, ..., c_k\}$ is replaced by k new binary features with numerical values in $\{0, 1\}$. Then, each value c_j of the original categorical feature is represented by a bit-value assignment to the new k binary features in which the j-th feature is set to 1 and the remaining $k - 1$ binary features are set to 0.

Classifiers can be evaluated and compared through several metrics. A basic metric is *accuracy* on a test dataset: given a ground truth test dataset $T \subseteq X \times \mathcal{L}$, the accuracy of a classifier C on T is

$$acc_T(C) \triangleq \frac{|\{(\boldsymbol{x}, y) \in T \mid C(\boldsymbol{x}) = \{y\}\}|}{|T|}.$$

According to a growing belief [22], however, accuracy is not enough in machine learning, since robustness to adversarial inputs of a ML classifier may significantly affect its safety and generalization properties [12, 22]. Given an input perturbation modeled by a function $P\colon X \to \wp(X)$, a classifier $C\colon X \to \wp_+(\mathcal{L})$ is *stable* [36] on the perturbation attack $P(\boldsymbol{x})$ of $\boldsymbol{x} \in X$ when C consistently assigns the same label(s) to every attack ranging in $P(\boldsymbol{x})$, i.e.,

$$\text{stable}(C, \boldsymbol{x}, P) \triangleq \forall \boldsymbol{x'} \in P(\boldsymbol{x})\colon C(\boldsymbol{x'}) = C(\boldsymbol{x}). \tag{1}$$

When the sample $\boldsymbol{x} \in X$ has a ground truth label $y_{\boldsymbol{x}} \in \mathcal{L}$, robustness of C on $\boldsymbol{x}$ for attacks in $P(\boldsymbol{x})$ boils down to stability, i.e. $\text{stable}(C, \boldsymbol{x}, P)$ holds, together with correct classification, i.e. $C(\boldsymbol{x}) = \{y_{\boldsymbol{x}}\}$ holds. The most common example of perturbation is induced by the ℓ_∞ norm such that $||\boldsymbol{x}||_\infty = \max\{\boldsymbol{x}_1, ..., \boldsymbol{x}_d\}$, where a given input $\boldsymbol{x} \in X$ is perturbed by considering those vectors $\boldsymbol{x'} \in X$ such that the ℓ_∞ norm of $\boldsymbol{x} - \boldsymbol{x'}$ is less than or equal to a given threshold $\tau \in \mathbb{R}_{\geq 0}$, i.e., this perturbation is defined by $P_\infty(\boldsymbol{x}) \triangleq \{\boldsymbol{x'} \in X \mid ||\boldsymbol{x} - \boldsymbol{x'}||_\infty \leq \tau\}$.

2.2 Decision Trees and Tree Ensembles

We consider standard decision tree classifiers commonly referred to as CARTs (Classification And Regression Trees) [8]. A *decision tree* $t\colon X \to \wp_+(\mathcal{L})$ is defined inductively. A *base tree* t is a single leaf λ storing a (normalized) frequency distribution of labels for the samples of the training dataset, hence $\lambda \in [0,1]^{|\mathcal{L}|}$, or, equivalently, $\lambda\colon \mathcal{L} \to [0,1]$. Some algorithmic rule converts this frequency distribution into a set of labels, typically as $\arg\max_{y\in\mathcal{L}} \lambda(y) \in \wp_+(\mathcal{L})$. A *composite tree* t is $\Gamma(\mathit{split}, t_l, t_r)$, where $\mathit{split}\colon X \to \{\mathbf{tt}, \mathbf{ff}\}$ is a Boolean split criterion for the internal parent node of its left and right subtrees t_l and t_r, respectively; thus, for all $\boldsymbol{x} \in X$, $t(\boldsymbol{x}) \triangleq$ **if** $\mathit{split}(\boldsymbol{x})$ **then** $t_l(\boldsymbol{x})$ **else** $t_r(\boldsymbol{x})$. Although split rules can be of any type, most common decision trees employ univariate hard splits of type $\mathit{split}(\boldsymbol{x}) \triangleq \boldsymbol{x}_i \leq k$, for some feature $i \in [1,d]$ and threshold $k \in \mathbb{R}$.

Tree ensembles, also known as *forests*, are sets of decision trees which together contribute to formulate a unique classification output. Training algorithms as well as methods for computing the final output label(s) vary among different tree ensemble models. Random forests (RFs) [7] are a major instance of tree ensemble where each tree of the ensemble is trained independently from the other trees on a random subset of the features. Gradient boosted decision trees (GBDTs) [20] represent a different training algorithm where an ensemble of trees is incrementally built by training each new tree on the basis of the data samples which are mis-classified by the previous trees. For RFs, the final classification output is typically obtained through a voting mechanism (e.g., majority voting), while GBDTs are usually trained for binary classification problems and use some binary reduction scheme, such as one-vs-all or one-vs-one, for multi-class classification.

2.3 Abstract Interpretation Basics

A *numerical abstract domain* is a tuple $\langle A, \sqsubseteq^A, \gamma^A\rangle$ where: $\langle A, \sqsubseteq^A\rangle$ is at least a preordered set of abstract values and the concretization function $\gamma^A : A \to \wp(\mathbb{R}^d)$, mapping abstract values to sets of numerical vectors, monotonically preserves the ordering relation $\sqsubseteq^A$, that is, $a_1 \sqsubseteq^A a_2$ implies $\gamma^A(a_1) \subseteq \gamma^A(a_2)$. The intuition is that an abstract domain A defines a symbolic abstract representation of sets of vectors ranging in the concrete domain $\wp(\mathbb{R}^d)$. Known examples of numerical abstract domains used in machine learning verification include intervals, zonotopes, and octagons (see the survey [44]).

Given a concrete k-ary operation on vectors $f : (\mathbb{R}^d)^k \to \mathbb{R}^d$, for some $k \in \mathbb{N}$, an abstract function $f^A : A^k \to A$ is called a *sound* (or *correct*) (over-)approximation of f when, for all $(a_1, ..., a_k) \in A^k$, $\{f(\boldsymbol{x}_1, ..., \boldsymbol{x}_k) \mid \forall i.\ \boldsymbol{x}_i \in \gamma^A(a_i)\} \subseteq \gamma^A(f^A(a_1, ..., a_k))$ holds. When equality holds, f^A is defined to be *complete*. In words, this means that soundness holds when $f^A(a_1, ..., a_k)$ never misses a concrete computation of f on some input $(\boldsymbol{x}_1, ..., \boldsymbol{x}_k)$ which is abstractly represented by $(a_1, ..., a_k)$, while completeness implies that each abstract computation $f^A(a_1, ..., a_k)$ is precisely a symbolic abstract representation in A of

the set of computations of f on all the concrete inputs that are abstractly represented by $(a_1, ..., a_k)$.

We will use the well-known abstract domain of not necessarily closed real d-dimensional *hyper-rectangles* HR_d, commonly called interval or box abstraction in abstract interpretation [15, 38]. For concrete vectors ranging in $\mathbb{R}^d$, the abstract values of HR_d are d-dimensional vectors of type

$$h = \langle \boldsymbol{x}_1 \in [l_1, u_1], \ldots, \boldsymbol{x}_d \in [l_d, u_d] \rangle \in \mathrm{HR}_d$$

where lower and upper bounds $l_i, u_i \in \mathbb{R} \cup \{-\infty, +\infty\}$ with $l_i \leq u_i$. Thus, the concretization function is defined by $\gamma^{\mathrm{HR}_d}(h) \triangleq \{\boldsymbol{x} \in \mathbb{R}^d \mid \forall i.\, l_i \leq \boldsymbol{x}_i \leq u_i\}$. Let us recall that $\langle \mathrm{HR}_d, \sqsubseteq^{\mathrm{HR}_d} \rangle$ is a complete lattice for the componentwise ordering: $\langle [l_1, u_1], ..., [l_d, u_d] \rangle \sqsubseteq^{\mathrm{HR}_d} \langle [l_1{}', u_1{}'], ..., [l_d{}', u_d{}'] \rangle$ iff for all i, $l_i{}' \leq l_i$ and $u_i \leq u_i{}'$. More on this abstract domain can be found, e.g., in [15, 38].

3 Individual Fairness

Dwork et al. [18, Section 1.1] define *individual fairness* as "the principle that two individuals who are similar with respect to a particular task should be classified similarly". They formalize this notion as a Lipschitz condition of the classifier, which requires that any two individuals $\boldsymbol{x}, \boldsymbol{y} \in X$ whose distance is $\delta(\boldsymbol{x}, \boldsymbol{y}) \in [0, 1]$, are mapped to distributions $D_{\boldsymbol{x}}$ and $D_{\boldsymbol{y}}$, respectively, such that the distance between $D_{\boldsymbol{x}}$ and $D_{\boldsymbol{y}}$ is at most $\delta(\boldsymbol{x}, \boldsymbol{y})$. The intuition is that the output distributions for $\boldsymbol{x}$ and $\boldsymbol{y}$ are indistinguishable up to their distance. The distance metric $\delta \colon X \times X \to \mathbb{R}_{\geq 0}$ is problem specific, and for their applications Dwork et al. [18, Section 2] study the total variation or relative ℓ_∞ distances.

By following Dwork et al.'s standard definition [18, Section 2], we consider a classifier $C \colon X \to \wp_+(\mathcal{L})$ to be *fair* when C outputs the same set of labels for every pair of individuals $\boldsymbol{x}, \boldsymbol{y} \in X$ which satisfy a similarity relation $S \subseteq X \times X$ between input samples. Thus, S can be derived from any distance δ as $(\boldsymbol{x}, \boldsymbol{y}) \in S \mathrel{\hat{\Leftrightarrow}} \delta(\boldsymbol{x}, \boldsymbol{y}) \leq \tau$, where $\tau \in \mathbb{R}$ is a threshold of similarity. In order to provide a *fairness metric* for a classifier C, we count how often C is fair on sets of similar individuals ranging into a test dataset $T \subseteq X \times \mathcal{L}$:

$$\mathit{fair}_{T,S}(C) \triangleq \frac{|\{(\boldsymbol{x}, y) \in T \mid \mathrm{fair}(C, \boldsymbol{x}, S)\}|}{|T|} \tag{2}$$

where $\mathrm{fair}(C, \boldsymbol{x}, S)$ is defined as follows:

Definition 1 (**Individual fairness**)**.** A classifier $C \colon X \to \wp_+(\mathcal{L})$ is *fair* on an individual $\boldsymbol{x} \in X$ with respect to a similarity relation $S \subseteq X \times X$, denoted by $\mathrm{fair}(C, \boldsymbol{x}, S)$, when

$$\forall \boldsymbol{x}' \in X \colon (\boldsymbol{x}, \boldsymbol{x}') \in S \Rightarrow C(\boldsymbol{x}') = C(\boldsymbol{x}). \qquad \square$$

Hence, fairness for a similarity relation S boils down to stability on the perturbation $P_S(\boldsymbol{x}) \triangleq \{\boldsymbol{x}' \in X \mid (\boldsymbol{x}, \boldsymbol{x}') \in S\}$ (cf. Equation (1) in Section 2.1),

namely, for all $\boldsymbol{x} \in X$,

$$\text{fair}(C, \boldsymbol{x}, S) \;\Leftrightarrow\; \text{stable}(C, \boldsymbol{x}, P_S). \tag{3}$$

Let us remark that fairness is orthogonal to accuracy since it does not depend on the correctness of the label assigned by the classifier, so that training algorithms that maximize accuracy-based metrics do not necessarily achieve fair models. Thus, this is also the case of a natural learning algorithm for CART trees and random forests, that locally optimizes split criteria by measuring entropy or Gini impurity, which are both indicators of the correct classification of training data.

It is also worth observing that fairness is anti-monotonic w.r.t. the logical implication of similarity relations, meaning that

$$\text{fair}(C, \boldsymbol{x}, S) \wedge S' \subseteq S \;\Rightarrow\; \text{fair}(C, \boldsymbol{x}, S'). \tag{4}$$

We will exploit this anti-monotonicity property, since it implies that, on one hand, fair classification is preserved for finer similarity relations and, on the other hand, fairness verification and fair training is more challenging for coarser similarity relations.

4 Verifying Individual Fairness

As individual fairness is equivalent to stability, we verify individual fairness of ensembles of decision trees by means of Silva [36], an abstract interpretation-based algorithm for checking stability properties of decision tree ensembles.

4.1 Sound and Complete Verification

Silva performs a static analysis of an ensemble of decision trees leveraging an abstract domain A that approximates computations on real vectors. Each abstract value $a \in A$ symbolically represents a set of real vectors $\gamma^A(a) \in \wp(\mathbb{R}^d)$. Then, Silva over-approximates an input region $P(\boldsymbol{x}) \in \wp(\mathbb{R}^d)$ for an input vector $\boldsymbol{x} \in \mathbb{R}^d$ by an abstract value $a \in A$ such that $P(\boldsymbol{x}) \subseteq \gamma^A(a)$, and for each decision tree t, it computes an over-approximation of the set of leaves of t that can be reached from some vector in $\gamma^A(a)$. This is computed by collecting the constraints of all the split nodes in each root-leaf path of t, so that each leaf λ of t stores the minimum set of constraints C_λ that makes λ reachable from the root of t. The verification algorithm then checks whether this set of constraints C_λ can be satisfied by the input abstract value $a \in A$: this check $a \models^? C_\lambda$ must be *sound*, meaning that if some input sample $\boldsymbol{x} \in X$ abstractly represented by a, i.e. $\boldsymbol{x} \in \gamma^A(a)$, actually reaches the leaf λ, then $a \models C_\lambda$ must necessarily hold. When $a \models C_\lambda$ holds the leaf λ is marked as reachable from the abstract value a.

Example 1 (**Sound verification**). *Consider the scenario in Fig. 1 where the set of split constraints for some leaf λ is $C_\lambda = \{x_1 \leq 2, \neg(x_1 \leq -1),\ x_2 \leq -1\}$,*

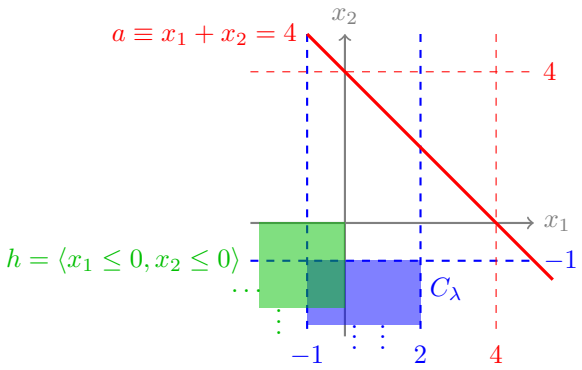


Figure 1: An example of abstract check $a \models^? C_\lambda$

depicted in blue. An hyper-rectangle abstract value such as $h = \langle x_1 \leq 0, x_2 \leq 0 \rangle \in \mathrm{HR}_2$*, depicted in green, abstractly represents some points which satisfy all the constraints in* C_λ*, so that* $a \models C_\lambda$ *must hold. On the other hand, for a relational abstract value such as the line* $a \equiv x_1 + x_2 = 4$*, depicted in red, we have that no point in the line satisfies* C_λ*, so that no sure information could be inferred from a merely sound check of* $a \models^? C_\lambda$*.* □

This sound approach therefore provides an over-approximation of the set of leaves of t reachable from an abstract value a that allows us to compute a set of labels $t^A(a) \in \wp_+(\mathcal{L})$ which is an over-approximation of the actual set of labels assigned by t to all the input vectors ranging in $\gamma^A(a)$, i.e.,

$$\cup_{\boldsymbol{x} \in \gamma^A(a)} t(\boldsymbol{x}) \subseteq t^A(a) \tag{5}$$

holds. Thus, given a similarity relation $S \subseteq X \times X$, it turns out that for all $\boldsymbol{x} \in X$ and $a \in A$,

$$P_S(\boldsymbol{x}) \subseteq \gamma^A(a) \ \wedge \ t^A(a) = t(\boldsymbol{x}) \ \Rightarrow \ \mathrm{fair}(t, \boldsymbol{x}, S),$$

meaning that we have a sound verification method for individual fairness of decision trees.

For the most common classification trees with hard univariate splits of type $x_i \leq k$, it turns out that the abstract domain HR_d guarantees that for each leaf constraint C_λ and hyper-rectangle $h \in \mathrm{HR}_d$, the check $h \models^? C_\lambda$ is sound and *complete*, meaning that

$$h \models C_\lambda \ \Leftrightarrow \ \exists \boldsymbol{x} \in \gamma(h) \text{ reaching } \lambda.$$

This noteworthy completeness property of the hyper-rectangle abstraction entails that the set of labels $t^{\mathrm{HR}}(h)$ computed by the analysis on HR_d coincides *exactly* with the set of classification labels computed by t for all the samples in $\gamma^{\mathrm{HR}_d}(h)$, i.e., the set inclusion in Equation (5) is strengthened to a set equality.

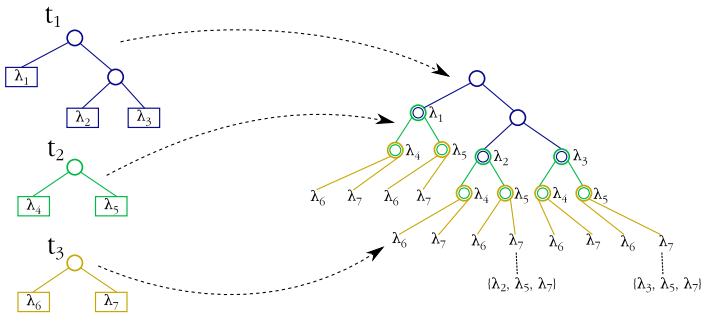


Figure 2: An example of forest to single tree reduction

Thus, for the ℓ_∞-based similarity relation $S^\tau_\infty \subseteq \mathbb{R}^d \times \mathbb{R}^d$ between individuals, for each input $\boldsymbol{x} \in \mathbb{R}^d$ there exists an hyper-rectangle $h_{\boldsymbol{x}} \in \mathrm{HR}_d$ such that $S^\tau_\infty(\boldsymbol{x}) = \{\boldsymbol{x'} \in X \mid ||\boldsymbol{x} - \boldsymbol{x'}||_\infty \leq \tau\} = \gamma^{\mathrm{HR}_d}(h_{\boldsymbol{x}})$ holds and, in turn, we have that

$$t^{\mathrm{HR}}(h_{\boldsymbol{x}}) = t(\boldsymbol{x}) \;\Leftrightarrow\; \mathrm{fair}(t, \boldsymbol{x}, S^\tau_\infty). \tag{6}$$

Hence, we devised a *complete verification method* for individual fairness with respect to ℓ_∞-based similarity.

Verification of Forests.

In order to analyse a forest F of trees, Silva reduces the whole forest to a single tree t_F, by stacking every tree $t \in F$ on top of each other, i.e., each leaf becomes the root of the next tree in F, where the ordering of this stacking operation does not affect the output of a forest verification. Then, each leaf λ of this stacked single tree t_F collects all the constraints of the leaves of trees in the path from the root of t_F to λ. An example is shown in Fig. 2, where the trees t_1, t_2, t_3 are combined into a single tree, emphasizing how the leaves of the original trees t_i become internal nodes in this stacked tree t_F, while their sets of constraints are collected and stored in the new leaves of t_F.

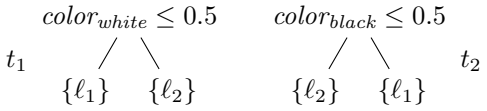
Since the stacked tree t_F suffers from a combinatorial explosion of the number of leaves, Silva deploys a number of optimisation strategies for performing its abstract interpretation. The most relevant ones include a best-first search algorithm to look for a pair of input samples in $\gamma^A(a)$ which are differently labeled. If one such pair $\boldsymbol{x}, \boldsymbol{x'} \in \gamma^A(a)$ can be found then unfairness of the forest F in classifying $\boldsymbol{x}$ and $\boldsymbol{x'}$ is proved and the analysis is terminated, otherwise fairness holds. A further optimisation consists in combining sound analyses with complete ones. While exploring an internal node n, a fast sound but possibly incomplete analysis of t_F is used to estimate a superset of the leaves of t_F reachable from n, which, in turn, allows us to compute a superset of the output labels.

If such superset consists of a single label, then every leaf reachable from n necessarily yields that same label, and the rest of the visit of the subtree rooted in n can be safely omitted; otherwise, that information provided by the node n is nonetheless used to refine the analysis of t_F. We refer to [36] for further details on Silva.

4.2 Verification with One-Hot Enconding

As described above by the equivalence (6), when the similarity relation is defined by the ℓ_∞ norm and the abstract domain of hyper-rectangles HR_d is used, our verification method for individual fairness turns out to be complete. However, completeness does not hold for classification problems involving categorical features, as shown by the following simple example.

Example 2 (**Incompleteness of one-hot encoding**). *The following diagram depicts a toy forest F consisting of two trees t_1 and t_2, where left/right branches are followed when the split condition is false/true.*



Here, a categorical feature $color \in \{white, black\}$ is one-hot encoded by $color_{white}, color_{black} \in \{0, 1\}$. Since colors are mutually exclusive, every white individual in the input space, i.e., a vector $\langle color_{white} = 1, color_{black} = 0\rangle$, will be labeled ℓ_1 by both trees t_1 and t_2. Our fairness verification method on the input ℓ_∞-based similarity relation $\langle color_{white} \in [0, 1], color_{black} \in [0, 1]\rangle = [0, 1] \times [0, 1]$, which is an hyper-rectangle, infers that the forest F is unfair on the individual $in = \langle color_{white} = 1, color_{black} = 0\rangle$ because there exists an input sample $cex \in [0, 1] \times [0, 1]$ such that $F(cex) = \{\ell_1, \ell_2\} \neq \{\ell_1\} = F(in)$. This is due to the counterexample $cex = \langle 0, 0\rangle \in [0, 1]^2$ which is therefore similar to in, although it does not represent any actual individual in the input space. Indeed, $t_1(cex) = \{\ell_2\}$ and $t_2(cex) = \{\ell_1\}$, so that, by a majority voting, $F(cex) = \{\ell_1, \ell_2\}$, thus making F unfair on in, i.e., unfair on white individuals. □

To overcome this issue with one-hot encoding, we use a refined abstract domain which is designed as *reduced product*[1] of the underlying main abstract domain, in our case the hyper-rectangle abstraction HR_d, with a relational abstract domain that keeps track of the relationships among the multiple binary features introduced by one-hot encoding categorical features. More formally, this relational abstract domain maintains the following two additional constraints on the k features $x^c_1, ..., x^c_k$ introduced by one-hot encoding a categorical variable x^c with k distinct values:

(a) the possible values for each x^c_i are restricted to the discrete set $\{0, 1\}$ (rather than ranging in the continuous interval $[0, 1]$);

[1] Readers can find more details on reduced products of abstract domains, e.g., in [15].

(b) the sum of all the x_i^c's must satisfy the relational constraint $\sum_{i=1}^{k} x_i^c = 1$.

Therefore, these conditions (a)-(b) guarantee that any abstract value for $x_1^c, ..., x_k^c$ represents precisely a feasible category for x^c. This abstract domain for a categorical variable x with k distinct values is denoted by $\mathrm{OH}_k(x)$. In the example above, any hyper-rectangle $\langle color_{white} \in [0,1], color_{black} \in [0,1] \rangle$ is reduced by $\mathrm{OH}_2(color)$, so that just two different abstract values $\langle color_{white} = 0, color_{black} = 1 \rangle$ and $\langle color_{white} = 1, color_{black} = 0 \rangle$ are allowed.

Summing up, we employed a reduced hyper-rectangle abstract domain, denoted by $\mathrm{HR}_d \otimes \mathrm{OH}$, whose generic abstract value for data vectors consisting of d numerical variables $x^j \in \mathbb{R}$ and m categorical variables c^j with $k_j \in \mathbb{N}$ distinct categories is:

$$\langle x^j \in [l_j, u_j] \rangle_{j=1}^{d} \times \langle c_i^j \in \{0,1\} \mid \textstyle\sum_{i=1}^{k_j} c_i^j = 1 \rangle_{j=1}^{m} \in \mathrm{HR}_d \otimes \mathrm{OH},$$

where $l_j, u_j \in \mathbb{R} \cup \{-\infty, +\infty\}$ and $l_j \leq u_j$.

5 Fairness-Aware Training of Trees

Several algorithms for training robust decision trees and ensembles thereof have been put forward in the literature [2, 10, 11, 13, 26, 37]. These algorithms encode the robustness of a tree classifier as a loss function which is minimized either by exact methods such as MILP or by suboptimal heuristics such as genetic algorithms.

The robust learning algorithm Meta-Silvae, introduced in [37], aims at maximizing a tunable weighted linear combination of accuracy and stability metrics, therefore, thanks to the equivalence (3), instantiating to a combination of accuracy and individual fairness for our purposes. Meta-Silvae relies on a genetic algorithm for evolving a population of decision trees t which are ranked by their accuracy $acc_D(t)$ and individual fairness $fair_{D,S}(t)$ w.r.t. a training dataset D, where the individual fairness metric $fair_{D,S}(t)$ for a similarity relation S is computed by the verification method described in Section 4.1. At the end of the genetic evolution, the best tree (or best trees are) is returned. It turns out that Meta-Silvae typically outputs significantly compact models and often achieves high accuracy and fairness already with a single decision tree rather than a forest of them. Thus, the instantiation of Meta-Silvae to the reduced product abstract domain $\mathrm{HR}_d \otimes \mathrm{OH}$ described in Section 4.2 provides a learning algorithm for ensembles of decision trees, called FATT, that enhances their individual fairness.

More specifically, the objective function φ of FATT, for learning fair decision trees, is given by a weighted sum of the accuracy and individual fairness scores over the training dataset D, i.e.,

$$\varphi_D(t) \triangleq w_a \, acc_D(t) + w_f \, fair_{D,S}(t). \tag{7}$$

We use $w_a = 0.9$ and $w_f = 0.1$ for all our experiments (presented in the next section).

While standard learning algorithms for tree ensembles require tuning a number of hyper-parameters, such as maximum depth of trees, minimum amount of information on leaves and maximum number of trees, FATT is able to infer them automatically, so that the customary tuning process is not needed. Instead, some standard parameters are required by the underlying genetic algorithm, notably, the size of the evolving population, the maximum number of evolving iterations, the selection of crossover and mutation functions [25, 42]. It should be remarked that, given an objective function, the genetic algorithm of FATT converges to an optimal (or suboptimal) solution regardless of the chosen parameters of the genetic algorithm, which just affect the rate of convergence and therefore should be chosen for tuning its convergence speed only.

Crossover and mutation functions are the two main features of the genetic algorithm of FATT. The crossover function combines two parent trees t_1 and t_2 of the current evolving population by randomly substituting a subtree of t_1 with a subtree of t_2. After a crossover operation, two types of mutation strategies are available: grow-only, which only allows trees to grow, and grow-and-prune, which also allows pruning the mutated trees. We just sketch an example of crossover and mutation functions and refer to [37] for a detailed definition.

Example 3 (**Crossover and mutation**). *Fig. 3 depicts a simple example of crossover, where every node is represented by a tuple (i, k, R), with i, k denoting a split $x_i \leq k$ (therefore not relevant for leaves), while $R \subseteq X$ represents the subset of samples in the training set reaching that node. The crossover tree t_{cross} is built by selecting the subtree s_1 of t_1 given by the leaf $(\_, \_, R_1)$ only, the subtree s_2 of t_2 rooted at node $(2, 1, R_5)$, and by replacing s_1 with s_2 in t_1. Selection of both subtrees can be fully random, or relies on some mixed heuristic/stochastic criteria, for example by replacing a poorly performing subtree with a better one from another tree, where subtrees are stochastically selected with a probability proportional to their Gini impurity. While the split constraints $x_i \leq k$ are directly copied from the donor tree, the set R of reaching samples may well be different, and must therefore be recomputed. After a crossover operation, it may happen that the set of reachable samples R of some node n of t_{cross} becomes empty, thus making n unreachable. In this case, the tree is pruned by removing these unreachable nodes and by replacing their parents with the siblings. In Fig. 3, t_{cross}^{prune} is the result of pruning the leaf $(\_, \_, R_3 = \varnothing)$ of t_{cross}.*
Fig. 3 also depicts two mutations of t_{cross}^{prune}: t_{mute_1} is obtained from t_{cross}^{prune} by expanding the leaf $(\_, \_, R_9)$, while t_{mute_2} prunes the subtree rooted at $(2, 1, R_1)$. In both cases, the leaf to grow or the subtree to prune can be selected randomly or stochastically with a probability proportional to Gini impurity. □

FATT

Let us recall that the standard learning method for CART trees [8] is a greedy algorithm which incrementally builds a decision tree by locally computing new split nodes or new leaves, where a new split $x_i \leq k$ is inferred by testing every possible combination of i and k in order to locally minimize a Gini impurity

or entropy index. It is also worth recalling that while this greedy learning approach yields trees which are very accurate on the training set, it often leads to overfitting, a well-known phenomenon with decision trees [6].

FATT relies on an objective function that takes into account both accuracy and individual fairness, cf. Equation (7). Thus, in FATT the selection of a candidate split relies on an individual fairness index $fair_{D,S}(t)$ computed by the verifier described in Section 4.1 on the whole corresponding candidate tree t. This learning process is inherently not incremental and consequently could be computationally burdensome. Such a computational cost is alleviated by introducing a *search aggressiveness* parameter which sets up a bound on the number of new split candidates to consider or a timeout mechanism. It turns out that this optimization is effective in reducing the computational burden without sacrificing the overall generalization, because the evaluation of split candidates which are ruled out is simply delayed to later iterations of the evolutionary process.

Our fair learning method FATT uses the following basic parameters for the underlying genetic algorithm Meta-Silvae: the population size is kept fixed to 32, as our experimental evaluation showed that this provides an effective balance between achieved individual fairness and training time; the standard roulette wheel algorithm [32] is employed as selection function (which is proportional to the objective function φ); the number of iterations of the evolutionary process is typically dataset-specific and can be estimated by running a preliminary analysis of the convergence speed, for instance by stopping after a given number of iterations with no improvement.

6 Experimental Evaluation

We consider the main standard datasets used in the fairness literature [31] and we preprocess them by following the steps of Ruoss et al. [40, Section 5] for their experiments on individual fairness for deep neural networks: (1) standardize numerical attributes to zero mean and unit variance; (2) one-hot encoding of all categorical features; (3) drop rows/columns containing missing values; (4) split into train and test set. These datasets concern binary classification tasks, although our fair learning algorithm naturally extends to multiclass classification with no specific effort. The source code of FATT as well as all the datasets and preprocessing pipelines are publicly available on GitHub [35].

Adult. The Adult income dataset [17] is extracted from the 1994 US Census database. Every sample assigns a yearly income (below or above 50K US$) to an individual based on personal attributes such as gender, race, and occupation.

Compas. The COMPAS dataset contains data collected on the use of the COMPAS risk assessment tool in Broward County, Florida [3]. Each sample predicts the risk of recidivism for individuals based on personal attributes and criminal history.

Crime. The Communities and Crime dataset [17] contains socio-economic, law enforcement, and crime data for communities within the US. Each sample indicates whether a community is above or below the median number of violent crimes per population.

German. The German Credit dataset [17] contains samples assigning a good or bad credit score to individuals.

Health. The heritage Health dataset (https://www.kaggle.com/c/hhp) contains physician records and insurance claims. Each sample predicts the ten-year mortality (above or below the median Charlson index) for a patient.

The following table displays size and distribution of positive samples for these datasets.

Dataset	#features	**Training Set**		**Test Set**	
		Size	Positive	Size	Positive
Adult	103	30162	24.9%	15060	24.6%
Compas	371	4222	53.3%	1056	55.6%
Crime	147	1595	50.0%	399	49.6%
German	56	800	69.8%	200	71.0%
Health	110	174732	68.1%	43683	68.0%

As noticed by Ruoss et al. [40], some datasets exhibit a highly unbalanced label distribution. For example, for the Adult dataset, a constant classifier $C(\boldsymbol{x}) = 1$ would achieve 75.4% test set accuracy and 100% individual fairness with respect to any similarity relation. Hence, we follow [40] and we additionally evaluate and report the *balanced accuracy* of our FATT classifiers, i.e.,

$$0.5\,(\frac{\mathit{truePositive}}{\mathit{truePositive} + \mathit{falseNegative}} + \frac{\mathit{trueNegative}}{\mathit{trueNegative} + \mathit{falsePositive}}).$$

6.1 Similarity Relations

Following Ruoss et al. [40, Section 5.1], we consider four different types of similarity relations. In the following, let $I \subseteq \mathbb{N}$ denote the set of indexes of features of an individual after one-hot encoding.

NOISE: Two individuals $\boldsymbol{x}, \boldsymbol{y} \in X$ are similar when a subset of their (standardized) numerical features indexed by a given subset $I' \subseteq I$ differs less than a given threshold $\tau \geq 0$, while all the other features are unchanged: $(\boldsymbol{x}, \boldsymbol{y}) \in S_{noise}$ iff $|\boldsymbol{x}_i - \boldsymbol{y}_i| \leq \tau$ for all $i \in I'$, and $\boldsymbol{x}_i = \boldsymbol{y}_i$ for all $i \in I \setminus I'$. For our experiments, we consider $\tau = 0.3$ in the standardized input space, e.g., for Adult two individuals are similar if their age difference is at most 3.95 years.

CAT: Two individuals are similar if they are identical except for one or more categorical sensitive attributes indexed by $I' \subseteq I$: $(\boldsymbol{x}, \boldsymbol{y}) \in S_{cat}$ iff $\boldsymbol{x}_i = \boldsymbol{y}_i$ for all $i \in I \setminus I'$. For Adult and German, we select the gender attribute. For Compas, we identify race as sensitive attribute. For Crime, we consider two individuals similar regardless of their state. Lastly, for Health, neither gender nor age group should affect the final prediction.

NOISE-CAT: Given noise and categorical similarity relations S_{noise} and S_{cat}, their union $S_{noise\text{-}cat} \triangleq S_{noise} \cup S_{cat}$ models a relation where two individuals are similar when some of their numerical attributes differ up to a given threshold while the other attributes are equal except some categorical features.

CONDITIONAL-ATTRIBUTE: Here, similarity is a disjunction of two mutually exclusive cases. Consider a numerical attribute $\boldsymbol{x}_i$, a threshold $\tau \geq 0$ and two noise similarities S_{n_1}, S_{n_2}. Two individuals are defined to be similar if either their i-th attributes are similar for S_{n_1} and are bounded by τ or these attributes are above τ and similar for S_{n_2}: $S_{cond} \triangleq \{(\boldsymbol{x}, \boldsymbol{y}) \in S_{n_1} \mid \boldsymbol{x}_i \leq \tau, \boldsymbol{y}_i \leq \tau\} \cup \{(\boldsymbol{x}, \boldsymbol{y}) \in S_{n_2} \mid \boldsymbol{x}_i > \tau, \boldsymbol{y}_i > \tau\}$. For Adult, we consider the median age as threshold $\tau = 37$, and two noise similarities based on age with thresholds 0.2 and 0.4, which correspond to age differences of 2.63 and 5.26 years, respectively. For German, we also consider the median age $\tau = 33$ and the same noise similarities on age, that correspond to age differences of 0.24 and 0.47 years.

It is worth remarking that our fairness-aware learning algorithm is not limited to support these similarity relations. Further domain-specific similarities can be defined and handled by our approach by simply instantiating the underlying fairness verifier with a suitable over-approximating abstract domain A to retain a sound verification. Moreover, if the similarity relation can be precisely represented in A, completeness is preserved as well.

6.2 Setup

Our experimental evaluation compares CART trees and Random Forests with our FATT tree models. CARTs and RFs are trained by scikit-learn. We first run a preliminary phase for tuning the hyper-parameters for CARTs and RFs. In particular, we considered both entropy-based information gain and Gini index as split criteria, and we checked maximum tree depths ranging from 5 to 100 with step 10. For RFs, we scanned the maximum number of trees ranging from 5 to 100, step 10. Cross validation inferred the optimal hyper-parameters, where the datasets have been split in 80% training and 20% validation sets. The hyper-parameters of FATT (i.e., weights of accuracy w_a and fairness w_f in the objective function (7), type of mutation strategy, number of iterations) are inferred by assessing convergence speed, maximum fitness value and variance among fitness in the evolving population during the training phase. We report here the best results, obtained with $w_a = 0.9$ and $w_f = 0.1$. The number of

Dataset	Accuracy %		Balanced Accuracy %		Individual Fairness $fair_T$ %							
					CAT		NOISE		NOISE-CAT		CONDITIONAL-ATTRIBUTE	
	RF	FATT	RF	FATT	RF	FATT	RF	FATT	RF	FATT	RF	FATT
Adult	82.76	80.84	70.29	61.86	91.71	100.00	85.44	95.21	77.50	95.21	84.75	94.12
Compas	66.57	64.11	66.24	63.83	48.01	100.00	35.51	85.98	30.87	85.98	-	-
Crime	80.95	79.45	80.98	79.43	86.22	100.00	31.83	75.19	32.08	75.19	-	-
German	76.50	72.00	63.62	52.54	91.50	100.00	92.00	99.50	90.00	99.50	91.50	99.50
Health	85.29	77.87	83.27	73.59	7.84	99.99	47.66	97.04	2.91	97.03	-	-
Average	**78.41**	74.85	**72.88**	66.25	65.06	**100.00**	58.49	**90.58**	46.67	**90.58**	88.13	**96.81**

Table 1: RF and FATT Comparison

evolutionary iterations for the experiments ranges between 10 and 100. Specifically, we obtained the best results with 10 iterations for the Crime and German datasets, and with 100 iterations for Adult, Compas, and Health. It turned out that accuracy and fairness of single FATT trees, rather than forests, were already competitive, with individual fairness exceeding 85% for the most challenging similarities. We therefore concluded that ensembles of FATT trees do not introduce statistically significant benefits over single decision trees. Since FATT trees are stochastic, by relying on random seeds, each experimental test has been repeated 1000 times and the reported results refer to their median value.

6.3 Results

Table 1 provides a comparison between RF and FATT models by showing accuracy, balanced accuracy and individual fairness with respect to the four similarity relations of Section 6.1 computed on the test sets T. The CONDITIONAL-ATTRIBUTE similarity relation only applies to the Adult and German datasets. As expected, FATT decision trees are slightly less accurate than RFs — about -3.6% on average, which also reflects to balanced accuracy — but outperform RFs in every fairness test. On average, the fairness increment ranges between $+32.09\%$ to $+43.91\%$ among the NOISE, CAT, and NOISE-CAT similarity relations. For the CONDITIONAL-ATTRIBUTE similarity the average fairness increase of FATT models is $+8.68\%$.

Fig. 4 shows the distribution of accuracy and individual fairness for FATT trees over 1000 runs of the FATT learning algorithm. This boxplot is for fairness with respect to NOISE-CAT similarity, as this is the most challenging relation for achieving individual fairness, as a consequence of anti-monotonicity (4). We observe a stable behaviour for accuracy, with about 50% of the observations laying within one percentile from the median. The results for fairness are similar, although for Compas we report a higher variance of the distribution, where the lowest fairness percentage is about 10% higher than the corresponding one for RFs. We claim that this may depend on the high number of features in the dataset Compas, which makes fair training a challenging task.

Table 2 compares the size of RF and FATT models, defined as total number of leaves, and their average verification time of individual fairness per input

sample. It turns out that FATT tree models are orders of magnitude smaller and, therefore, more interpretable than RFs, while the average verification time per sample for our FATT models is always negligible (< 0.1 milliseconds).

Dataset	Model Size		Average Verification Time per Sample (ms)							
			CAT		NOISE		NOISE-CAT		CONDITIONAL-ATTRIBUTE	
	RF	FATT	RF	FATT	RF	FATT	RF	FATT	RF	FATT
Adult	1427	43	0.03	0.02	0.03	0.02	0.03	0.02	0.03	0.02
Compas	147219	75	0.36	0.07	0.47	0.07	0.61	0.07	-	-
Crime	14148	11	0.12	0.07	2025.13	0.07	2028.47	0.07	-	-
German	5743	2	0.06	0.03	0.06	0.02	0.07	0.03	0.06	0.02
Health	2558676	84	1.40	0.06	0.91	0.05	3.10	0.06	-	-

Table 2: Model Sizes and Verification Times

Dataset	FATT			Natural CART			CART with Hints		
	Accuracy %	Fairness %	Size	Accuracy %	Fairness %	Size	Accuracy %	Fairness %	Size
Adult	80.84	95.21	43	85.32	77.56	270	84.77	87.46	47
Compas	64.11	85.98	75	65.91	22.25	56	65.91	22.25	56
Crime	79.45	75.19	11	77.69	24.31	48	77.44	60.65	8
German	72.00	99.50	2	75.50	57.50	115	73.50	86.00	4
Health	77.87	97.03	84	83.85	79.98	2371	82.25	93.64	100
Average	74.85	**90.58**	**43**	**77.65**	52.32	572	76.77	70.00	**43**

Table 3: Decision Trees Comparison

Finally, Table 3 compares FATT models with natural CART trees in terms of accuracy, individual fairness with respect to the NOISE-CAT similarity, and size. While CARTs are approximately 3% more accurate than FATT models on average, they are roughly half less fair and more than $10\times$ larger.

CART with Hints

As already recalled, decision trees frequently overfit [6] due to their high number of leaves, thus yielding unstable/unfair models. Post-training techniques, such as tree pruning, are commonly used to mitigate overfitting [28], although they are deployed when a tree has been already fully trained and, therefore, pruning is often poorly beneficial. As a byproduct of our fairness-aware learning approach, we trained a set of natural CART trees, denoted by CART with Hints in Table 3, which exploit hyper-parameters as "hinted" by our FATT training. In particular, in this learning of CART trees with hints, the maximum tree depth and the minimum number of samples per leaf are obtained as tree depth and minimum number of samples of our best FATT models. Interestingly, the results in Table 3 show that these decision trees with hints have roughly the same size of our FATT trees, are approximately 20% more fair than natural CART trees and just 1% less accurate. Overall, it turns out that the general performance of these CARTs with hints is halfway between natural CARTs and FATT trees, both in term of accuracy and individual fairness, while having the same compactness of FATT models.

7 Conclusion

We believe that this work contributes to push forward the use of formal verification methods in decision tree learning, in particular a well known program analysis technique such as abstract interpretation is proved to be successful for training and verifying decision tree classifiers which are both accurate and fair, improve on state-of-the-art CART and random forest models, while being much more compact and thus interpretable. We also showed how information from our FATT trees can be exploited to tune the natural training process of decision trees.

As future work we plan to extend our fairness verification and learning method by considering alternative fairness definitions, such as group or statistical fairness, or stronger notions such as causal [21] or dependency [43] fairness. We also aim at designing *quantitative* verification methods for both stability and fairness, in order to provide probabilistic guarantees on the behavior of decision tree and tree ensemble models.

Acknowledgments. Francesco Ranzato has been partially funded by: *University of Padova*, under the SID2018 project "Analysis of STatic Analyses (ASTA)"; *Italian Ministry of University and Research*, under the PRIN2017 project no. 201784YSZ5 "AnalysiS of PRogram Analyses (ASPRA)"; *Facebook Research*, under a "Probability and Programming Research Award".

References

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

[2] Maksym Andriushchenko and Matthias Hein. Provably Robust Boosted Decision Stumps and Trees against Adversarial Attacks. In *Proc. 33rd Annual Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine Bias. *ProPublica, May*, 23:2016, 2016.

[4] Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. *California Law Review*, 104:671, 2016.

[5] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Mach. Learn.*, 106(7):1039–1082, 2017.

[6] Max Bramer. *Avoiding Overfitting of Decision Trees*, pages 121–136. Springer, 2013.

[7] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[8] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[9] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency (FAT 2018)*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 2018.

[10] Stefano Calzavara, Claudio Lucchese, and Gabriele Tolomei. Adversarial Training of Gradient-Boosted Decision Trees. In *Proc. 28th ACM International Conference on Information and Knowledge Management (CIKM 2019)*, pages 2429–2432, 2019.

[11] Stefano Calzavara, Claudio Lucchese, Gabriele Tolomei, Seyum Assefa Abebe, and Salvatore Orlando. TREANT: training evasion-aware decision trees. *Data Mining and Knowledge Discovery*, 2020.

[12] Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *Proc. of 38th IEEE Symposium on Security and Privacy (S & P 2017)*, pages 39–57, 2017.

[13] Hongge Chen, Huan Zhang, Duane S. Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. In *Proc. 36th Int. Conf. on Machine Learning, (ICML 2019)*, pages 1122–1131, 2019.

[14] Alexandra Chouldechova. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2):153–163, 2017.

[15] Patrick Cousot. *Principles of Abstract Interpretation*. MIT Press, 2021.

[16] Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. 4th ACM Symposium on Principles of Programming Languages (POPL 1977)*, pages 238–252, 1977.

[17] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017.

[18] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012*, pages 214–226. ACM, 2012.

[19] European Commission. Proposal for a Regulation Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence-artificial-intelligence, April 2021.

[20] Jerome H Friedman. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, pages 1189–1232, 2001.

[21] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness Testing: Testing Software for Discrimination. In *FSE*, pages 498–510, 2017.

[22] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making Machine Learning Robust Against Adversarial Inputs. *Commun. ACM*, 61(7):56–66, 2018.

[23] Vincent Grari, Boris Ruf, Sylvain Lamprier, and Marcin Detyniecki. Achieving Fairness with Decision Trees: An Adversarial Approach. *Data Sci. Eng.*, 5(2):99–110, 2020.

[24] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Proc. 30th Annual Conference on Neural Information Processing Systems (NeurIPS 2016)*, pages 3315–3323, 2016.

[25] John H Holland. Genetic algorithms and adaptation. In *Adaptive Control of Ill-Defined Systems*, pages 317–333. Springer, 1984.

[26] Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and Hardening of Tree Ensemble Classifiers. In *Proc. 33rd International Conference on Machine Learning (ICML 2016)*, pages 2387–2396, 2016.

[27] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI 2015)*, pages 3819–3828. ACM, 2015.

[28] Michael J. Kearns and Yishay Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, pages 269–277. Morgan Kaufmann, 1998.

[29] Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer Credit-Risk Models via Machine-Learning Algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010.

[30] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher A. Strong, Clark W. Barrett, and Mykel J. Kochenderfer. Algorithms for verifying deep neural networks. *Found. Trends Optim.*, 4(3-4):244–404, 2021.

[31] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), July 2021.

[32] Frank Neumann, Pietro Simone Oliveto, and Carsten Witt. Theoretical analysis of fitness-proportional selection: Landscapes and efficiency. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO 2009)*, page 835–842. ACM, 2009.

[33] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366:447–453, 2019.

[34] Edward Raff, Jared Sylvester, and Steven Mills. Fair forests: Regularized tree induction to minimize model bias. In *Proc. 1st AAAI/ACM Conference on AI, Ethics, and Society (AIES 2018)*, page 243–250, 2018.

[35] Francesco Ranzato, Caterina Urban, and Marco Zanella. FATT: Fairness aware tree training. https://github.com/fatt21/fatt, 2021.

[36] Francesco Ranzato and Marco Zanella. Abstract interpretation of decision tree ensemble classifiers. In *Proc. 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, pages 5478–5486, 2020.

[37] Francesco Ranzato and Marco Zanella. Genetic adversarial training of decision trees. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, page 358–367, New York, NY, USA, 2021. Association for Computing Machinery.

[38] Xavier Rival and Kwangkeun Yi. *Introduction to Static Analysis: An Abstract Interpretation Perspective*. The MIT Press, 2020.

[39] Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. Fr-train: A mutual information-based approach to fair and robust training. In *Proc. of the 37th Int. Conf. on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pages 8147–8157. PMLR, 2020.

[40] Anian Ruoss, Mislav Balunovic, Marc Fischer, and Martin T. Vechev. Learning certified individually fair representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[41] Candice Schumann, Jeffrey S. Foster, Nicholas Mattei, and John P. Dickerson. We Need Fairness and Explainability in Algorithmic Hiring. In *Proc. 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, pages 1716–1720, 2020.

[42] M. Srinivas and L. M. Patnaik. Genetic algorithms: a survey. *Computer*, 27(6):17–26, 1994.

[43] Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang. Perfectly parallel fairness certification of neural networks. *Proc. ACM Program. Lang.*, 4(OOPSLA):185:1–185:30, 2020.

[44] Caterina Urban and Antoine Miné. A Review of Formal Methods applied to Machine Learning. *CoRR*, abs/2104.02466, 2021.

[45] Mikhail Yurochkin, Amanda Bower, and Yuekai Sun. Training individually fair ML models with sensitive subspace robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[46] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proc. 26th International Conference on World Wide Web (WWW 2017)*, pages 1171–1180, 2017.

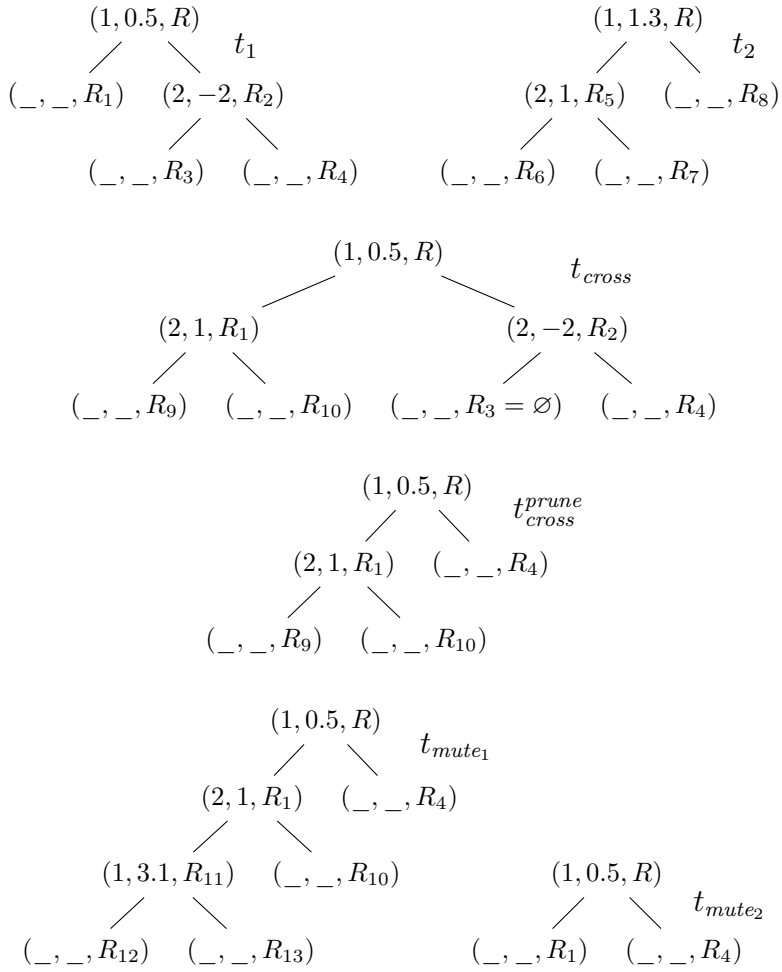


Figure 3: Example of crossover and mutation

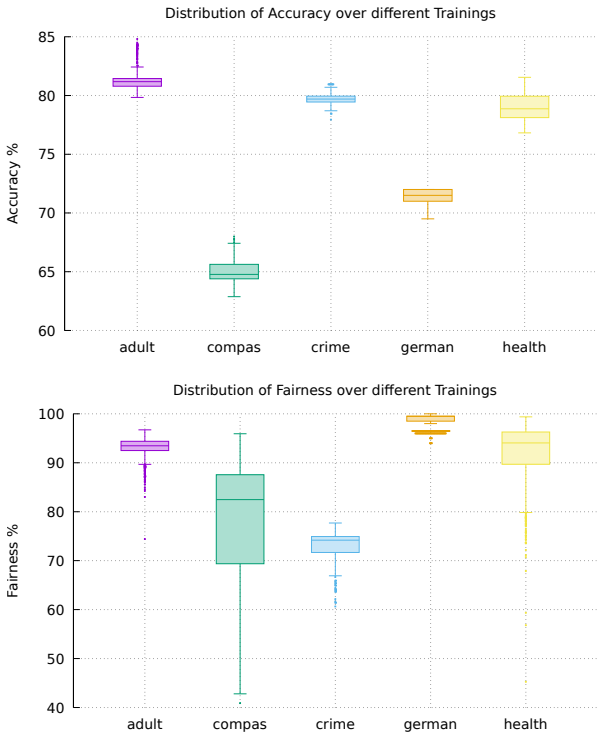


Figure 4: Distribution of Accuracy (left) and Fairness (right)