Fair Training of Decision Tree Classifiers

Francesco Ranzato[1], **Caterina Urban**[2], **Marco Zanella**[1]
[1] Dipartimento di Matematica, University of Padova, Italy
{ranzato, mzanella}@math.unipd.it
[2] INRIA and École Normale Supérieure | Université PSL, France
caterina.urban@inria.fr

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 classification model, which is a variant of standard robustness under input perturbations used in adversarial machine learning. Our verification and training methods leverage abstract interpretation, a well established technique for static program analysis which is able to automatically infer assertions about stability properties of decision trees. By relying on a tool for adversarial training of decision trees, our fairness-aware learning method has been implemented and experimentally evaluated on the reference datasets used to assess fairness properties. The experimental results show that our approach is able to train tree models exhibiting a high degree of individual fairness w.r.t. the natural state-of-the-art CART trees and random forests. Moreover, as a by-product, these fair decision trees turn out to be significantly compact, thus enhancing the interpretability of their fairness properties.

1 Introduction

The widespread adoption of data-driven automated decision-making software with far-reaching societal impact, e.g., for credit scoring (Khandani, Kim, and Lo 2010), recidivism prediction (Chouldechova 2017), or hiring tasks (Schumann et al. 2020), raises concerns on the fairness properties of these tools (Barocas and Selbst 2016). Several fairness verification and bias mitigation approaches for machine learning (ML) systems have been proposed in recent years, e.g. (Aghaei, Azizi, and Vayanos 2019; Grari et al. 2020; Roh et al. 2020; Ruoss et al. 2020; Urban et al. 2020; Yurochkin, Bower, and Sun 2020; Zafar et al. 2017) among the others. However, most works focus on neural networks (Roh et al. 2020; Ruoss et al. 2020; Urban et al. 2020; Yurochkin, Bower, and Sun 2020) or on group-based notions of fairness (Grari et al. 2020; Zafar et al. 2017), e.g., demographic parity (Dwork et al. 2012) or equalized odds (Hardt, Price, and Srebro 2016). 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* (Dwork et al. 2012), intuitively meaning that similar individuals in the population receive similar outcomes, and on decision tree ensembles (Breiman 2001; Friedman 2001), which are commonly used for tabular datasets since they are easily interpretable ML models with high accuracy rates.

Contributions. We propose an approach for verifying individual fairness of decision tree ensembles, as well as training tree models which maximize both accuracy and fairness. The approach is based on *abstract interpretation* (Cousot and Cousot 1977; Rival and Yi 2020), a well known static program analysis technique, and builds upon a framework for training robust decision tree ensembles called Meta-Silvae (Ranzato and Zanella 2020b), which in turn leverages a verification tool for robustness properties of decision trees (Ranzato and Zanella 2020a). Our approach is fully parametric on a given underlying abstract domain representing input space regions containing similar individuals. We instantiate it with a product of two abstract domains: (a) the well-known abstract domain of hyper-rectangles (or boxes) (Cousot and Cousot 1977), 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 decision rules of type $x_i \leq k$; and (b) a specific relational abstract domain which accounts for one-hot encoded categorical features.

Our Fairness-Aware Tree Training method, called FATT, is designed as an extension of Meta-Silvae (Ranzato and Zanella 2020b), a learning methodology for ensembles of decision trees based on a genetic algorithm which is able to train a decision tree for maximizing both its accuracy and its robustness to adversarial perturbations. We demonstrate the effectiveness of FATT in training accurate and fair models on the standard datasets used in the literature on fairness. Overall, the experimental results show that our fair-trained 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 more interpretable. Finally, we show how our models can be used as "hints" for setting the size and shape hyperparameters (i.e., maximum 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.

Related Work. The most related work to ours is by Aghaei et al. (Aghaei, Azizi, and Vayanos 2019), Raff et al. (Raff, Sylvester, and Mills 2018) and Ruoss et al. (Ruoss et al. 2020).

By relying on the mixed-integer optimization learning approach by Bertsimas and Dunn (Bertsimas and Dunn 2017), Aghaei et al. (Aghaei, Azizi, and Vayanos 2019) 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, also individuals that are not necessarily part of a given dataset).

Raff et al. (Raff, Sylvester, and Mills 2018) 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. (Aghaei, Azizi, and Vayanos 2019). 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 (i.e., the 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 (cf. Table 1).

Finally, Ruoss et al. (Ruoss et al. 2020) 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).

2 Background

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 is a function $C\colon X \to \wp_+(\mathcal{L})$, where $\wp_+(\mathcal{L})$ is the set of nonempty subsets of $\mathcal{L}$, which 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 $C\colon X \to \wp_+(\mathcal{L})$ which optimizes some objective function, such as the Gini index or the information gain for decision trees.

Categorical features can be converted into numerical ones through one-hot encoding, where a single feature with k possible distinct categories $\{c_1, ..., c_k\}$ is replaced by k new binary features with 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. Accuracy on a test set is a basic metric: given a ground truth test set $T \subseteq X \times \mathcal{L}$, the accuracy of C on T is $acc_T(C) \triangleq |\{(\boldsymbol{x}, y) \in T \mid C(\boldsymbol{x}) = \{y\}\}|/|T|$. However, according to a growing belief (Goodfellow, McDaniel, and Papernot 2018), accuracy is not enough in machine learning, since robustness to adversarial inputs of a ML classifier may significantly affect its safety and generalization properties (Carlini and Wagner 2017; Goodfellow, McDaniel, and Papernot 2018). Given an input perturbation modeled by a function $P\colon X \to \wp(X)$, a classifier $C : X \to \wp_+(\mathcal{L})$ is *stable* (Ranzato and Zanella 2020a) on the perturbation $P(\boldsymbol{x})$ of $\boldsymbol{x} \in X$ when C consistently assigns the same label(s) to every attack ranging in $P(\boldsymbol{x})$, i.e.,

$$\mathrm{stable}(C, \boldsymbol{x}, P) \overset{\triangle}{\Leftrightarrow} \forall \boldsymbol{x}' \in P(\boldsymbol{x})\colon C(\boldsymbol{x}') = C(\boldsymbol{x}).$$

When the sample $\boldsymbol{x} \in X$ has a ground truth label $y_{\boldsymbol{x}} \in \mathcal{L}$, robustness of C on $\boldsymbol{x}$ boils down to stability $\mathrm{stable}(C, \boldsymbol{x}, P)$ together with correct classification $C(\boldsymbol{x}) = \{y_{\boldsymbol{x}}\}$.

We consider standard classification decision trees commonly referred to as CARTs (Classification And Regression Trees) (Breiman et al. 1984). A decision tree $t\colon X \to \wp_+(\mathcal{L})$ is defined inductively. A base tree t is a single leaf λ storing a 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 $\arg\max_{y \in \mathcal{L}} \lambda(y)$. A composite tree t is $\gamma(split, t_l, t_r)$, where $split\colon X \to \{\mathbf{tt}, \mathbf{ff}\}$ is a Boolean split criterion for the internal parent node of its left and right subtrees t_l and t_r; thus, for all $\boldsymbol{x} \in X$, $t(\boldsymbol{x}) \triangleq$ **if** $split(\boldsymbol{x})$ **then** $t_l(\boldsymbol{x})$ **else** $t_r(\boldsymbol{x})$. Although split rules can be of any type, most decision trees employ univariate hard splits of type $split(\boldsymbol{x}) \triangleq \boldsymbol{x}_i \leq k$ 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) (Breiman 2001) 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 (GBDT) (Friedman 2001) represent a different training algorithm where an ensemble of trees is incrementally build 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.

3 Individual Fairness

Dwork et al. (Dwork et al. 2012) 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 $\delta(\boldsymbol{x}, \boldsymbol{y}) \in [0,1]$ map to distributions $D_{\boldsymbol{x}}$ and $D_{\boldsymbol{y}}$, respectively, such that the statistical 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 $\delta(\boldsymbol{x}, \boldsymbol{y})$. The distance metric $\delta \colon X \times X \to \mathbb{R}_{\geq 0}$ is problem specific and satisfies the basic axioms $\delta(\boldsymbol{x}, \boldsymbol{y}) = \delta(\boldsymbol{y}, \boldsymbol{x})$ and $\delta(\boldsymbol{x}, \boldsymbol{x}) = 0$.

By following Dwork et al's standard definition (Dwork et al. 2012), 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 $S \subseteq X \times X$. Thus, S can be derived from a distance δ as $(\boldsymbol{x}, \boldsymbol{y}) \in S \overset{\triangle}{\Leftrightarrow} \delta(\boldsymbol{x}, \boldsymbol{y}) \leq \epsilon$, where $\epsilon \in \mathbb{R}$ is a threshold of similarity. In order to estimate a fairness metric for a classifier C, we count how often C is fair on sets of similar individuals ranging into a test set $T \subseteq X \times \mathcal{L}$:

$$\mathit{fair}_T(C) \triangleq \frac{|\{(\boldsymbol{x}, y) \in T \mid \mathrm{fair}(C, \boldsymbol{x}, S)\}|}{|T|} \tag{1}$$

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

Definition 3.1 (Individual Fairness). A classifier $C \colon X \to \wp_+(\mathcal{L})$ is *fair* on an input sample $\boldsymbol{x} \in X$ with respect to a similarity relation $S \subseteq X \times X$, denoted by $\mathrm{fair}(C, \boldsymbol{x}, S)$, when $\forall \boldsymbol{x}' \in X \colon (\boldsymbol{x}, \boldsymbol{x}') \in S \Rightarrow C(\boldsymbol{x}') = C(\boldsymbol{x})$. □

Hence, fairness for a similarity relation S boils down to stability on the perturbation $P_S(\boldsymbol{x}) \triangleq \{\boldsymbol{x}' \in X \mid (\boldsymbol{x}, \boldsymbol{x}') \in S\}$, namely, for all $\boldsymbol{x} \in X$,

$$\mathrm{fair}(C, \boldsymbol{x}, S) \;\Leftrightarrow\; \mathrm{stable}(C, \boldsymbol{x}, P_S) \tag{2}$$

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 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 RFs, 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 monotonic with respect to the similarity relation, meaning that

$$\mathrm{fair}(C, \boldsymbol{x}, S) \wedge S' \subseteq S \;\Rightarrow\; \mathrm{fair}(C, \boldsymbol{x}, S') \tag{3}$$

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

4 Verifying Fairness

As individual fairness is equivalent to stability, individual fairness of decision trees can be verified by Silva (Ranzato and Zanella 2020a), an abstract interpretation-based algorithm for checking stability properties of decision tree ensembles.

4.1 Verification by Silva

Silva performs a static analysis of an ensemble of decision trees in a so-called abstract domain A that approximates properties of real vectors, meaning that each abstract value $a \in A$ represents a set of real vectors $\gamma(a) \in \wp(\mathbb{R}^d)$. Silva 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 $P(\boldsymbol{x}) \subseteq \gamma(a)$ 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 $\gamma(a)$. This is computed by collecting the constraints of split nodes for each root-leaf path, so that each leaf λ of t stores the minimum set of constraints C_λ which makes λ reachable from the root of t. It is then checked if this set of constraints C_λ can be satisfied by the input abstract value $a \in A$: this check is denoted by $a \models^? C_\lambda$ and its *soundness* requirement means that if some input sample $\boldsymbol{z} \in \gamma(a)$ may reach 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 a. For example, if $C_\lambda = \{x_1 \leq 2, \neg(x_1 \leq -1), x_2 \leq -1\}$ then an abstract value such as $\langle x_1 \leq 0, x_2 \leq 0 \rangle$ satisfies C_λ while a relational abstract value such as $x_1 + x_2 = 4$ does not. This over-approximation of the set of leaves of t reachable from a allows us to compute a set of labels, denoted by $t^A(a) \in \wp_+(\mathcal{L})$ which is an over-approximation of the set of labels assigned by t to all the input vectors ranging in $\gamma(a)$, i.e., $\cup_{\boldsymbol{z} \in \gamma(a)} t(\boldsymbol{z}) \subseteq t^A(a)$ holds. Thus, if $P(\boldsymbol{x}) \subseteq \gamma(a)$ and $t^A(a) = t(\boldsymbol{x})$ then t is stable on $P(\boldsymbol{x})$.

For standard classification trees with hard univariate splits of type $x_i \leq k$, we will use the well-known hyper-rectangle abstract domain HR whose abstract values for vectors $\boldsymbol{x} \in \mathbb{R}^d$ are of type

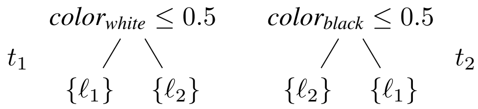
$$h = \langle \boldsymbol{x}_i \in [l_1, u_1], \ldots, \boldsymbol{x}_d \in [l_d, u_d] \rangle \in \mathrm{HR}_d$$

where lower and upper bounds $l, u \in \mathbb{R} \cup \{-\infty, +\infty\}$ with $l \leq u$ (more on this abstract domain can be found in (Rival and Yi 2020)). Thus, $\gamma(h) = \{\boldsymbol{x} \in \mathbb{R}^d \mid \forall i.\, l_i \leq \boldsymbol{x}_i \leq u_i\}$. The hyper-rectangle abstract domain guarantees that for each leaf constraint C_λ and $h \in \mathrm{HR}$, the check $h \models^? C_\lambda$ is (sound and) *complete*, meaning that $h \models C_\lambda$ holds iff there exists some input sample in $\gamma(h)$ reaching λ. This completeness property therefore entails that the set of labels $t^{\mathrm{HR}}(h)$ computed by this analysis coincides exactly with the set of classification labels computed by t for all the samples in $\gamma(h)$, so that for the ℓ_∞-based perturbation such that $P_\infty(\boldsymbol{x}) = \gamma(h)$ then it turns out that t is stable on $P_\infty(\boldsymbol{x})$ iff $t^{\mathrm{HR}}(h) = t(\boldsymbol{x})$ holds.

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 unfolding operation does not matter. Then, each leaf λ of this huge single tree t_F collects all the constraints of the leaves in the path from the root of t_F to λ. Since this stacked tree t_F suffers from a combinatorial explosion of the number of leaves, Silva deploys a number of optimisation strategies for its analysis. Basically, Silva exploits a best-first search algorithm to look for a pair of input samples in $\gamma(a)$ which are differently labeled, hence showing instability. If one such instability counterexample can be found then instability is proved and the analysis terminates, otherwise stability is proved. Also, Silva allows to set a safe timeout which, when met, stops the analysis and outputs the current sound over-approximation of labels.

4.2 Verification with One-Hot Enconding

As described above, the soundness of Silva guarantees that no true reachable leaf is missed by this static analysis. Moreover, when the input region $P(\boldsymbol{x})$ is defined by the ℓ_∞ norm and the static analysis is performed using the abstract domain of hyper-rectangles HR, Silva is also complete, meaning that no false positive (i.e., a false reachable leaf) can occur. However, that this is not true anymore when dealing with classification problems involving some categorical features.



The diagram above 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. $color_{white} = 1, color_{black} = 0$, will be labeled as ℓ_1 by both trees. However, by running the stability analysis on the hyper-rectangle $\langle color_{white} \in [0,1], color_{black} \in [0,1]\rangle$, Silva would mark the classifier as unstable because there exists a sample in $[0,1]^2$ whose output is $\{\ell_1,\ell_2\} \neq \{\ell_1\} = F(color_{white} = 1, color_{black} = 0)$. This is due to the point $(0,0) \in [0,1]^2$ which is a feasible input sample for the analysis, although it does not represent any actual individual in the input space. In fact, $t_1(0,0) = \{\ell_2\}$ and $t_2(0,0) = \{\ell_1\}$, so that by a majority voting $F(0,0) = \{\ell_1,\ell_2\}$, thus making F unstable (i.e., unfair) on $(1,0)$ (i.e., on white individuals).

To overcome this issue, we instantiate Silva to an abstract domain which is designed as a reduced product (more details on reduced products can be found in (Rival and Yi 2020)) with a relational abstract domain keeping track of the relationships among the multiple binary features introduced by one-hot encoding a categorical feature. More formally, this relational domain maintains the following two additional constraints on the k features $x_1^c, ..., x_k^c$ introduced by one-hot encoding a categorical variable x^c with k distinct values:

1. the possible values for each x_i^c are restricted to $\{0,1\}$;
2. the sum of all x_i^c must satisfy $\sum_{i=1}^{k} x_i^c = 1$.

Hence, these conditions guarantee that any abstract value for $x_1^c, ..., x_k^c$ represents precisely one possible 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 values $\langle color_{white} = 0, color_{black} = 1\rangle$ and $\langle color_{white} = 1, color_{black} = 0\rangle$ are allowed.

Summing up, the generic abstract value of the reduced hyper-rectangle domain computed by the analyzer Silva 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 values is:

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

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

5 FATT: Fairness-Aware Training of Trees

Several algorithms for training robust decision trees and ensembles have been put forward (Andriushchenko and Hein 2019; Calzavara, Lucchese, and Tolomei 2019; Calzavara et al. 2020; Chen et al. 2019; Kantchelian, Tygar, and Joseph 2016; Ranzato and Zanella 2020b). These algorithms encode the robustness of a tree classifier as a loss function which is minimized either by either exact methods such as MILP or by suboptimal heuristics such as genetic algorithms.

The robust learning algorithm of (Ranzato and Zanella 2020b), called Meta-Silvae, aims at maximizing a tunable weighted linear combination of accuracy and stability metrics. Meta-Silvae relies on a genetic algorithm for evolving a population of trees which are ranked by their accuracy and stability, where tree stability is computed by resorting to the verifier Silva (Ranzato and Zanella 2020a). At the end of this genetic evolution, Meta-Silvae returns the best tree(s). It turns out that Meta-Silvae typically outputs compact models which are easily interpretable and often achieve accurate and stable models already with a single decision tree rather than a forest. By exploiting the equivalence (2) between individual fairness and stability and the instantiation of the verifier Silva to the product abstract domain tailored for one-hot encoding, we use Meta-Silvae as a learning algorithm for decision trees, called FATT, that enhances their individual fairness.

While standard learning algorithms for tree ensembles require tuning some hyper-parameters, such as maximum depth of trees, minimum amount of information on leaves and maximum number of trees, Meta-Silvae is able to infer them automatically, so that the traditional 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 crossover and mutation functions (Holland 1984; Srinivas and Patnaik 1994). Moreover, we need to specify the objective function of FATT that, for learning fair decision trees, is given by a weighted sum of the accuracy and individual fairness scores over the training set. It is worth remarking that, given an objective function, the genetic algorithm of FATT converges to an optimal (or suboptimal) solution regardless of the chosen parameters, which just affect the rate of convergence and therefore should be chosen for tuning its speed.

Crossover and mutation functions are two main distinctive features of the genetic algorithm of Meta-Silvae. The crossover function of Meta-Silvae combines two parent trees t_1 and t_2 by randomly substituting a subtree of t_1 with a subtree of t_2. Also, Meta-Silvae supports two types of mutation strategies: grow-only, which only allows trees to grow, and grow-and-prune, which also allows pruning the mutated trees. Finally, let us point out that Meta-Silvae allows to set the basic parameters used by generic algorithms: population size, selection function, number of iterations. In our instantiation of Meta-Silvae to fair learning: the population size is kept fixed to 32, as the experimental evaluation showed that this provides an effective balance between achieved fairness and training time; the standard roulette wheel algorithm is employed as selection function; the number of iterations is typically dataset-specific.

6 Experimental Evaluation

We consider the main standard datasets used in the fairness literature and we preprocess them by following the steps of Ruoss et al. (Ruoss et al. 2020, 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; and (4) split into train and test set. These datasets concern binary classification tasks, although our fair learning naturally extends to multiclass classification with no specific effort. We will make all the code, datasets and preprocessing pipelines of FATT publicly available upon publication of this work.

Adult. The Adult income dataset (Dua and Graff 2017) is extracted from the 1994 US Census database. Every sample assigns a yearly income (below or above \$50K) 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 (Angwin et al. 2016). Each sample predicts the risk of recidivism for individuals based on personal attributes and criminal history.

Crime. The Communities and Crime dataset (Dua and Graff 2017) 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 (Dua and Graff 2017) 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.

		Training Set		**Test Set**	
dataset	#features	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%

Table 1: Overview of Datasets.

Table 1 displays size and distribution of positive samples for these datasets. As noticed by (Ruoss et al. 2020), some datasets exhibit a highly unbalanced label distribution. For example, for the adult dataset, the 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 (Ruoss et al. 2020) and we will evaluate and report the balanced accuracy of our FATT classifiers, i.e.,

$$0.5\,\left(\frac{\textit{truePositive}}{\textit{truePositive}+\textit{falseNegative}} + \frac{\textit{trueNegative}}{\textit{trueNegative}+\textit{falsePositive}}\right).$$

6.1 Similarity Relations

We consider four different types of similarity relations, as described by Ruoss et al. (Ruoss et al. 2020, Section 5.1). 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: $(\boldsymbol{x}, \boldsymbol{y}) \in S_{\textit{noise}}$ iff $|\boldsymbol{x}_i - \boldsymbol{y}_i| \leq \tau$ for all $i \in I'$, and $\boldsymbol{x}_i = \boldsymbol{y}_i$ for all $i \in I \smallsetminus I'$. For our experiments, we consider $\epsilon = 0.3$ 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 $I' \subseteq I$: $(\boldsymbol{x}, \boldsymbol{y}) \in S_{\textit{cat}}$ iff $\boldsymbol{x}_i = \boldsymbol{y}_i$ for all $i \in I \smallsetminus I'$. 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 $S_{\textit{noise}}$ and $S_{\textit{cat}}$, their union $S_{\textit{noise-cat}} \triangleq S_{\textit{noise}} \cup S_{\textit{cat}}$ 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 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}: $S_{\textit{cond}} \triangleq \{(\boldsymbol{x}, \boldsymbol{y}) \in S_{n_1} \mid \boldsymbol{x}_i \leq \tau,\ \boldsymbol{y}_i \leq \tau\} \cup \{(\boldsymbol{x}, \boldsymbol{y}) \in S_{n_2} \mid \boldsymbol{x}_i > \tau,\ \boldsymbol{y}_i > \tau\}$. 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.

Note that our approach is not limited to supporting these similarity relations. Further domain-specific similarities can be defined and handled by our approach by instantiating the underlying verifier Silva with an appropriately over-approximating abstract domain to retain soundness. Moreover, if the similarity relation can be precisely represented in the chosen abstract domain, we also retain completeness.

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 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 (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 and fairness in the objective function, type of mutation, selection function, number of iterations) by assessing convergence speed, maximum fitness value and variance among fitness in the population during the training phase. FATT trained single decision trees rather than forests, thus providing more compact and interpretable models. It turned out that accuracy and fairness of single FATT trees are already competitive, where individual fairness may exceed 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 results refer to their median value.

6.3 Results

	Acc. %		Bal.Acc. %		Individual Fairness $fair_T$ %					
					CAT		NOISE		NOISE-CAT	
Dataset	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
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
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**

Table 2: RF and FATT comparison.

Table 2 shows a comparison between RFs and FATTs. We show accuracy, balanced accuracy and individual fairness with respect to the NOISE, CAT, and NOISE-CAT similarity relations as computed on the test sets T. As expected, FATT trees are slightly less accurate than RFs — 3.6% on average, which also reflects to balanced accuracy — but outperform them in every fairness test. On average, the fairness increment ranges between $+35\%$ to $+45\%$ among different similarity relations. Table 3 shows the comparison for the conditional-attribute similarity, which applies to adult and german datasets only. Here, the average fairness increase of FATT models is $+8.5\%$.

	Individual Fairness $fair_T$ %	
Dataset	RF	FATT
adult	84.75	94.12
german	91.50	99.50

Table 3: Comparison for conditional-attribute.

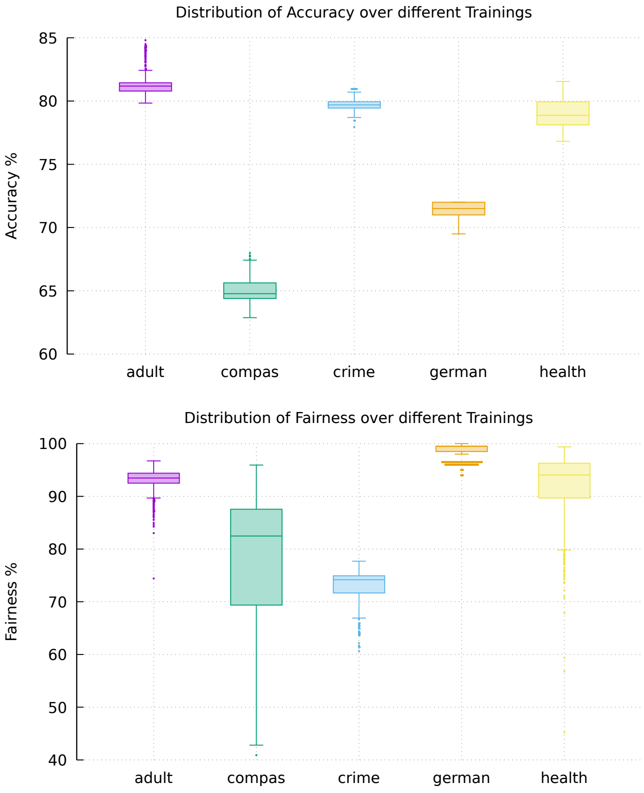


Figure 1: Accuracy (top) and Fairness (bottom).

Fig. 1 shows the distribution of accuracy and individual fairness for FATT trees over 1000 runs of the FATT learning algorithm. This plot is for fairness with respect to noise-cat similarity, as this is the most challenging relation to train for (this is a consequence of (3)). We can observe a stable behaviour for accuracy, with $\approx 50\%$ of the observations laying within one percentile from the median. The results for fairness are analogous, although for compas we report a higher variance of the distribution, where the lowest observed fairness percentage is $\approx 10\%$ higher than the corresponding one for RFs. We claim that this may depend by the high number of features in the dataset, which makes fair training a challenging task.

Table 4 compares the size of RF and FATT models, defined as total number of leaves. It turns out that FATT tree models are orders of magnitude smaller and, thus, more interpretable than RFs (while having comparable accuracy and

	Model size		**Avg. verification time per sample (ms)**					
			CAT		NOISE		NOISE-CAT	
Dataset	RF	FATT	RF	FATT	RF	FATT	RF	FATT
adult	1427	43	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
health	2558676	84	1.40	0.06	0.91	0.05	3.10	0.06

Table 4: Model sizes and verification times.

significantly enhanced fairness). Let us also remark that the average verification time per sample for our FATT models is always less than 0.01 milliseconds.

	FATT			**Natural CART**			**Hinted CART**		
Dataset	Acc. %	Fair. %	Size	Acc. %	Fair. %	Size	Acc. %	Fair. %	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 5: Decision trees comparison.

Finally, in Table 5 compares FATT models with natural CART trees in terms of accuracy, size, and fairness with respect to the noise-cat similarity. While CARTs are approximately 3% more accurate than FATT on average, they are roughly half less fair and more than ten times larger.

It is well known that decision trees often overfit (Bramer 2007) due to their high number of leaves, thus yielding unstable/unfair models. Post-training techniques such as tree pruning are often used to mitigate overfitting (Kearns and Mansour 1998), although they are deployed when a tree has been already fully trained and thus often pruning is poorly beneficial. As a byproduct of our approach, we trained a set of natural CART trees, denoted by Hint in Table 5, which exploits hyper-parameters as "hinted" by FATT training. In particular, in this "hinted" learning of CART trees, 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 5 show that these "hinted" decision trees 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 "hinted" CARTs is halfway between natural CARTs and FATTs, both in term of accuracy and 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 very 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 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 further our fairness analysis by considering alternative fairness definitions, such as group or statistical fairness.

References

Aghaei, S.; Azizi, M. J.; and Vayanos, P. 2019. Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, 1418–1426. AAAI Press. doi:10.1609/aaai.v33i01.33011418. URL https://doi.org/10.1609/aaai.v33i01.33011418.

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

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine Bias. *ProPublica, May* 23: 2016.

Barocas, S.; and Selbst, A. D. 2016. Big Data's Disparate Impact. *California Law Review* 104: 671.

Bertsimas, D.; and Dunn, J. 2017. Optimal classification trees. *Mach. Learn.* 106(7): 1039–1082. URL http://dblp.uni-trier.de/db/journals/ml/ml106.html#BertsimasD17.

Bramer, M. 2007. Avoiding overfitting of decision trees. *Principles of data mining* 119–134.

Breiman, L. 2001. Random Forests. *Machine Learning* 45(1): 5–32. doi:10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984. *Classification and Regression Trees*. Wadsworth. ISBN 0-534-98053-8.

Calzavara, S.; Lucchese, C.; and Tolomei, G. 2019. Adversarial Training of Gradient-Boosted Decision Trees. In *Proc. 28th ACM International Conference on Information and Knowledge Management (CIKM 2019)*, 2429–2432. ISBN 978-1-4503-6976-3. doi:10.1145/3357384.3358149. URL http://doi.acm.org/10.1145/3357384.3358149.

Calzavara, S.; Lucchese, C.; Tolomei, G.; Abebe, S. A.; and Orlando, S. 2020. TREANT: training evasion-aware decision trees. *Data Mining and Knowledge Discovery* doi:10.1007/s10618-020-00694-9. URL https://doi.org/10.1007/s10618-020-00694-9.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. of 38th IEEE Symposium on Security and Privacy (S & P 2017)*, 39–57. doi:10.1109/SP.2017.49. URL https://doi.org/10.1109/SP.2017.49.

Chen, H.; Zhang, H.; Boning, D. S.; and Hsieh, C. 2019. Robust Decision Trees Against Adversarial Examples. In *Proc. 36th Int. Conf. on Machine Learning, (ICML 2019)*, 1122–1131. URL http://proceedings.mlr.press/v97/chen19m.html.

Chouldechova, A. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5(2): 153–163. doi:10.1089/big.2016.0047. URL https://doi.org/10.1089/big.2016.0047.

Cousot, P.; and Cousot, R. 1977. 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)*, 238–252. doi:10.1145/512950.512973. URL http://doi.acm.org/10.1145/512950.512973.

Dua, D.; and Graff, C. 2017. UCI machine learning repository.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness Through Awareness. In *Proc. 3rd Innovations in Theoretical Computer Science Conference*, 214–226.

Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of statistics* 1189–1232.

Goodfellow, I.; McDaniel, P.; and Papernot, N. 2018. Making Machine Learning Robust Against Adversarial Inputs. *Commun. ACM* 61(7): 56–66. ISSN 0001-0782. doi:10.1145/3134599. URL http://doi.acm.org/10.1145/3134599.

Grari, V.; Ruf, B.; Lamprier, S.; and Detyniecki, M. 2020. Achieving Fairness with Decision Trees: An Adversarial Approach. *Data Sci. Eng.* 5(2): 99–110. doi:10.1007/s41019-020-00124-2. URL https://doi.org/10.1007/s41019-020-00124-2.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of Opportunity in Supervised Learning. In *Proc. 30th Annual Conference on Neural Information Processing Systems (NeurIPS 2016)*, 3315–3323. URL http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning.

Holland, J. H. 1984. Genetic algorithms and adaptation. In *Adaptive Control of Ill-Defined Systems*, 317–333. Springer.

Kantchelian, A.; Tygar, J. D.; and Joseph, A. D. 2016. Evasion and Hardening of Tree Ensemble Classifiers. In *Proc. 33rd International Conference on Machine Learning (ICML 2016)*, 2387–2396. URL http://dl.acm.org/citation.cfm?id=3045390.3045642.

Kearns, M. J.; and Mansour, Y. 1998. A Fast, Bottom-Up Decision Tree Pruning Algorithm with Near-Optimal Generalization. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, 269–277.

Khandani, A. E.; Kim, A. J.; and Lo, A. W. 2010. Consumer Credit-Risk Models via Machine-Learning Algorithms. *Journal of Banking & Finance* 34(11): 2767–2787. doi:https://doi.org/10.1016/j.jbankfin.2010.06.001.

Raff, E.; Sylvester, J.; and Mills, S. 2018. Fair Forests: Regularized Tree Induction to Minimize Model Bias. In *Proc. 1st AAAI/ACM Conference on AI, Ethics, and Society (AIES 2018)*, 243–250. doi:10.1145/3278721.3278742. URL https://doi.org/10.1145/3278721.3278742.

Ranzato, F.; and Zanella, M. 2020a. Abstract Interpretation of Decision Tree Ensemble Classifiers. In *Proc. 34th AAAI Conference on Artificial Intelligence (AAAI 2020),* Github: https://github.com/abstract-machine-learning/silva, 5478–5486. URL https://aaai.org/ojs/index.php/AAAI/article/view/5998.

Ranzato, F.; and Zanella, M. 2020b. Genetic Adversarial Training of Decision Trees. *arXiv:2012.11352,* Github: https://github.com/abstract-machine-learning/meta-silvae .

Rival, X.; and Yi, K. 2020. *Introduction to Static Analysis: An Abstract Interpretation Perspective*. The MIT Press.

Roh, Y.; Lee, K.; Whang, S.; and Suh, C. 2020. FR-Train: A Mutual Information-Based Approach to Fair and Robust Training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 8147–8157. PMLR. URL http://proceedings.mlr.press/v119/roh20a.html.

Ruoss, A.; Balunovic, M.; Fischer, M.; and Vechev, M. 2020. Learning Certified Individually Fair Representations. In *Proc. 34th Annual Conference on Advances in Neural Information Processing Systems (NeurIPS 2020).*

Schumann, C.; Foster, J. S.; Mattei, N.; and Dickerson, J. P. 2020. We Need Fairness and Explainability in Algorithmic Hiring. In *Proc. 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, 1716–1720. URL https://dl.acm.org/doi/abs/10.5555/3398761.3398960.

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

Urban, C.; Christakis, M.; Wüstholz, V.; and Zhang, F. 2020. Perfectly Parallel Fairness Certification of Neural Networks. *Proceedings of the ACM on Programming Languages* 4(OOPSLA): 185:1–185:30.

Yurochkin, M.; Bower, A.; and Sun, Y. 2020. 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. URL https://openreview.net/forum?id=B1gdkxHFDH.

Zafar, M. B.; Valera, I.; Gomez-Rodriguez, M.; and Gummadi, K. P. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *Proc. 26th International Conference on World Wide Web (WWW 2017)*, 1171–1180. doi:10.1145/3038912.3052660. URL https://doi.org/10.1145/3038912.3052660.