Justicia: A Stochastic SAT Approach to Formally Verify Fairness *

Bishwamittra Ghosh[1], **Debabrota Basu**[2,3], **Kuldeep S. Meel**[1]
[1] National University of Singapore, Singapore
[2] Chalmers University of Technology, Göteborg, Sweden
[3] Scool, Inria Lille- Nord Europe, France

Abstract

As a technology ML is oblivious to societal good or bad, and thus, the field of fair machine learning has stepped up to propose multiple mathematical definitions, algorithms, and systems to ensure different notions of fairness in ML applications. Given the multitude of propositions, it has become imperative to formally verify the fairness metrics satisfied by different algorithms on different datasets. In this paper, we propose a *stochastic satisfiability* (SSAT) framework, Justicia, that formally verifies different fairness measures of supervised learning algorithms with respect to the underlying data distribution. We instantiate Justicia on multiple classification and bias mitigation algorithms, and datasets to verify different fairness metrics, such as disparate impact, statistical parity, and equalized odds. Justicia is scalable, accurate, and operates on non-Boolean and compound sensitive attributes unlike existing distribution-based verifiers, such as FairSquare and VeriFair. Being distribution-based by design, Justicia is more robust than the verifiers, such as AIF360, that operate on specific test samples. We also theoretically bound the finite-sample error of the verified fairness measure.

Introduction

Machine learning (ML) is becoming the omnipresent technology of our time. ML algorithms are being used for high-stake decisions like college admissions, crime recidivism, insurance, and loan decisions. Thus, human lives are now pervasively influenced by data, ML, and their inherent bias.

Example 0.1. Let us consider an example (Figure 1) of deciding eligibility for health insurance depending on the fitness and income of the individuals of different age groups (20-40 and 40-60). Typically, incomes of individuals increase as their ages increase while their fitness deteriorates. We assume relation of income and fitness depends on the age as per the Normal distributions in Figure 1. Now, if we train a decision tree (Narodytska et al. 2018) on these fitness and income indicators to decide the eligibility of an individual to get a health insurance, we observe that the 'optimal' decision tree (ref. Figure 1) selects a person above and below 40 years with probabilities 0.18 and 0.72 respectively. This simple example demonstrates that even if an ML algorithm does not explicitly learn to differentiate on the basis of a sensitive attribute, it discriminates different age groups due to the utilitarian sense of accuracy that it tries to optimize.

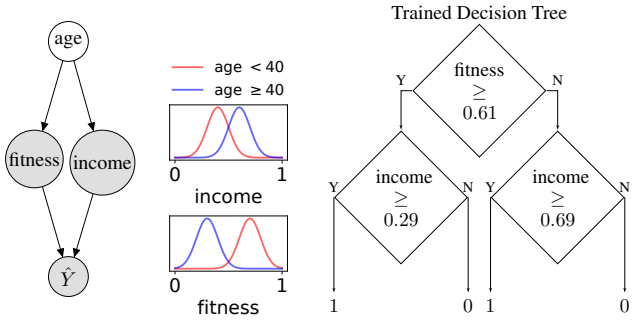

Figure 1: A trained decision tree to learn eligibility for health insurance using age-dependent fitness and income indicators.

Fair ML. Statistical discriminations caused by ML algorithms have motivated researchers to develop several frameworks to ensure fairness and several algorithms to mitigate bias. Existing fairness metrics mostly belong to three categories: *independence*, *separation*, and *sufficiency* (Mehrabi et al. 2019). Independence metrics, such as demographic parity, statistical parity, and group parity, try and ensure the outcomes of an algorithm to be independent of the groups that the individuals belong to (Feldman et al. 2015; Dwork et al. 2012). Separation metrics, such as equalized odds, define an algorithm to be fair if the probability of getting the same outcomes for different groups are same (Hardt, Price, and Srebro 2016). Sufficiency metrics, such as counterfactual fairness, constrain the probability of outcomes to be independent of individual's sensitive data given their identical non-sensitive data (Kusner et al. 2017).

In Figure 1, independence is satisfied if the probability of getting insurance is same for both the age groups. Separation is satisfied if the number of 'actually' (ground-truth) ineligible and eligible people getting the insurance are same. Sufficiency is satisfied if the eligibility is independent of their age given their attributes are the same. Thus, we see that the metrics of fairness can be contradictory and complimentary depending on the application and the data (Corbett-Davies

*Source code: https://github.com/meelgroup/justicia

and Goel 2018). Different algorithms have also been devised to ensure one or multiple of the fairness definitions. These algorithms try to rectify and mitigate the bias in the data and thus in the prediction-model in three ways: *pre-processing* the data (Kamiran and Calders 2012; Zemel et al. 2013; Calmon et al. 2017), *in-processing* the algorithm (Zhang, Lemoine, and Mitchell 2018), and *post-processing* the outcomes (Kamiran, Karim, and Zhang 2012; Hardt, Price, and Srebro 2016).

Fairness Verifiers. Due to the abundance of fairness metrics and difference in algorithms to achieve them, it has become necessary to verify different fairness metrics over datasets and algorithms.

In order to verify fairness as a model property on a dataset, verifiers like *FairSquare* (Albarghouthi et al. 2017) and *VeriFair* (Bastani, Zhang, and Solar-Lezama 2019) have been proposed. These verifiers are referred to as *distributional verifiers* owing to the fact that their inputs are a probability distribution of the attributes in the dataset and a model of a suitable form, and their objective is to verify fairness w.r.t. the distribution and the model. Though FairSquare and VeriFair are robust and have asymptotic convergence guarantees, we observe that they scale up poorly with the size of inputs and also do not generalize to non-Boolean and compound sensitive attributes. In contrast to the distributional verifiers, another line of work, referred to as sample-based verifiers, has focused on the design of testing methodologies on a given fixed data sample (Galhotra, Brun, and Meliou 2017; Bellamy et al. 2018). Since sample-based verifiers are dataset-specific, they generally do not provide robustness over the distribution.

Thus, a *unified formal framework* to verify *different fairness metrics* of an ML algorithm, which is *scalable*, capable of *handling compound protected groups*, *robust* with respect to the test data, and *operational on real-life* datasets and fairness-enhancing algorithms, is missing in the literature.

Our Contribution. From this vantage point, *we propose to model verifying different fairness metrics as a Stochastic Boolean Satisfiability (SSAT) problem* (Littman, Majercik, and Pitassi 2001). SSAT was originally introduced by (Papadimitriou 1985) to model *games against nature*. In this work, we primarily focus on reductions to the exist-random quantified fragment of SSAT, which is also known as E-MAJSAT (Littman, Majercik, and Pitassi 2001). SSAT is a conceptual framework that has been employed to capture several fundamental problems in AI such as computation of maximum a posteriori (MAP) hypothesis (Fremont, Rabe, and Seshia 2017), propositional probabilistic planning (Majercik 2007), and circuit verification (Lee and Jiang 2018). Furthermore, our choice of SSAT as a target formulation is motivated by the recent algorithmic progress that has yielded efficient SSAT tools (Lee, Wang, and Jiang 2017, 2018).

Our contributions are summarised below:

- We propose a unified SSAT-based approach, Justicia, to verify independence and separation metrics of fairness for different datasets and classification algorithms.
- Unlike previously proposed formal distributional verifiers, namely FairSquare and VeriFair, Justicia verifies fairness for compound and non-Boolean sensitive attributes.
- Our experiments validate that our method is more accurate and scalable than the distributional verifiers, such as FairSquare and VeriFair, and more robust than the sample-based empirical verifiers, such as AIF360.
- We prove a finite-sample error bound on our estimated fairness metrics which is stronger than the existing asymptotic guarantees.

It is worth remarking that significant advances in AI bear testimony to the right choice of formulation, for example, formulation of planning as SAT (Kautz, Selman et al. 1992). In this context, we view that formulation of fairness as SSAT has potential to spur future work from both the modeling and encoding perspective as well as core algorithmic improvements in the underlying SSAT solvers.

Background: Fairness and SSAT

In this section, we define different fairness metrics for a supervised learning problem. Following that, we discuss Stochastic Boolean Satisfiability (SSAT) problem.

Fairness Metrics for Machine Learning

Let us represent a dataset D as a collection of triples (X, A, Y) sampled from an underlying data generating distribution $\mathcal{D}$. $X \triangleq \{X_1, \ldots, X_m\} \in \mathbb{R}^m$ is the set of non-protected (or non-sensitive) attributes. $A \triangleq \{A_1, \ldots, A_n\}$ is the set of categorical protected attributes. Y is the binary label (or class) of (X, A). A compound protected attribute $\mathbf{a} = \{a_1, \ldots, a_n\}$ is a valuation to all A_i's and represents a *compound protected* group. For example, $A = \{\text{race}, \text{sex}\}$, where race $\in \{\text{Asian}, \text{Colour}, \text{White}\}$ and sex $\in \{\text{female}, \text{male}\}$. Thus, $\mathbf{a} = \{\text{Colour}, \text{female}\}$ is a compound protected group. We define $\mathcal{M} \triangleq \Pr(\hat{Y}|X, A)$ to be a binary classifier trained from samples in the distribution $\mathcal{D}$. Here, $\hat{Y}$ is the predicted label (or class) of the corresponding data.

As we illustrated in Example 0.1, a classifier $\mathcal{M}$ that solely optimizes accuracy, i.e., the average number of times $\hat{Y} = Y$, may discriminate certain compound protected groups over others (Chouldechova and Roth 2020). Now, we describe two family of fairness metrics that compute bias induced by a classifier and are later verified by Justicia.

Independence Metrics of Fairness. The *independence (or calibration) metrics* of fairness state that the output of the classifier should be independent of the compound protected group. A notion of independence is referred to *group fairness* that specifies an *equal positive predictive value (PPV) across all compound protected groups* for an algorithm $\mathcal{M}$, i.e., $\Pr[\hat{Y} = 1|A = \mathbf{a}, \mathcal{M}] = \Pr[\hat{Y} = 1|A = \mathbf{b}, \mathcal{M}], \forall \mathbf{a}, \mathbf{b} \in A$. Since satisfying group fairness exactly is hard, relaxations of group fairness, such as *disparate impact* and *statistical parity* (Dwork et al. 2012; Feldman et al. 2015), are proposed.

Disparate impact (DI) (Feldman et al. 2015) measures the ratio of PPVs between the most favored group and least favored group, and prescribe it to be close to 1. Formally, a classifier satisfies $(1-\epsilon)$-disparate impact if, for $\epsilon \in [0, 1]$,

$$\min_{\mathbf{a} \in A} \Pr[\hat{Y} = 1|\mathbf{a}, \mathcal{M}] \geq (1 - \epsilon) \max_{\mathbf{b} \in A} \Pr[\hat{Y} = 1|\mathbf{b}, \mathcal{M}].$$

Another popular relaxation of group fairness, *statistical parity* (SP) measures the difference of PPV among the compound groups, and prescribe this to be near zero. Formally, an algorithm satisfies ϵ-statistical parity if, for $\epsilon \in [0, 1]$,

$$\max_{\mathbf{a},\mathbf{b}\in A} |\Pr[\hat{Y} = 1|\mathbf{a}, \mathcal{M}] - \Pr[\hat{Y} = 1|\mathbf{b}, \mathcal{M}]| \leq \epsilon.$$

For both disparate impact and statistical parity, lower value of ϵ indicates higher group fairness of the classifier $\mathcal{M}$.

Separation Metrics of Fairness. In the *separation (or classification parity)* notion of fairness, the predicted label $\hat{Y}$ of a classifier $\mathcal{M}$ is independent of the sensitive attributes A given the actual class labels Y. In case of binary classifiers, a popular separation metric is *equalized odds* (EO) (Hardt, Price, and Srebro 2016) that computes the difference of false positive rates (FPR) and the difference of true positive rates (TPR) among all compound protected groups. Lower value of equalized odds indicates better fairness. A classifier $\mathcal{M}$ satisfies ϵ-equalized odds if, for all compound protected groups $\mathbf{a}, \mathbf{b} \in A$,

$$|\Pr[\hat{Y} = 1|A = \mathbf{a}, Y = 0] - \Pr[\hat{Y} = 1|A = \mathbf{b}, Y = 0]| \leq \epsilon,$$
$$|\Pr[\hat{Y} = 1|A = \mathbf{a}, Y = 1] - \Pr[\hat{Y} = 1|A = \mathbf{b}, Y = 1]| \leq \epsilon.$$

In this paper, we formulate verifying the aforementioned independence and separation metrics of fairness as stochastic Boolean satisfiability (SSAT) problem, which we define next.

Stochastic Boolean Satisfiability (SSAT)

Let $\mathbf{B} = \{B_1, \ldots, B_m\}$ be a set of Boolean variables. A *literal* is a variable B_i or its complement $\neg B_i$. A propositional formula ϕ defined over $\mathbf{B}$ is in *Conjunctive Normal Form (CNF)* if ϕ is a conjunction of clauses and each clause is a disjunction of literals. Let σ be an assignment to the variables $B_i \in \mathbf{B}$ such that $\sigma(B_i) \in \{1, 0\}$ where 1 is logical TRUE and 0 is logical FALSE. The propositional *satisfiability* problem (SAT) (Biere, Heule, and van Maaren 2009) finds an assignment σ to all $B_i \in \mathbf{B}$ such that the formula ϕ is evaluated to be 1. In contrast to the SAT problem, the *Stochastic Boolean Satisfiability* (SSAT) problem (Littman, Majercik, and Pitassi 2001) is concerned with the probability of the satisfaction of the formula ϕ. An SSAT formula is of the form

$$\Phi = Q_1 B_1, \ldots, Q_m B_m,\ \phi, \tag{1}$$

where $Q_i \in \{\exists, \forall, \mathfrak{R}^{p_i}\}$ is either of the existential ($\exists$), universal ($\forall$), or randomized ($\mathfrak{R}^{p_i}$) quantifiers over the Boolean variable B_i and ϕ is a quantifier-free CNF formula. In the SSAT formula Φ, the quantifier part $Q_1 B_1, \ldots, Q_m B_m$ is known as the *prefix* of the formula ϕ. In case of randomized quantification $\mathfrak{R}^{p_i}$, $p_i \in [0, 1]$ is the probability of B_i being assigned to 1. Given an SSAT formula Φ, let B be the outermost variable in the prefix. The satisfying probability of Φ can be computed by the following *rules*:

1. $\Pr[\text{TRUE}] = 1$, $\Pr[\text{FALSE}] = 0$,
2. $\Pr[\Phi] = \max_B\{\Pr[\Phi|_B], \Pr[\Phi|_{\neg B}]\}$ if B is existentially quantified ($\exists$),
3. $\Pr[\Phi] = \min_B\{\Pr[\Phi|_B], \Pr[\Phi|_{\neg B}]\}$ if B is universally quantified ($\forall$),
4. $\Pr[\Phi] = p\Pr[\Phi|_B] + (1-p)\Pr[\Phi|_{\neg B}]$ if B is randomized quantified ($\mathfrak{R}^p$) with probability p of being TRUE,

where $\Phi|_B$ and $\Phi|_{\neg B}$ denote the SSAT formulas derived by eliminating the outermost quantifier of B by substituting the value of B in the formula ϕ with 1 and 0 respectively. In this paper, we focus on two specific types of SSAT formulas: *random-exist* (RE) SSAT and *exist-random* (ER) SSAT. In the ER-SSAT (resp. RE-SSAT) formula, all existentially (resp. randomized) quantified variables are followed by randomized (resp. existentially) quantified variables in the prefix.

Lemma 1. (Littman, Majercik, and Pitassi 2001) Solving the ER-SSAT and RE-SSAT problems are $\mathrm{NP}^{\mathrm{PP}}$ hard.

The problem of SSAT and its variants have been pursued by theoreticians and practitioners alike for over three decades (Majercik and Boots 2005; Fremont, Rabe, and Seshia 2017; Huang et al. 2006). We refer the reader to (Lee, Wang, and Jiang 2017, 2018) for detailed survey. It is worth remarking that the past decade has witnessed a significant performance improvements thanks to close integration of techniques from SAT solving with advances in weighted model counting (Sang et al. 2004; Chakraborty, Meel, and Vardi 2013; Chakraborty et al. 2014).

Justicia: An SSAT Framework to Verify Fairness Metrics

In this section, we present the primary contribution of this paper, Justicia, which is an SSAT-based framework for verifying independence and separation metrics of fairness.

Given a binary classifier $\mathcal{M}$ and a probability distribution over dataset $(X, A, Y) \sim \mathcal{D}$, our goal is to verify whether $\mathcal{M}$ achieves independence and separation metrics with respect to the distribution $\mathcal{D}$. We focus on a classifier that can be translated to a CNF formula of Boolean variables $\mathbf{B}$. The probability p_i of $B_i \in \mathbf{B}$ being assigned to 1 is induced by the data generating distribution $\mathcal{D}$. In order to verify fairness metrics in compound protected groups, we discuss an enumeration-based approach and an equivalent learning-based approach. We then provide a theoretical analysis for a high-probability error bound on the fairness metric and conclude with extension of Justicia in practical settings.

Evaluating Fairness with RE-SSAT Encoding

In order to verify independence and separation metrics, the core component of Justicia is to compute the positive predictive value $\Pr[\hat{Y} = 1|A = \mathbf{a}]$ for a compound protected group $\mathbf{a}$. For simplicity, we initially make some assumptions and discuss their practical relaxations later in this section. We first assume the classifier $\mathcal{M}$ is representable as a CNF formula, namely $\phi_{\hat{Y}}$, such that $\hat{Y} = 1$ when $\phi_{\hat{Y}}$ is satisfied and $\hat{Y} = 0$ otherwise. Since a Boolean CNF classifier is defined over Boolean variables, we assume all attributes in X and A to be Boolean. Finally, we assume independence of non-protected attributes on protected attributes and p_i is the probability of the attribute X_i being assigned to 1 for any $X_i \in X$.

Now, we define an RE-SSAT formula $\Phi_{\mathbf{a}}$ to compute the probability $\Pr[\hat{Y} = 1|A = \mathbf{a}]$. In the prefix of $\Phi_{\mathbf{a}}$, all non-protected Boolean attributes in X are assigned randomized

quantification and they are followed by the protected Boolean attributes in A with existential quantification. The CNF formula ϕ in $\Phi_{\mathbf{a}}$ is constructed such that ϕ encodes the event inside the target probability $\Pr[\hat{Y} = 1|A = \mathbf{a}]$. In order to encode the conditional $A = \mathbf{a}$, we take the conjunction of the Boolean variables in A that symbolically specifies the compound protected group $\mathbf{a}$. For example, we represent two protected attributes: race $\in \{\text{White, Colour}\}$ and sex $\in \{\text{male, female}\}$ by the Boolean variables R and S respectively. Hence, the compound groups $\{\text{White, male}\}$ and $\{\text{Colour, female}\}$ are represented by $R \wedge S$ and $\neg R \wedge \neg S$, respectively. Thus, the RE-SSAT formula for computing the probability $\Pr[\hat{Y} = 1|A = \mathbf{a}]$ is

$$\Phi_{\mathbf{a}} := \underbrace{\mathfrak{R}^{p_1} X_1, \ldots, \mathfrak{R}^{p_m} X_m}_{\text{non-protected attributes}}, \underbrace{\exists A_1, \ldots, \exists A_n}_{\text{protected attributes}}, \phi_{\hat{Y}} \wedge (A = \mathbf{a}).$$

In $\Phi_{\mathbf{a}}$, the existentially quantified variables $A_1, \ldots, A_n$ are assigned values according to the constraint $A = \mathbf{a}$. [1] Therefore, by solving the SSAT formula $\Phi_{\mathbf{a}}$, the SSAT solver finds the probability $\Pr[\Phi_{\mathbf{a}}]$ for the protected group $A = \mathbf{a}$ given the random values of $X_1, \ldots, X_m$, which is the PPV of the protected group $\mathbf{a}$ for the distribution $\mathcal{D}$ and algorithm $\mathcal{M}$.

For simplicity, we have described computing the PPV of each compound protected group without considering the correlation between the protected and non-protected attributes. In reality, correlation exists between the protected and non-protected attributes. Thus, the non-protected attributes may have different conditional distributions for different protected groups. We incorporate these conditional distributions in RE-SSAT encoding by evaluating the conditional probability $p_i = \Pr[X_i = \text{TRUE}|A = \mathbf{a}]$ instead of the independent probability $\Pr[X_i = \text{TRUE}]$ for any $X_i \in X$. We illustrate this method in Example 0.2.

Example 0.2 (RE-SSAT encoding)**.** Here, we illustrate the RE-SSAT formula for calculating the PPV for the protected group 'age ≥ 40' in the decision tree of Figure 1. We assign three Boolean variables F, I, J for the three nodes in the tree such that the literal F, I, J denote 'fitness ≥ 0.61', 'income ≥ 0.29', and 'income ≥ 0.69', respectively. We consider another Boolean variable A where the literal A represents the protected group 'age ≥ 40'. Thus, the CNF formula for the decision tree is $(\neg F \vee I) \wedge (F \vee J)$. From the distribution in Figure 1, we get $\Pr[F] = 0.41, \Pr[I] = 0.93$, and $\Pr[J] = 0.09$. Given this information, we calculate the PPV for the protected group 'age ≥ 40' by solving the RE-SSAT formula:

$$\Phi_A := \mathfrak{R}^{0.41} F, \mathfrak{R}^{0.93} I, \mathfrak{R}^{0.09} J, \exists A,\ (\neg F \vee I) \wedge (F \vee J) \wedge A.$$

From the solution to this SSAT formula, we get $\Pr[\Phi_A] = 0.43$. Similarly, to calculate the PPV for the group 'age < 40', we replace the unit (single-literal) clause A with $\neg A$ in the CNF in Φ_A and construct another SSAT formula $\Phi_{\neg A}$ where $\Pr[\Phi_{\neg A}] = 0.43$. Therefore, if $\Pr[F], \Pr[I], \Pr[J]$ are computed independently of A and $\neg A$, both age groups demonstrate equal PPV as the protected attribute is not explicitly present in the classifier. However, there is an implicit bias in the data distribution for different protected groups and the classifier unintentionally learns it. To capture this implicit bias, we calculate the conditional probabilities $\Pr[F|A] = 0.01, \Pr[I|A] = 0.99$, and $\Pr[J|A] = 0.18$ from the distribution. Using the conditional probabilities in Φ_A, we find that $\Pr[\Phi_A] = 0.18$ for 'age ≥ 40'. For 'age < 40', we similarly obtain $\Pr[F|\neg A] = 0.82, \Pr[I|\neg A] = 0.88$, and $\Pr[J|\neg A] = 0.01$, and thus $\Pr[\Phi_{\neg A}] = 0.72$. Therefore, presented RE-SSAT encoding detects the discrimination of the classifier among different protected groups. An astute reader would observe that I and J are not independent. Following (Chavira and Darwiche 2008), we can simply capture relationship between the variables using constraints and if needed, auxiliary variables. In this case, it suffices to add the the constraint $J \rightarrow I$.

[1] An RE-SSAT formula becomes an R-SSAT formula when the assignment to the existential variables are fixed.

Algorithm 1 Justicia: SSAT-based Fairness Verifier

```
1: function Justicia_enum(X, A, Ŷ)
2:     φ_Ŷ := CNF(Ŷ = 1)
3:     for all a ∈ A do
4:         p_i ← CalculateProb(X_i|a), ∀X_i ∈ X
5:         φ := φ_Ŷ ∧ (A = a)
6:         Φ_a := ℜ^{p_1} X_1, ..., ℜ^{p_m} X_m, ∃A_1, ..., ∃A_n, φ
7:         Pr[Φ_a] ← SSAT(Φ_a)            ▷ returns a probability
8:     return max_a Pr[Φ_a], min_a Pr[Φ_a]
9: function Justicia_learn(X, A, Ŷ)
10:    φ_Ŷ := CNF(Ŷ = 1)
11:    p_i ← CalculateProb(X_i), ∀X_i ∈ X
12:    Φ_ER := ∃A_1, ..., ∃A_n, ℜ^{p_1} X_1, ..., ℜ^{p_m} X_m, φ_Ŷ
13:    Φ'_ER := ∃A_1, ..., ∃A_n, ℜ^{p_1} X_1, ..., ℜ^{p_m} X_m, ¬φ_Ŷ
14:    return SSAT(Φ_ER), 1 − SSAT(Φ'_ER)
```

Measuring Fairness Metrics. As we compute the probability $\Pr[\hat{Y} = 1|A = \mathbf{a}]$ by solving the SSAT formula $\Phi_{\mathbf{a}}$, we use $\Pr[\Phi_{\mathbf{a}}]$ to measure different fairness metrics. For that, we compute $\Pr[\Phi_{\mathbf{a}}]$ for all compound groups $\mathbf{a} \in A$ that requires solving exponential (with n) number of SSAT instances. We elaborate this enumeration approach, namely Justicia_enum, in Algorithm 1 (Line 1–8).

We calculate the ratio of the minimum and the maximum probabilities according to the definition of disparate impact. We compute statistical parity by taking the difference between the maximum and the minimum probabilities of all $\Pr[\Phi_{\mathbf{a}}]$. Moreover, to measure equalized odds, we compute two SSAT instances for each compound group with modified values of p_i. Specifically, to compute TPR, we use the conditional probability $p_i = \Pr[X_i|Y = 1]$ on samples with class label $Y = 1$ and take the difference between the maximum and the minimum probabilities of all compound groups. In addition, to compute FPR, we use the conditional probability $p_i = \Pr[X_i|Y = 0]$ on samples with $Y = 0$ and take the difference similarly. Thus, Justicia_enum allows us to compute different fairness metrics using a unified algorithmic framework.

Learning Fairness with ER-SSAT Encoding

In most practical problems, there can be exponentially many compound groups based on the different combinations of valuation to the protected attributes. Therefore, the enumera-

tion approach may suffer from scalability issues. Hence, we propose efficient SSAT encodings to *learn* the most favored group and the least favored group for given $\mathcal{M}$ and $\mathcal{D}$, and to compute their PPVs to measure different fairness metrics.

Learning the Most Favored Group. In an SSAT formula Φ, the order of quantification of the Boolean variables in the prefix carries distinct interpretation of the satisfying probability of Φ. In ER-SSAT formula, the probability of satisfying Φ is the *maximum* satisfying probability over the existentially quantified variables given the randomized quantified variables (by Rule 2, Sec.). In this paper, we leverage this property to compute the most favored group with the highest PPV. We consider the following ER-SSAT formula.

$$\Phi_{\mathsf{ER}} := \exists A_1, \ldots, \exists A_n, \mathfrak{R}^{p_1} X_1, \ldots, \mathfrak{R}^{p_m} X_m,\ \phi_{\hat{Y}}. \quad (2)$$

The CNF formula $\phi_{\hat{Y}}$ is the CNF translation of the classifier $\hat{Y} = 1$ without any specification of the compound protected group. Therefore, as we solve Φ_{ER}, we find the assignment to the existentially quantified variables $A_1 = a_1^{\max}, \ldots, A_n = a_n^{\max}$ for which the satisfying probability $\Pr[\Phi_{\mathsf{ER}}]$ is maximum. Thus, we compute the most favored group $\mathbf{a}_{\mathsf{fav}} \triangleq \{a_1^{\max}, \ldots, a_n^{\max}\}$ achieving the highest PPV.

Learning the Least Favored Group. In order to learn the least favored group in terms of PPV, we compute the *minimum* satisfying probability of the classifier $\phi_{\hat{Y}}$ given the random values of the non-protected variables $X_1, \ldots, X_m$. In order to do so, we have to solve a 'universal-random' (UR) SSAT formula (Eq. (3)) with universal quantification over the protected variables and randomized quantification over the non-protected variables (by Rule 3, Sec.).

$$\Phi_{\mathsf{UR}} := \forall A_1, \ldots, \forall A_n, \mathfrak{R}^{p_1} X_1, \ldots, \mathfrak{R}^{p_m} X_m,\ \phi_{\hat{Y}}. \quad (3)$$

A UR-SSAT formula returns the minimum satisfying probability of ϕ over the universally quantified variables in contrast to the ER-SSAT formula that returns the maximum satisfying probability over the existentially quantified variables. Due to practical issues to solve UR-SSAT formula, in this paper, we leverage the *duality* between UR-SSAT (Eq. (3)) and ER-SSAT formulas (Eq. (4))

$$\Phi'_{\mathsf{ER}} := \exists A_1, \ldots, \exists A_n, \mathfrak{R}^{p_1} X_1, \ldots, \mathfrak{R}^{p_m} X_m,\ \neg\phi_{\hat{Y}}. \quad (4)$$

and solve the UR-SSAT formula on the CNF ϕ using the ER-SSAT formula on the complemented CNF $\neg\phi$ (Littman, Majercik, and Pitassi 2001). Lemma 2 encodes this duality.

Lemma 2. Given Eq. (3) and (4), $\Pr[\Phi_{\mathsf{UR}}] = 1 - \Pr[\Phi'_{\mathsf{ER}}]$.

As we solve Φ'_{ER}, we obtain the assignment to the protected attributes $\mathbf{a}_{\mathsf{unfav}} \triangleq \{a_1^{min}, \ldots, a_n^{min}\}$ that maximizes Φ'_{ER}. If p is the maximum satisfying probability of Φ'_{ER}, according to Lemma 2, $1 - p$ is the minimum satisfying probability of Φ_{UR}, which is the PPV of the least favored group $\mathbf{a}_{\mathsf{unfav}}$. We present the algorithm for this learning approach, namely Justicia_learn in Algorithm 1 (Line 9–14).

In ER-SSAT formula of Eq. (4), we need to negate the classifier $\phi_{\hat{Y}}$ to another CNF formula $\neg\phi_{\hat{Y}}$. The naïve approach of negating a CNF to another CNF generates exponential number of new clauses. Here, we can apply Tseitin transformation that increases the clauses linearly while introducing linear number of new variables (Tseitin 1983). As an alternative, we also directly encode the classifier $\mathcal{M}$ for the negative class label $\hat{Y} = 0$ as a CNF formula and pass it to Φ'_{ER}, if possible. The last approach is generally more efficient than the other approaches as the resulting CNF is often smaller.

Example 0.3 (ER-SSAT encoding)**.** Here, we illustrate the ER-SSAT encodings for learning the most favored and the least favored group in presence of multiple protected groups. As the example in Figure 1 is degenerate for this purpose, we introduce another protected group 'sex $\in$ {male, female}'. Consider a Boolean variable S for 'sex' where the literal S denotes 'sex = male'. With this new protected attribute, let the classifier be $\mathcal{M} \triangleq (\neg H \vee I \vee S) \wedge (H \vee J)$, where A, H, I, J have same distributions as discussed in Example 0.2. Hence, we obtain the ER-SSAT formula of $\mathcal{M}$ to learn the most favored group: $\Phi_{\mathsf{ER}} = \exists S, \exists A, \mathfrak{R}^{0.41} H, \mathfrak{R}^{0.93} I, \mathfrak{R}^{0.09} J,\ (\neg H \vee I \vee S) \wedge (H \vee J)$.

As we solve Φ_{ER}, we learn that the assignment to the existential variables $\sigma(S) = 1, \sigma(A) = 0$, i.e. 'male individuals with age < 40' is the most favored group with PPV computed as $\Pr[\Phi_{\mathsf{ER}}] = 0.46$. Similarly, to learn the least favored group, we negate the CNF of the classifier $\mathcal{M}$ to obtain the following ER-SSAT formula: $\Phi_{\mathsf{ER}'} = \exists S, \exists A, \mathfrak{R}^{0.41} H, \mathfrak{R}^{0.93} I, \mathfrak{R}^{0.09} J,\ \neg((\neg H \vee I \vee S) \wedge (H \vee J))$.

Solving $\Phi_{\mathsf{ER}'}$, we learn the assignment $\sigma(S) = 0, \sigma(A) = 0$ and $\Pr[\Phi_{\mathsf{ER}'}] = 0.57$. Thus, 'female individuals with age < 40' constitute the least favored group with PPV: $1 - 0.57 = 0.43$. Thus, Justicia_learn allows us to learn the most and least favored groups and the corresponding discrimination.

We use the PPVs of the most and least favored groups to compute different fairness metrics. We next prove the equivalence of Justicia_enum and Justicia_learn in Lemma 3.

Lemma 3. Let $\Phi_{\mathbf{a}}$ be the RE-SSAT formula for computing the PPV of the compound protected group $\mathbf{a} \in A$. If Φ_{ER} is the ER-SSAT formula for learning the most favored group and Φ_{UR} is the UR-SSAT formula for learning the least favored group, then $\max_{\mathbf{a}} \Pr[\Phi_{\mathbf{a}}] = \Pr[\Phi_{\mathsf{ER}}]$ and $\min_{\mathbf{a}} \Pr[\Phi_{\mathbf{a}}] = \Pr[\Phi_{\mathsf{UR}}]$.

Theoretical Analysis: Error Bounds

We access the data generating distribution through finite number of samples observed from it. These finite sample set introduce errors in the computed probabilities of the randomised quantifiers being 1. These finite-sample errors in computed probabilities induce further errors in the computed positive predictive value (PPV) and fairness metrics. We next provide a bound on this finite-sample error.

Let us consider that $\hat{p_i}$ is the estimated probability of a Boolean variable B_i being assigned to 1 from k-samples and p_i is the true probability according to $\mathcal{D}$. Thus, the true satisfying probability p of Φ is the weighted sum of all satisfying assignments of the CNF ϕ: $p = \sum_\sigma \prod_{B_i \in \sigma} p_i$. This probability is estimated as $\hat{p}$ using k-samples from the data generating distribution $\mathcal{D}$ such that $\hat{p} \leq \epsilon_0 p$ for $\epsilon_0 \geq 1$.

Theorem 4. For an ER-SSAT problem, the sample complexity is given by $k = O\left((n + \ln(1/\delta))\frac{\ln m}{\ln \epsilon_0}\right)$, where $\frac{\hat{p}}{p} \leq \epsilon_0$ with probability $1 - \delta$ such that $\epsilon_0 \geq 1$.

Corollary 1. If k samples are considered from the data-generating distribution in Justicia such that $k = O\left((n + \ln(1/\delta))\frac{\ln m}{\ln \epsilon_0}\right)$, the estimated disparate impact $\hat{DI}$ and statistical parity $\hat{SP}$ satisfy, with probability $1 - \delta$, $\hat{DI} \leq \epsilon_0 DI, \quad \text{and} \quad \hat{SP} \leq 2\epsilon_0 SP$.

This implies that given a classifier $\mathcal{M} \triangleq \Pr(\hat{Y}|X, A)$ represented as a CNF formula and a data-generating distribution $(X, A, Y) \sim \mathcal{D}$, Justicia can verify independence and separation notion of fairness up to an error level ϵ_0 and $2\epsilon_0$ with probability $1 - \delta$. Thus, Justicia is a sound framework of fairness verification with high probability.

Practical Settings

We now relax the assumptions on access to Boolean classifiers and Boolean attributes, and extend Justicia to verify fairness metrics for more practical settings of decision trees, linear classifiers, and continuous attributes.

Extending to Decision Trees and Linear Classifiers. In the SSAT approach, we assume that the classifier $\mathcal{M}$ is represented as a CNF formula. We extend Justicia beyond CNF classifiers to decision trees and linear classifiers, which are widely used in the fairness studies (Zemel et al. 2013; Raff, Sylvester, and Mills 2018; Zhang and Ntoutsi 2019).

Binary decision trees are trivially encoded as CNF formulas. In the binary decision tree, each node in the tree is a literal. A *path from the root to the leaf* is a conjunction of literals and thus, a *clause*. The *tree* itself is a disjunction of all paths and thus, a *DNF (Disjunctive Normal Form)*. In order to derive a CNF of a decision tree, we first construct a DNF by including all paths terminating at leaves with negative class label ($\hat{Y} = 0$) and then complement the DNF to CNF using De Morgan's rule.

Linear classifiers on Boolean attributes are encoded into CNF formulas using pseudo-Boolean encoding (Philipp and Steinke 2015). We consider a linear classifier $W^T X + b \geq 0$ on Boolean attributes X with weights $W \in \mathbb{R}^{|X|}$ and bias $b \in \mathbb{R}$. We first normalize W and b in $[-1, 1]$ and then round to integers so that the decision boundary becomes a pseudo-Boolean constraint. We then apply pseudo-Boolean constraints to CNF translation to encode the decision boundary to CNF. This encoding usually introduces additional Boolean variables and results in large CNF. In order to generate a smaller CNF, we can trivially apply thresholding on the weights to consider attributes with higher weights only. For instance, if the weight $|w_i| \leq \lambda$ for a threshold $\lambda \in \mathbb{R}^+$ and $w_i \in W$, we can set $w_i = 0$. Thus, the attributes with lower weights and thus, less importance do not appear in the encoded CNF. Moreover, all introduced variables in this CNF translation are given existential ($\exists$) quantification and they appear in the inner-most position in the prefix of the SSAT formula. Thus, the presented ER-SSAT formulas become effectively ERE-SSAT formulas.

Extending to Continuous Attributes. In practical problems, attributes are generally real-valued or categorical but classifiers, which are naturally expressed as CNF such as (Ghosh, Malioutov, and Meel 2020), are generally trained on a Boolean abstraction of the input attributes. In order to perform this Boolean abstraction, each categorical attribute is one-hot encoded and each real-valued attribute is discretised into a set of Boolean attributes (Lakkaraju et al. 2019; Ghosh, Malioutov, and Meel 2020).

For a binary decision tree, each attribute, including the continuous ones, is compared against a constant at each internal node of the tree. We fix a Boolean variable for each internal node, where the Boolean assignment to the variable decides one of the two branches to choose from the current node.

Linear classifiers are generally trained on continuous attributes, where we apply the following discretization. Let us consider a continuous attribute X_c, where w is its weight during training. We discretize X_c to a set $\mathbf{B}$ of Boolean attributes and recalculate the weight of each variable in $\mathbf{B}$ based on w. For the discretization of X_c, we consider the interval-based approach[2]. For each interval in the continuous space of X_c, we consider a Boolean variable $B_i \in \mathbf{B}$, such that B_i is assigned TRUE when the attribute-value of X_c lies within the i^{th} interval and B_i is assigned FALSE otherwise. Following that, we assign the weight of B_i to be $\mu_i \times w$, when μ_i is the mean of the i^{th} interval and B_i is TRUE. We can show that if we consider infinite number of intervals, $X_c \approx \sum_i \mu_i B_i$.

Empirical Performance Analysis

In this section, we discuss the empirical studies to evaluate the performance of Justicia in verifying different fairness metrics. We first discuss the experimental setup and the objective of the experiments and then evaluate the experimental results.

Experimental Setup

We have implemented a prototype of Justicia in Python (version 3.7.3). The core computation of Justicia relies on solving SSAT formulas using an off-the-shelf SSAT solver. To this end, we employ the state of the art RE-SSAT solver of (Lee, Wang, and Jiang 2017) and the ER-SSAT solver of (Lee, Wang, and Jiang 2018). Both solvers output the exact satisfying probability of the SSAT formula.

For comparative evaluation of Justicia, we have experimented with two state-of-the-art distributional verifiers FairSquare and VeriFair, and also a sample-based fairness measuring tool: AIF360. In the experiments, we have studied three type of classifiers: CNF learner, decision trees and logistic regression classifier. Decision tree and logistic regression are implemented using scikit-learn module of Python (Pedregosa et al. 2011) and we use the MaxSAT-based CNF learner IMLI of (Ghosh and Meel 2019). We have used the PySAT library (Ignatiev, Morgado, and Marques-Silva 2018) for encoding the decision function of the logistic regression classifier into a CNF formula. We have also verified two fairness-enhancing algorithms: reweighing algorithm (Kamiran and Calders 2012) and the optimized pre-processing algorithm (Calmon et al. 2017). We have experimented on multiple datasets containing multiple protected attributes: the UCI[3]

[2]Our implementation is agnostic to any discretization technique.

[3]http://archive.ics.uci.edu/ml

Metric	Exact	Justicia	FairSquare	VeriFair	AIF360
Disparate impact	0.26	0.25	0.99	0.99	0.25
Stat. parity	0.53	0.54	—	—	0.54

Table 1: Results on synthetic benchmark. '—' refers that the verifier cannot compute the metric.

Dataset	Ricci		Titanic		COMPAS		Adult	
Classifier	DT	LR	DT	LR	DT	LR	DT	LR
Justicia	0.1	0.2	0.1	0.9	0.1	0.2	0.2	1.0
FairSquare	4.8	—	16.0	—	36.9	—	—	—
VeriFair	5.3	2.2	1.2	0.8	15.9	11.3	295.6	61.1

Table 2: Scalability of different verifiers in terms of execution time (in seconds). DT and LR refer to decision tree and logistic regression respectively. '—' refers to timeout.

Adult and German-credit dataset, ProPublica's COMPAS recidivism dataset (Angwin et al. 2016), Ricci dataset (McGinley 2010), and Titanic dataset[4].

Our empirical studies have the following objectives:

1. How accurate and scalable Justicia is with respect to existing fairness verifiers, FairSquare and VeriFair?
2. Can Justicia verify the effectiveness of different fairness-enhancing algorithms on different datasets?
3. Can Justicia verify fairness in the presence of compound sensitive groups?
4. How robust is Justicia in comparison to sample-based tools like AIF360 for varying sample sizes?
5. How do the computational efficiencies of Justicia_learn and Justicia_enum compare?

Our experimental studies validate that Justicia is more accurate and scalable than the state-of-the-art verifiers FairSquare and VeriFair. Justicia is able to verify the effectiveness of different fairness-enhancing algorithms for multiple fairness metrics, and datasets. Justicia achieves scalable performance in the presence of compound sensitive groups that the existing verifiers cannot handle. Justicia is also more robust than the sample-based tools such as AIF360. Finally, Justicia_learn is significantly efficient in terms of runtime than Justicia_enum.

Experimental Analysis

Accuracy: Less Than 1%-error. In order to assess the accuracy of different verifiers, we have considered the decision tree in Figure 1 for which the fairness metrics are analytically computable. In Table 1, we show the computed fairness metrics by Justicia, FairSquare, VeriFair, and AIF360. We observe that Justicia and AIF360 yield more accurate estimates of DI and SP compared against the ground truth with less than 1% error. FairSquare and VeriFair estimate the disparate impact to be 0.99 and thus, being unable to verify the fairness violation. Thus, Justicia is significantly accurate than the existing formal verifiers: FairSquare and VeriFair.

Scalability: 1 to 3 Orders of Magnitude Speed-up. We have tested the scalability of Justicia, FairSquare, and VeriFair on practical benchmarks with a timeout of 900 seconds and reported the execution time of these verifiers on decision tree and logistic regression in Table 2. We observe that Justicia shows impressive scalability than the competing verifiers. Particularly, Justicia is 1 to 2 orders of magnitude faster than FairSquare and 1 to 3 orders of magnitude faster than VeriFair. Additionally, FairSquare times out in most benchmarks. Thus, Justicia is not only accurate but also scalable than the existing verifiers.

Verification: Detecting Compounded Discrimination in Protected Groups. We have tested Justicia for datasets consisting of multiple protected attributes and reported the results in Figure 2. Justicia operates on datasets with even 40 compound protected groups and can potentially scale more than that while the state-of-the-art fairness verifiers (e.g., FairSquare and VeriFair) consider a single protected attribute. Thus, Justicia removes an important limitation in practical fairness verification. Additionally, we observe in most datasets the disparate impact decreases and thus, discrimination increases as more compound protected groups are considered. For instance, when we increase the total groups from 5 to 40 in the Adult dataset, disparate impact decreases from around 0.9 to 0.3, thereby detecting higher discrimination. Thus, Justicia detects that the marginalized individuals of a specific type (e.g., 'race') are even more discriminated and marginalized when they also belong to a marginalized group of another type (e.g., 'sex').

Verification: Fairness of Algorithms on Datasets. We have experimented with two fairness-enhancing algorithms: the reweighing (RW) algorithm and the optimized-preprocessing (OP) algorithm. Both of them pre-process to remove statistical bias from the dataset. We study the effectiveness of these algorithms using Justicia on three datasets each with two different protected attributes. In Table 3, we report different fairness metrics on logistic regression and decision tree. We observe that Justicia verifies fairness improvement as the bias mitigating algorithms are applied. For example, for the Adult dataset with 'race' as the protected attribute, disparate impact increases from 0.23 to 0.85 for applying the reweighing algorithm on logistic regression classifier. In addition, statistical parity decreases from 0.09 to 0.01, and equalized odds decreases from 0.13 to 0.03, thereby showing the effectiveness of reweighing algorithm in all three fairness metrics. Justicia also finds instances where the fairness algorithms fail, specially when considering the decision tree classifier. Thus, Justicia enables verification of different fairness enhancing algorithms in literature.

Robustness: Stability to Sample Size. We have compared the robustness of Justicia with AIF360 by varying the sample-size and reporting the standard deviation of different fairness metrics. In Figure 3, AIF360 shows higher standard deviation for lower sample-size and the value decreases as the sample-size increases. In contrast, Justicia shows significantly lower

[4]https://www.kaggle.com/c/titanic

Classifier	Dataset →	Adult						COMPAS					
	Protected →	Race			Sex			Race			Sex		
	Algorithm →	orig.	RW	OP	orig.	RW	OP	orig.	RW	OP	orig.	RW	OP
Logistic regression	Disparte impact	0.23	**0.85**	**0.59**	0.03	**0.61**	**0.62**	0.34	**0.36**	**0.47**	0.48	**0.80**	**0.74**
	Stat. parity	0.09	**0.01**	**0.05**	0.16	**0.04**	**0.03**	0.39	**0.33**	**0.21**	0.23	**0.09**	**0.10**
	Equalized odds	0.13	**0.03**	**0.10**	0.30	**0.02**	**0.06**	0.38	**0.33**	**0.18**	0.17	0.19	**0.07**
Decision tree	Disparte impact	0.82	0.60	0.67	0.00	**0.73**	**0.95**	0.61	0.58	0.57	0.94	0.78	0.63
	Stat. parity	0.02	0.05	0.04	0.14	**0.05**	**0.01**	0.18	**0.17**	**0.17**	0.02	0.09	0.18
	Equalized odds	0.07	**0.05**	**0.03**	0.47	**0.03**	**0.04**	0.17	**0.16**	**0.16**	0.07	**0.05**	0.16

Table 3: Verification of different fairness enhancing algorithms for multiple datasets and classifiers using Justicia. Numbers in bold refer to fairness improvement compared against the unprocessed (orig.) dataset. RW and OP refer to reweighing and optimized-preprocessing algorithm respectively.

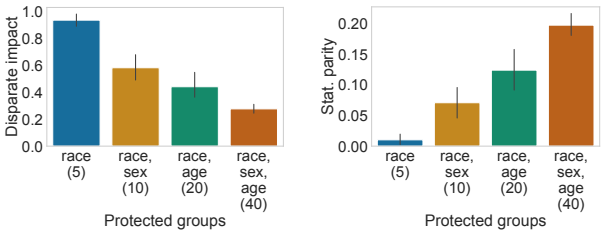

Figure 2: Fairness metrics measured by Justicia for different protected groups in the Adult dataset. The number within parenthesis in the xticks denotes total compound groups.

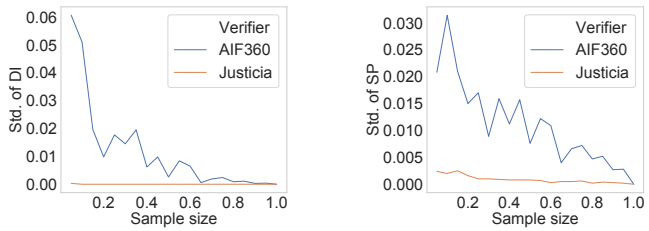

Figure 3: Standard deviation in estimation of disparate impact (DI) and stat. parity (SP) for different sample sizes. Justicia is more robust with variation of sample size than AIF360.

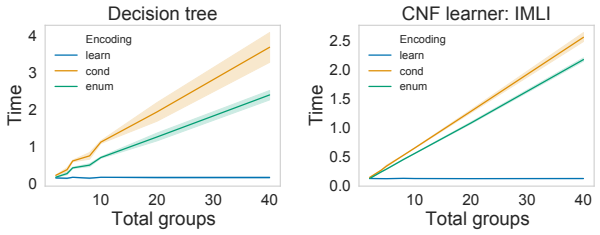

Figure 4: Runtime comparison of different encodings while varying total protected groups in the Adult dataset.

($\sim 10\times$ to $100\times$) standard deviation for different sample-sizes. The reason is that AIF360 empirically measures on a fixed test dataset whereas Justicia provides estimates over the data generating distribution. Thus, Justicia is more robust than the sample-based verifier AIF360.

Comparative Evaluation of Different Encodings. While both Justicia_enum and Justicia_learn have the same output according to Lemma 3, Justicia_learn encoding improves exponentially in runtime than Justicia_enum encoding on both decision tree and Boolean CNF classifiers as we vary the total compound groups in Figure 4. Justicia_cond (conditional probabilities w.r.t. protected groups) also has an exponential trend in runtime similar to Justicia_enum. This analysis justifies that the naïve enumeration-based approach cannot verify large-scale fairness problems containing multiple protected attributes, and Justicia_learn is a more efficient approach for practical use.

Discussion and Future Work

Though formal verification of different fairness metrics of an ML algorithm for different datasets is an important question, existing verifiers are not scalable, accurate, and extendable to non-Boolean protected attributes. We propose a stochastic SAT-based approach, Justicia, that formally verifies independence and separation metrics of fairness for different classifiers and distributions for compound protected groups. Experimental evaluations demonstrate that Justicia achieves *higher accuracy* and *scalability* in comparison to the state-of-the-art verifiers, FairSquare and VeriFair, while yielding *higher robustness* than the sample-based tools, such as AIF360.

Our work opens up several new directions of research. One direction is to develop SSAT models and verifiers for popular classifiers like Deep networks and SVMs. Other direction is to develop SSAT solvers that can accommodate continuous variables and conditional probabilities by design.

Acknowledgments

We are grateful to Jie-Hong Roland Jiang and Teodora Baluta for the useful discussion at the earlier stage of this project. We thank Nian-Ze Lee for the technical support of the SSAT solvers. This work was supported in part by the National Research Foundation Singapore under its NRF Fellowship Programme [NRF- NRFFAI1-2019-0004] and the AI Singa-

pore Programme [AISG-RP-2018-005], and NUS ODPRT Grant [R-252-000-685-13]. The computational work for this article was performed on resources of Max Planck Institute for Software Systems, Germany and the National Supercomputing Centre, Singapore. Debabrota Basu was funded by WASP-NTU grant of the Knut and Alice Wallenberg Foundation during the initial phase of this work.

References

Albarghouthi, A.; D'Antoni, L.; Drews, S.; and Nori, A. V. 2017. FairSquare: probabilistic verification of program fairness. *Proceedings of the ACM on Programming Languages* 1(OOPSLA): 1–30.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine bias risk assessments in criminal sentencing. *ProPublica, May* 23.

Bastani, O.; Zhang, X.; and Solar-Lezama, A. 2019. Probabilistic verification of fairness properties via concentration. *Proceedings of the ACM on Programming Languages* 3(OOPSLA): 1–27.

Bellamy, R. K. E.; Dey, K.; Hind, M.; Hoffman, S. C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilovic, A.; Nagar, S.; Ramamurthy, K. N.; Richards, J.; Saha, D.; Sattigeri, P.; Singh, M.; Varshney, K. R.; and Zhang, Y. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. URL https://arxiv.org/abs/1810.01943.

Biere, A.; Heule, M.; and van Maaren, H. 2009. *Handbook of satisfiability*, volume 185. IOS press.

Calmon, F.; Wei, D.; Vinzamuri, B.; Ramamurthy, K. N.; and Varshney, K. R. 2017. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, 3992–4001.

Chakraborty, S.; Fremont, D. J.; Meel, K. S.; Seshia, S. A.; and Vardi, M. Y. 2014. Distribution-aware sampling and weighted model counting for SAT. *arXiv preprint arXiv:1404.2984* .

Chakraborty, S.; Meel, K. S.; and Vardi, M. Y. 2013. A scalable approximate model counter. In *International Conference on Principles and Practice of Constraint Programming*, 200–216. Springer.

Chavira, M.; and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence* 172(6-7): 772–799.

Chouldechova, A.; and Roth, A. 2020. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM* 63(5): 82–89.

Corbett-Davies, S.; and Goel, S. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023* .

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.

Fremont, D. J.; Rabe, M. N.; and Seshia, S. A. 2017. Maximum Model Counting. In *AAAI*, 3885–3892.

Galhotra, S.; Brun, Y.; and Meliou, A. 2017. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 498–510.

Ghosh, B.; Malioutov, D.; and Meel, K. S. 2020. Classification Rules in Relaxed Logical Form. In *Proceedings of ECAI*.

Ghosh, B.; and Meel, K. S. 2019. IMLI: An Incremental Framework for MaxSAT-Based Learning of Interpretable Classification Rules. In *Proc. of AIES*.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, 3315–3323.

Huang, J.; et al. 2006. Combining Knowledge Compilation and Search for Conformant Probabilistic Planning. In *ICAPS*, 253–262.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2018. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In *SAT*, 428–437. doi:10.1007/978-3-319-94144-8_26. URL https://doi.org/10.1007/978-3-319-94144-8_26.

Kamiran, F.; and Calders, T. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33(1): 1–33.

Kamiran, F.; Karim, A.; and Zhang, X. 2012. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, 924–929. IEEE.

Kautz, H. A.; Selman, B.; et al. 1992. Planning as Satisfiability. In *ECAI*, volume 92, 359–363. Citeseer.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual fairness. In *Advances in neural information processing systems*, 4066–4076.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Leskovec, J. 2019. Faithful and Customizable Explanations of Black Box Models. In *Proc. of AIES*.

Lee, N.-Z.; and Jiang, J.-H. R. 2018. Towards formal evaluation and verification of probabilistic design. *IEEE Transactions on Computers* 67(8): 1202–1216.

Lee, N.-Z.; Wang, Y.-S.; and Jiang, J.-H. R. 2017. Solving Stochastic Boolean Satisfiability under Random-Exist Quantification. In *IJCAI*, 688–694.

Lee, N.-Z.; Wang, Y.-S.; and Jiang, J.-H. R. 2018. Solving Exist-Random Quantified Stochastic Boolean Satisfiability via Clause Selection. In *IJCAI*, 1339–1345.

Littman, M. L.; Majercik, S. M.; and Pitassi, T. 2001. Stochastic boolean satisfiability. *Journal of Automated Reasoning* 27(3): 251–296.

Majercik, S. M. 2007. APPSSAT: Approximate probabilistic planning using stochastic satisfiability. *International Journal of Approximate Reasoning* 45(2): 402–419.

Majercik, S. M.; and Boots, B. 2005. DC-SSAT: a divide-and-conquer approach to solving stochastic satisfiability problems efficiently. In *AAAI*, 416–422.

McGinley, A. C. 2010. Ricci v. DeStefano: A Masculinities Theory Analysis. *Harv. JL & Gender* 33: 581.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* .

Narodytska, N.; Ignatiev, A.; Pereira, F.; Marques-Silva, J.; and RAS, I. 2018. Learning Optimal Decision Trees with SAT. In *IJCAI*, 1362–1368.

Papadimitriou, C. H. 1985. Games against nature. *Journal of Computer and System Sciences* 31(2): 288–301.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12(Oct): 2825–2830.

Philipp, T.; and Steinke, P. 2015. PBLib–a library for encoding pseudo-boolean constraints into CNF. In *International Conference on Theory and Applications of Satisfiability Testing*, 9–16. Springer.

Raff, E.; Sylvester, J.; and Mills, S. 2018. Fair forests: Regularized tree induction to minimize model bias. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 243–250.

Sang, T.; Bacchus, F.; Beame, P.; Kautz, H. A.; and Pitassi, T. 2004. Combining Component Caching and Clause Learning for Effective Model Counting. *SAT* 4: 7th.

Tseitin, G. S. 1983. On the complexity of derivation in propositional calculus. In *Automation of reasoning*, 466–483. Springer.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *International Conference on Machine Learning*, 325–333.

Zhang, B. H.; Lemoine, B.; and Mitchell, M. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335–340.

Zhang, W.; and Ntoutsi, E. 2019. Faht: an adaptive fairness-aware decision tree classifier. *arXiv preprint arXiv:1907.07237* .