Automated Feature Engineering for Algorithmic Fairness

Ricardo Salazar Diaz
TU Berlin
ricardo.salazar@alumni.tu-berlin.de

Felix Neutatz
TU Berlin
f.neutatz@tu-berlin.de

Ziawasch Abedjan
Leibniz Universität Hannover
L3S Research Center
abedjan@dbs.uni-hannover.de

ABSTRACT

One of the fundamental problems of machine ethics is to avoid the perpetuation and amplification of discrimination through machine learning applications. In particular, it is desired to exclude the influence of attributes with sensitive information, such as gender or race, and other causally related attributes on the machine learning task. The state-of-the-art bias reduction algorithm Capuchin breaks the causality chain of such attributes by adding and removing tuples. However, this horizontal approach can be considered invasive because it changes the data distribution. A vertical approach would be to prune sensitive features entirely. While this would ensure fairness without tampering with the data, it could also hurt the machine learning accuracy. Therefore, we propose a novel multi-objective feature selection strategy that leverages feature construction to generate more features that lead to both high accuracy and fairness. On three well-known datasets, our system achieves higher accuracy than other fairness-aware approaches while maintaining similar or higher fairness.

1 INTRODUCTION

Algorithms might reinforce biases against groups of people that have been historically discriminated against [45, 46]. Examples include gender bias in machine learning (ML) applications on online advertising [11] and Google image search for occupations [28].

There are two main approaches to address data bias: associational and causal. Associational approaches link the sensitive feature, such as nationality, religion, or race, and the algorithm's outcomes through statistical measures [5, 15, 22]. This approach neglects the influence of sensitive features on other features, leading to paradoxical conclusions [36]. Causal approaches consider a causal structure on the data that allows for causal links between sensitive features, nonsensitive features, the target, and the predictions. Nonetheless, causal approaches typically assume knowledge of the underlying causal structure, which is unrealistic in practice [42]. Therefore, Salimi et al. [42] proposed to let the user categorize features as sensitive, admissible, or inadmissible. The admissible set contains features through which the user allows the influence of the sensitive feature on the classifier's predictions. Conversely, the inadmissible set is composed of features that potentially leak the bias of the sensitive attribute through a mapping. They further propose the system Capuchin (CA), which repairs the data that does not match the user's feature categorization by adding or removing tuples.

Although CA outperforms state-of-the-art (SOA) associational and causal approaches, it has three drawbacks. First, CA overfits fairness. It modifies the training set probability distribution by deleting and inserting tuples and then learns an unbiased classifier to predict unseen instances. The dissimilarities between the probability distributions of the "repaired" dataset used for training and unseen data lead to fairness overfitting. Second, CA's logic requires the binning of numerical features. Depending on the sizes and number of bins, CA might delete more or fewer tuples provoking a loss in classification accuracy. Third, the binning strategy is dataset-dependent, requiring previous knowledge of the problem and the data by the user.

Instead of addressing fairness repair with a tuple-wise, i.e., horizontal, approach as CA does, we propose a new approach that addresses the same problem with a feature-wise, i.e., vertical, strategy. A naive solution would drop sensitive and inadmissible features. This approach would successfully remove bias and avoid fairness overfitting, but might hurt the classification accuracy because of potential information loss.

To achieve both high accuracy and fairness, we propose to extract as much unbiased information as possible from inadmissible features using feature construction (FC) methods that apply non-linear transformations. Thus one can use FC first to generate more possible candidate features and then to drop inadmissible features and optimize for fairness and accuracy. Adapting existing feature generation frameworks [23, 27], we propose a two-phase multi-objective feature selection (FS) strategy that generates a feature set that simultaneously leads to high accuracy and fairness.

Finding a unique feature set that optimizes the trade-off between fairness and accuracy is challenging. By constructing new features, the search space rapidly grows depending on the number of original features. Thus, an exhaustive search, evaluating all objectives on the possible feature combinations, is infeasible. Therefore, a search strategy must consider two potentially competing objectives, and the selection strategy must optimize the trade-off between these objectives. In this paper, we demonstrate a greedy search strategy that fulfills this goal and outperforms SOA preprocessing methods for fairness optimization.

- We show that we can extract unbiased information from biased features by applying human-understandable transformations (Section 3.1). We found that FC through multiplication and group by aggregations are successful for this task.
- We propose a mixed-initiative approach [24] where our system and the user collaborate to balance the trade-off between accuracy and fairness to the user's needs (Section 3.4).
- We present a series of experiments on known datasets (Section 4) showing how different feature transformations, classification models, feature search space pruning, and weights for fairness and accuracy influence the performance of our approach.

2 FOUNDATION

First, we review general families of bias reduction algorithms. These methods address bias at different stages of the ML pipeline and

measure it differently. Finally, we formally define the problem of multi-objective bias reduction.

2.1 Bias Reduction Algorithms

To address algorithmic discrimination, different bias reduction strategies have been discussed and categorized in pre-processing methods, in-processing methods, and post-processing methods.

Pre-processing methods remove the bias by modifying the training data. Since pre-processing methods rely on the training data, they are independent of the ML model. This means that any ML model trained with the resulting training set should show unbiased predictions according to certain fairness criteria. Strategies in this category include cell-mapping of the nonsensitive features by randomized distortions [13], data-augmentation [42], and adversarial learning of feature representations [7, 49]. In-processing methods introduce constraints and regularization terms into the classifier's loss function. Usually, these terms try to moderate the link between the sensitive feature and the algorithm outcomes. For instance, one can add a regularization term to the logistic regression objective's function, penalizing mutual information between the sensitive feature and the predictions [26]. In-processing approaches are model dependent requiring to adjust the constraints and regularization terms depending on the classification model. Post-processing methods modify the outcomes of the classifiers. Strategies include prediction flipping [22] and threshold selection depending on the values of the sensitive attributes [9].

2.2 Measuring Fairness

Several fairness measures have been proposed to capture two legal definitions of discrimination [4]. First, *disparate treatment*, refers to discrimination based on the membership of individuals in a particular group, for instance, neglecting someone's medical treatment because of race. Second, *disparate impact* refers to discrimination in contexts where decisions are not based on sensitive features, and yet they have a larger impact on one or more groups of minorities.

Measures to quantify *disparate treatment* and *disparate impact* can be *associational* or *causal*. Associational measures aim to discover unfair situations through statistical inequalities of the outcomes between the different groups of the sensitive feature. *Demographic Parity (DP)*, *Conditional Statistical Parity*, and *Equalized Odds (EO)* [22] are the most representative measures in this category. Causality approaches analyze whether the classifier's predictions are influenced by the *sensitive feature* through *nonsensitive features* [3]. In contrast to causal approaches, associational approaches are inaccurate in identifying unfair situations in certain cases [42]. According to Salimi et al. [42], *justifiable fairness* is the strongest causal notion of fairness that is testable on data. For a classification task with the features V, the target O, and the sensitive feature S, one can classify the remaining features $V \setminus \{S\}$ as admissible or inadmissible. Through *admissible features*, the user allows an influence of the *sensitive feature* on the outcomes. Unlike other causality-based approaches that assume a fixed causal structure to prohibit paths from the *sensitive feature* to the outcomes [33], the feature categorization proposed by Salimi et al. allows for flexibility regarding the causal structure and it imposes the conditions for an ML model to

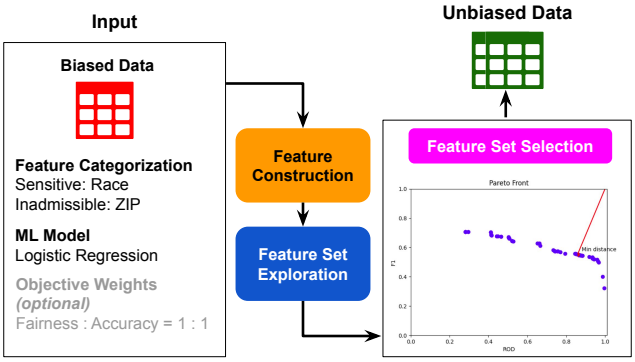

Figure 1: System Workflow.

be *justifiable fair* for these categories. To measure *justifiable fairness*, one can use the *Ratio of Observational Discrimination (ROD)* [42]:

$$\delta(S;O|a_b) = \frac{P(O=1|S=0,A_b=a_b)P(O=0|S=1,A_b=a_b)}{P(O=0|S=0,A_b=a_b)P(O=1|S=1,A_b=a_b)} \quad (1)$$

where S represents the membership to the protected group in the sensitive feature, O the outcomes of the model, and A the admissible set of features. A_b refers to the subset composed by admissible features influencing the outcomes of the classifier. Formally, $A_b = MB(O) - I$, where $MB(O)$ is the *Markov blanket* of the outcomes, and I is the inadmissible set. Modeling the causal relationship of attributes as a direct acyclic graph, the *Markov blanket* comprises the parents, children, and the co-parents of an attribute node. The *Markov blanket* of a node is the minimal set of nodes that isolates the node from the graph - the *Markov blanket* shields the node from the influence of other variables. If $\delta(S;O|a_b) = 1$ then there is no observational evidence that the algorithm discriminates subjects with similar characteristics a_b. If $\delta(S;O|a_b)$>1 the model potentially discriminates against the sensitive group.

2.3 Problem

We aim to minimize prediction error and algorithmic bias for classification tasks. Given a biased dataset $D = (S, A, I)$, where S is the sensitive or protected attribute, A is the set of admissible features and I is the set of inadmissible features, we aim to enrich the feature vector in a way that a subset $X \subseteq V$, where V is the set of constructed features from $A \cup I \cup S$, minimizes algorithmic bias and prediction error for an algorithm mapping X to an outcome O.

We adopt the causal perspective of *justifiable fairness* and use the *Ratio of Observational Discrimination (ROD)* $\delta(S;O|a_b)$ as the objective function $g_1 = |1 - \delta(S;O|a_b)|$ to minimize algorithmic bias. Moreover, we use the F1 score as the objective function g_2 to minimize prediction error. The F1 score is defined as $F1 = 2 \times \frac{(P \times R)}{(P+R)}$, where precision (P) is the fraction of correctly positive classified instances and recall (R) is the fraction of the truly positive classified instances that are discovered. Given the objective functions $g_1(X)$ and $g_2(X)$, the optimization problem can be written as:

$$maximize\ G([-g_1(X), g_2(X)]). \quad (2)$$

Table 1: Std(workclass = Local-gov) GroupyBy marital-status

Class	Sex	Std(workclass == "Local-gov") GroupBy marital-status
<=50K	Female	0.24 ± 0.03
	Male	0.24 ± 0.02
>50K	Female	0.25 ± 0.02
	Male	0.25 ± 0.01

3 AUTOMATED FAIR FEATURE ENGINEERING

Figure 1 illustrates the workflow of FairExp (FAIRness EXPlorer), which transforms a biased dataset D into a bias-reduced dataset D'. First, the user provides D and specifies the sensitive and inadmissible features. Optionally, the user can provide weights for the objectives accuracy and fairness. The workflow consists of three main steps: FC, feature set exploration, and feature set selection. FairExp constructs new features from the original features by recursively applying transformations. FC enables the extraction of unbiased information even from biased features as we describe in Section 3.1. Note that we do not prune any biased features in the construction phase.

In contrast to existing work, which applies simultaneous optimization of accuracy and fairness [12] and can run into a local minimum, we propose a two-phase exploration strategy. Having in mind that smaller and less complex feature sets generalize better, we first try to find the minimal set of features that maximizes accuracy. For this purpose, we propose a definition of feature set DL that serves our purpose. In the second phase, we trim the feature set to fulfill the fairness requirement, as we explain in Section 3.2. The exploration yields a broad candidate set of possible feature sets in the Pareto front trading off accuracy and fairness. As detailed in Section 3.4, we apply a geometric solution similar to the "elbow method" [6] to pick the optimal feature set while considering the objective weighting if specified. Finally, we return the preprocessed, bias-reduced dataset to the user.

3.1 Feature Construction

To reduce bias one has to avoid inadmissible features. However, if we drop all inadmissible features, we lose too much information. So, the idea is to use some information about inadmissible features by leveraging information from unbiased relationships with other features. Salimi et al. [41] showed that the Adult dataset is inconsistent because it reports household incomes for married individuals, and there are more male married individuals in the sample. Thus, the categorical feature *marital-status* is considered inadmissible because it could bias the outcomes in favor of married males. However, as observed from Table 1, the feature *Std(workclass == "Local-gov") GroupyBy marital-status* is not biased towards a specific sex and benefits the classification accuracy. To obtain features, such as *Std(workclass == "Local-gov") GroupyBy marital-status*, we apply transformations to the original features recursively.

FairExp constructs new features to increase the classification accuracy by applying transformations on the original features. We leverage the same set of transformations as proposed by Katz et al. [27]. The standard operations $+, *, 1/, -1*$, and *log* model arithmetic relationships among features. The encoding transformations equal-range discretization, min-max scaling, and one-hot encoding facilitate learning for ML models. The partial aggregates GroupyByThen{Min, Max, Mean, Standard Deviation} allow FairExp to model conditional relationships among features.

Theoretically, we can apply infinitely many transformations recursively. However, we stop once a feature reaches a certain description length (DL). Inspired by the *Minimum Description Length* (MDL) principle [39], we recursively define the DL of a feature:

$$\mathrm{DL}(\{f_m\}) = 1, f_m \in F_0, \tag{3}$$

$$\mathrm{DL}(F_a) = \sum_{fm \in F_a} \mathrm{DL}(f_m), \tag{4}$$

$$\mathrm{DL}(\{F_a, F_b\}) = \mathrm{DL}(F_a) + \mathrm{DL}(F_b), \tag{5}$$

$$\mathrm{DL}(\{t(F_a)\}) = \mathrm{DL}(F_a) + 1, t \in T, \tag{6}$$

$$\text{where } F_a, F_b \text{ are any feature set.} \tag{7}$$

For instance, we would consider a raw feature A to have DL 1. For every transformation that we apply, we increment the DL accordingly. E.g. $log(A)$ has DL 2 and *Max(A) GroupBy B* has DL 3.

Instead of generating all possible combinations of features (brute force) and transformations for the specified DL level, we leverage a linear algebra solver [32] to avoid generating algebraically equivalent features, such as the commutative property $a + b = b + a$, distributive property, e.g., $(a \cdot b) + (a \times c) = a \cdot (b + c)$, associative property, e.g., $(a + b) + c = (b + c) + a$, idempotent property, e.g., MinMaxScaling(MinMaxScaling(a)) = MinMaxScaling(a), input-dependent idempotent property, e.g, max(max(a) Group By b) Group By b = max(a) Group By b, and invertibility property, e.g. $a/b \cdot b = a$. Additionally, we prune all constant features, e.g. $a/a = 1$. Despite these optimizations, there is still an exponential growth of the feature space. For instance, in the German Credit dataset with 21 raw features (DL 1), we would still face 1k features of DL 2, 1.8k features of DL 3, and 7.5k features of DL 4. In Section 3.5, we describe how we parallelize our algorithm to quickly cover thousands of constructed features.

3.2 Two-Phase Feature Set Exploration

Algorithm 1 describes in detail how we gather the most promising candidate feature sets in a two-phase approach. We search for feature sets that contain few and simple features. First, we add features that benefit accuracy to find the least complex model that maximizes accuracy and then we remove biased features.

Algorithm 1 takes a dataset D, the user-specified feature categorization, the model, and the constructed features C as input. In the first phase (Lines 4 - 11), the algorithm iteratively adds constructed features to an initially empty feature set F as long as they improve accuracy on a validation set. Note that the constructed features are sorted by increasing DL - starting with the raw features and ending with the features that are the result of more complex transformations. This way our system prefers simple features, which are easier to understand by users, over complex ones. We further enforce this principle by applying the floating procedure [38] (Lines 8 - 11) to further reduce the size of the resulting feature sets. Floating removes redundant features from the current feature set if possible without harming the accuracy. After evaluating all constructed

features, phase 1 yields one large feature set with the highest validation accuracy. The complexity of the first phase is theoretically exponential but capped by the specified description lengths. Given n raw features and m operations and a cap of l as the maximum description length. The number of possible feature combinations are all the combinations of single raw features and permutations of raw features and operations of higher ranks: $O(n + \sum_{i=2}^{l} (n+m)^i)$. In phase 2, we start with this accurate feature set and incrementally remove one feature at a time if it improves fairness. Each time that we remove a feature, we apply floating. Here, floating means that we verify whether adding one of the removed accurate features benefits fairness of the current feature set. If that is the case, we add it to the feature set again. Floating is necessary because fairness is a non-monotonic measure and therefore removing one feature might significantly affect the properties of the underlying feature set. The second phase terminates once every feature in the set F has been considered for removal. The complexity of the second phase is $O(|F|^2)$ where $|F|$ is the number of the features that were found in the first phase. Therefore, the complexity of the entire algorithm is dominated by the first phase because usually $|F| << |C|$.

From all explored feature sets, we select those on the Pareto front. In the next section, we discuss how to choose one feature set from the candidates in this Pareto front.

Algorithm 1 Two-Phase Feature Set Exploration

Input: dataset D, feature categorization, model, constructed features C.
Output: evaluated feature subsets O.

1: $C \leftarrow$ the list of all constructed features sorted by increasing DL
2: $F \leftarrow \emptyset$ ▷ the current feature subset
3: $O \leftarrow \emptyset$

Phase 1: Exploration for accurate feature sets

4: **for** c *in* C **do**
5: **if** $F1(F \cup c) > F1(F)$ **then**
6: $F \leftarrow F \cup c$
7: $O \leftarrow O \cup (F, F1(F), ROD(F))$
8: **for** f *in* F **do** ▷ floating
9: **if** $F1(F \setminus f) > F1(F)$ **then**
10: $F \leftarrow F \setminus f$
11: $O \leftarrow O \cup (F, F1(F), ROD(F))$

Phase 2: Exploration for fair feature sets

12: $B \leftarrow \emptyset$ ▷ the set of potentially biased features
13: **for** f *in* F **do**
14: **if** $ROD(F \setminus f) > ROD(F)$ **then**
15: $F \leftarrow F \setminus f$
16: $O \leftarrow O \cup (F, F1(F), ROD(F))$
17: $B \leftarrow B \cup f$
18: **for** b *in* B **do** ▷ floating
19: **if** $ROD(F \cup b) > ROD(F)$ **then**
20: $F \leftarrow F \cup b$
21: $O \leftarrow O \cup (F, F1(F), ROD(F))$
22: $O \leftarrow$ Pareto Front(O)

3.3 Fairness Guarantee

The feature subset obtained after the first phase of Algorithm 1 might contain biased features. Improving ROD in the second phase of Algorithm 1, removes features with a direct influence on the outcomes but not necessarily provides a fairness guarantee. To guarantee that there are no hidden correlations between the sensitive features and the target, in addition to the condition in Line 14, we check whether the given features have any violating correlation. We apply the SeqSel [20] algorithm proposed by which uses conditional independence tests [47] to remove features that leak sensitive information and are not blocked by the admissible feature set. Ghalotra et al. also prove that SeqSel ensures causal fairness even if some features in the subset capture information about sensitive features [20].

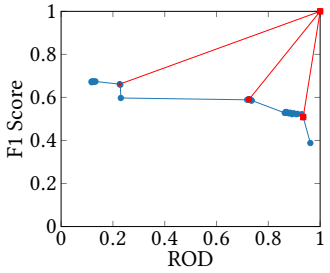

Figure 2: The Pareto front after feature set exploration. Selecting feature set with fairness weight of 0.25**,** 0.5**, and** 0.75**.**

3.4 Mixed-Initiative Feature Set Selection

Accuracy and fairness are two competing objectives. Therefore, all explored feature sets in the Pareto front are potential solutions for the ML application. Following the mixed-initiative approach [24] where our system and the user collaborate to achieve the user's goal, we propose three increasing levels of autonomy - supervised, semi-supervised, and unsupervised.

Supervised. The user has a clear understanding of which degree of fairness is required for the given ML application. Therefore, the user can pick the feature set from the Pareto front by choosing the most accurate feature set that still satisfies the fairness constraint.

Semi-Supervised. The user specifies the trade-off between accuracy and fairness by specifying weights for each objective. Based on the weights, FairExp automatically chooses the feature set that fits the user-specified trade-off best. FairExp chooses the feature set with the maximal weighted sum of both objectives:

$$z = \underset{x \in \mathcal{P}}{\text{argmax}}\ w_{\text{fair}} * ROD + (1 - w_{\text{fair}}) * \text{F1 score}, \tag{8}$$

where $\mathcal{P}$ contains all feature sets in the Pareto front and w_{fair} is the weight for the fairness objective. This approach resembles the "elbow method" [6] that was proposed to find the optimal number of clusters for unsupervised learning problems. For instance, Figure 2 illustrates how FairExp selects the feature set from the Pareto front for the Adult dataset based on the fairness weights 0.25, 0.5 (equal weights), and 0.75. In Section 4.5.2, we show the impact of considering different weighting schemes in the selected feature set.

Unsupervised. FairExp chooses the feature set automatically based on the assumption that both objectives are equal and the weights of both objectives are set to 0.5.

3.5 Scalability

Feature engineering consists of FC and FS. To scale feature engineering, one has to optimize both stages. The bottleneck of feature engineering is the exponentially growing number of constructed features. To reduce this exponential growth, multiple strategies have been proposed. First, our approach prunes all algebraically equivalent features, such as $a + b = b + a$. Second, one can always parallelize parallelizable parts of the algorithm. For instance, we

Table 2: Experimental datasets.

Dataset	Columns	Rows	Sensitive	Inadmissible
Traffic	17	1M	Race	-
Adult	10	48k	Gender	Marital status
COMPAS	5	7k	Race	-
German Credit	21	1k	Age	-

currently compute the features of the FC phase in parallel. Furthermore, we also parallelize the evaluation of one feature set. For instance, for some models, such as logistic regression, we can distribute the training to multiple cores (training parallelism). As we apply 5-fold cross-validation (CV), one can also compute each fold in parallel (CV parallelism). However, for CV parallelism the upper bound is the number of folds. Besides these trivial approaches to parallelize Algorithm 1, we analyzed the algorithm more closely and found two further parallelization opportunities. First, one approach is to parallelize the for loop of the floating procedure in Line 8. Second, we can also accelerate the forward pass of Algorithm 1 by leveraging additional compute resources. Inspired by branch prediction [44], we can speculatively evaluate additional potential feature sets to utilize all resources. As FC generates a large number of features and most of these features do not yield a gain in model accuracy, we can assume that the condition in Line 5 yields *False* in most cases and we can already evaluate the feature sets of the next k iterations where k is the number of cores. In the best case, we can evaluate the feature sets of k iterations in one cycle. In the worst case, we have no parallelism gain.

4 EXPERIMENTS

We performed several experiments to compare our approach against the SOA. We compare our approach in terms of prediction quality, fairness, and runtime. Also, we explore the impact of different feature evaluation methods, objective weighting, including features with longer DL, increasing number of instances, inadmissible features. Finally, we analyze our parallelization strategy.

4.1 Experimental Setup

4.1.1 Datasets. We conducted our experiments on three commonly studied datasets in algorithmic fairness literature [8, 13, 18, 42]. Moreover, we use the Traffic dataset [10] to analyze the runtime performance of our approach for an increasing number of instances. Table 2 summarizes the datasets along with the number of columns, rows, sensitive features, and inadmissible features if present.

The Adult dataset [14] contains information from the 1994 census in the United States. The prediction task is to determine whether a person makes over 50K dollars a year. Similar to prior work [42], we consider the sex (male, female) as the sensitive feature and marital status as an inadmissible feature because as pointed out in prior literature marital status is highly biased towards males. To compare with prior work [13, 42], we remove the features that contain other sensitive information such as race, native-country, and relationship.

The COMPAS dataset [30] contains records of offenders. The prediction task is to determine if an offender will relapse before trial. The dataset displays 1.5 times more African-Americans than Caucasians. Following prior work [42], we consider race as the sensitive feature, and the remaining features (age, number of prior convictions, and severity of charge degree) as admissible.

The German Credit dataset [14] contains financial information of credits granted to individuals in Germany. The prediction task is to classify whether a person will pay back the credit or not. We consider age as the sensitive feature (below average, above average).

The Montgomery County Traffic violations dataset [10], contains electronic traffic violations. The classification task is to predict whether the traffic violation corresponds to a warning or a citation. Moreover, the data shows that citations are issued in a significantly larger proportion to Hispanics when compared to other races.

4.1.2 Methods. We compare our method to the following methods.

Original. We leverage the original complete feature set.

Dropped. This baseline corresponds to the original feature set without the sensitive and inadmissible features, which are dropped.

CA [42]. We binned the numerical features as proposed by the authors.

Kamiran. We use the sample reweighting method proposed by Kamiran et al. [25]. **Calmon.** We use the same binning, feature subset, and distortion function as reported for the Adult and COMPAS datasets [13]. For the German credit dataset, we only consider numerical and ordinal features as it is required for this method. **Feldman.** We use the same binning and feature subset reported by Calmon et al. [13] for the adult dataset. For the German credit dataset, we only consider numerical and ordinal features as it is required for this method.

FC-NSGA-II. To evaluate a SOA multi-objective FS, we use the Non-dominated Sorting Genetic Algorithm (NSGA-II) [12] on the constructed features. Khan et al. [1] proposed to use feature set size and accuracy as objectives for selecting features using NSGAII. **FairExp**. For our system, we use by default a maximum feature DL of 4. Moreover, we assign equal weights (0.5) to fairness and accuracy objectives. For our feature evaluation component, we use logistic regression with 3-fold cross-validation on the training set to compute a validation score.

4.2 Evaluation Methodology

We measure the effectiveness of our methods by comparing the F1 score (as defined in Section 2.3) and the discrimination reduction using the *Ratio of Discrimination (ROD)* and all other fairness metrics reported by Salimi et al. [42] of the downstream classification task. F1 score is suitable in cases of class imbalance and ROD represents the strongest notion of fairness that is testable from data [42]. Following prior work [42], we normalize the distance to the best possible ROD score ($d = |1 - ROD|$) between 0 and 1, where 1 represents the best possible fairness. Accordingly, we also use the Grow-shrink algorithm [31] to learn the *Markov blanket* of the outcomes to compute the ROD score. We report the aggregated result of 5-fold cross-validation.

4.3 Effectiveness

Table 3 reports for each competing method the mean and standard deviation of achieved F1 and various fairness scores across different datasets. We bold the first and second-best results for each score. The results show that FairExp is competitive compared to all SOA baselines. Leveraging the original complete feature set yields high classification accuracy but poor fairness because bias is not considered. If we remove sensitive and inadmissible features from the

Table 3: Comparing FairExp to state-of-the-art and baselines with regard to downstream F1 score and fairness measures

Dataset	Method	F1	ROD	DP	TNB	TPB	CSP	CTNB	CTPB	Runtime
Adult	Capuchin	61 ± 1	82 ± 5	**84** ± 5	**84** ± 3	70 ± 13	**96** ± 3	**94** ± 5	**96** ± 3	69.89 ± 20
	Dropped	61 ± 1	**87** ± 4	75 ± 5	77 ± 6	**72** ± 5	**87** ± 1	**88** ± 2	**57** ± 6	9.43 ± 54
	FairExp	64 ± 3	81 ± 29	76 ± 32	82 ± 36	**86** ± 21	66 ± 30	66 ± 30	40 ± 11	$163k \pm 51k$
	FairExp-guaranteed	62 ± 2	**85** ± 4	74 ± 6	78 ± 6	66 ± 6	86 ± 1	87 ± 2	54 ± 5	$96k \pm 31k$
	FC-NSGAII	**70** ± 1	16 ± 8	12 ± 4	42 ± 6	14 ± 3	10 ± 4	11 ± 7	13 ± 3	$61k \pm 38$
	Kamiran-reweighting	63 ± 0	**85** ± 7	**96** ± 4	**88** ± 3	71 ± 13	60 ± 4	62 ± 4	47 ± 5	9.44 ± 18
	Original	**68** ± 1	22 ± 5	7 ± 5	4 ± 4	27 ± 8	4 ± 4	4 ± 4	10 ± 7	8.95 ± 15
COMPAS	Capuchin	64 ± 1	**98** ± 2	42 ± 9	**62** ± 14	42 ± 13	**95** ± 10	**96** ± 10	**95** ± 10	1.82 ± 17
	Dropped	**66** ± 3	89 ± 2	38 ± 15	61 ± 17	36 ± 15	**89** ± 4	**85** ± 7	85 ± 5	1.21 ± 11
	FairExp	**67** ± 2	**91** ± 4	**46** ± 11	58 ± 16	**52** ± 17	88 ± 6	84 ± 7	**88** ± 7	95.97 ± 22.85
	FairExp-guaranteed	**66** ± 3	78 ± 21	33 ± 20	58 ± 18	30 ± 23	74 ± 24	71 ± 21	72 ± 21	149.24 ± 9.04
	FC-NSGAII	65 ± 3	68 ± 18	26 ± 22	38 ± 22	33 ± 17	54 ± 35	49 ± 37	62 ± 22	$1k \pm 18$
	Kamiran-reweighting	65 ± 2	4 ± 5	**85** ± 13	**90** ± 10	**87** ± 11	33 ± 10	26 ± 16	36 ± 4	1.27 ± 12
	Original	**66** ± 3	52 ± 8	19 ± 16	45 ± 19	15 ± 17	56 ± 12	62 ± 16	54 ± 9	1.13 ± 10
German credit	Capuchin	**83** ± 3	80 ± 13	49 ± 31	66 ± 30	59 ± 31	0 ± 0	0 ± 0	0 ± 0	13.61 ± 3.08
	Dropped	78 ± 4	**83** ± 18	59 ± 10	**76** ± 31	72 ± 30	0 ± 0	0 ± 0	0 ± 0	1.24 ± 15
	FairExp	**81** ± 3	**1.00** ± 0	**91** ± 7	**78** ± 20	**94** ± 6	0 ± 0	0 ± 0	0 ± 0	$1k \pm 391.72$
	FairExp-guaranteed	78 ± 3	67 ± 29	38 ± 27	64 ± 29	49 ± 30	0 ± 0	0 ± 0	0 ± 0	$4k \pm 2665$
	FC-NSGAII	79 ± 5	57 ± 34	38 ± 26	45 ± 30	66 ± 29	0 ± 0	0 ± 0	0 ± 0	$10k \pm 167$
	Kamiran-reweighting	**83** ± 3	80 ± 27	**87** ± 10	60 ± 24	**80** ± 19	0 ± 0	0 ± 0	0 ± 0	1.33 ± 9
	Original	78 ± 3	70 ± 12	32 ± 19	71 ± 20	50 ± 35	0 ± 0	0 ± 0	0 ± 0	1.31 ± 14

original feature set, we significantly improve fairness but also cause a significant drop in classification accuracy as can be seen for the baseline **Dropped**. In addition to removing biased features, the SOA strategy CA further reduces bias in the data horizontally by adding and removing instances. However, the disadvantage of CA is that its horizontal bias repair significantly changes the underlying distribution of the training set which leads to it overfitting for fairness and less generalization for the unseen test set. Furthermore, CA requires discretization of numerical features that might cause an additional loss in information depending on the degree of binning.

The multi-objective FS strategy NSGA-II achieves high classification accuracy and low fairness. We found that NSGA-II often gets stuck in local optima due to the large number of features. Therefore, it tends to prefer one objective over the other. Another reason why FairExp outperforms NSGA-II is that NSGA-II does not consider the DL of constructed features and, therefore, might overfit.

To better showcase the approach of FairExp, consider the following features that were generated and selected. For instance, the following features are constructed and selected for the Adult dataset instead of using the inadmissible feature *marital-status*: *mean(marital-status = "Divorced") GroupBy capital-loss, mean(marital-status = "Widowed") GroupBy capital-loss*, and *std(workclass = "Local-gov") GroupBy marital-status*. These features show an equal distribution with regard to the different groups inside the sensitive attribute *Sex* and add predictive power for the classification model. By considering only specific values of *marital-status* and combining it with other features, we can extract unbiased information that leads to higher accuracy. On German Credit, the sensitive feature *age* was combined with the feature *existing credits* to extract additional information: *mean(age = "above average") GroupBy existing credits*.

For COMPAS, our method did not select features that captured data from the sensitive feature *race*. Instead, the following features were considered: *(max(age = "Less than 25") GroupyBy priors count), age = "Greater than 45" * priors count*.

To compare our system to Calmon et al. [13] and Feldman et al. [18], which are designed for numerical ordinal features, we removed all non-numerical and non-ordinal features.

Table 4 shows that FairExp performs poorly for this case because the number of features is very low, e.g. only *age* and *education* for Adult. Therefore, FairExp cannot uncover any meaningful relationships between features using FC. All in all, the experiment shows that FairExp is a promising alternative to SOA bias reduction approaches that follows a vertical approach.

4.4 Runtime

Table 3 reports the runtime in seconds for all described methods. We define runtime as the total elapsed time that a method takes to complete the pre-processing component (FC, feature exploration, and feature set selection) of a system, and to train the model with the selected representation. All strategies that leverage FC, such as FC-NSGA-II and FairExp, require multiple orders of magnitude more time than strategies that do not. There is a trade-off between runtime and effectiveness. However, we accelerate our approach by parallelizing the algorithm as described in Sec. 3.5.

4.5 Micro-Benchmark Results

We present an analysis of the influence of different FC parameters and representation selection components on accuracy, fairness, and runtime. Moreover, we conduct an experiment to show which FC transformations are most likely to remove bias from input features.

4.5.1 Feature Construction. Theoretically, we can apply transformations arbitrarily many times recursively. To reduce the runtime of FC, we have to limit the number of constructed features. Table 5 shows how choosing different degrees of DL, as defined in Section 3.1, affects accuracy and fairness. With increasing DL, the number of constructed features grows exponentially because new features can be combined with old and new features to create even more features. This exponential growth also affects the runtime significantly. However, with increasing DL, the classification accuracy improves because we can extract more and more information from the original features. At the same time, fairness slightly decreases but maintains a high level. Therefore, we construct features until a

Table 4: Comparison on datasets with numerical and ordinal features only

Dataset	Method	F1	ROD	DP	TNB	TPB	CSP	CTNB	CTPB	Runtime
Adult	Calmon	**49** ± 01	70 ± 02	**92** ± 03	79 ± 07	**86** ± 13	1.00 ± 00	94 ± 03	71 ± 07	725.21 ± 19.57
	Capuchin	41 ± 01	**90** ± 12	**92** ± 06	**95** ± 07	**87** ± 10	**1.00** ± 00	**1.00** ± 01	**99** ± 01	5.38 ± 1.70
	FairExp	**57** ± 01	32 ± 19	25 ± 23	47 ± 40	57 ± 32	25 ± 22	29 ± 24	29 ± 19	744.90 ± 295.40
	Feldman	45 ± 01	**1.00** ± 00	**81** ± 03	**86** ± 06	**87** ± 11	**1.00** ± 00	**1.00** ± 00	**90** ± 18	3.58 ± 07
COMPAS	Calmon	**66** ± 01	38 ± 34	55 ± 31	54 ± 32	**49** ± 27	**80** ± 45	**80** ± 45	**80** ± 45	116 ± 1.14
	Capuchin	64 ± 01	**78** ± 10	53 ± 15	**64** ± 24	35 ± 22	**92** ± 17	**92** ± 17	**92** ± 17	1.82 ± 17
	FairExp	**68** ± 02	48 ± 29	**56** ± 13	59 ± 17	44 ± 23	78 ± 17	67 ± 18	77 ± 12	102.83 ± 17.32
	Feldman	65 ± 01	**1.00** ± 01	**91** ± 06	**85** ± 11	**82** ± 13	**85** ± 13	80 ± 09	78 ± 12	2.60 ± 07
German credit	Calmon	**82** ± 02	61 ± 50	**89** ± 09	**85** ± 13	**92** ± 09	96 ± 07	92 ± 11	66 ± 24	1536 ± 1.93
	Capuchin	**82** ± 04	39 ± 28	39 ± 30	40 ± 35	59 ± 42	53 ± 36	41 ± 31	51 ± 35	8.17 ± 1.47
	FairExp	76 ± 06	**97** ± 02	**86** ± 15	**85** ± 13	84 ± 14	**98** ± 04	**93** ± 07	**88** ± 08	604.68 ± 28.10
	Feldman	**82** ± 02	**90** ± 09	**86** ± 07	74 ± 21	**90** ± 04	**99** ± 03	**1.00** ± 00	**79** ± 12	3.28 ± 1.44

Table 5: Impact of DL on accuracy, fairness, and runtime.

Dataset	DL	Const. features	F1	ROD	Runtime
Adult	1	0	44 ± 1	**98** ± 1	29 ± 1
	2	432	56 ± 1	95 ± 2	$1.2k \pm 178$
	3	647	60 ± 1	93 ± 2	$5.9k \pm 931$
	4	3.4k	**62** ± 3	91 ± 3	$50.5k \pm 4.2k$
COMPAS	1	0	46 ± 3	**92** ± 4	6 ± 1
	2	21	46 ± 3	**93** ± 4	18 ± 3
	3	61	46 ± 3	**93** ± 4	76 ± 15
	4	199	**55** ± 2	78 ± 10	$5.6k \pm 2.2k$
German Credit	1	0	55 ± 31	96 ± 3	15 ± 2
	2	1k	77 ± 5	**97** ± 4	190 ± 11
	3	1.8k	**83** ± 3	91 ± 5	$7.7k \pm 109$
	4	7.5k	80 ± 2	89 ± 8	$8.5k \pm 526$

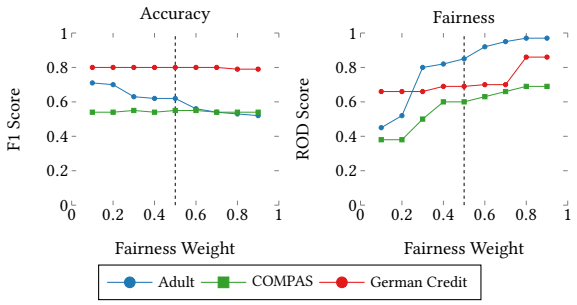

Figure 3: Impact of weighting on fairness and accuracy.

DL of 4 because, this way, we achieve high accuracy while maintaining competitive fairness. Constructing features with DL higher than 4 is not feasible because the runtime increases exponentially. To generate features of DL 5, more than a week of computation time can be expected for each dataset.

4.5.2 Objective Weighting . Depending on the use case, fairness might be more important than accuracy and vice-versa. Therefore, in our system, users can specify this trade-off. In this experiment, we evaluate the impact of different weighting schemes. Figure 3 shows the impact of different weighting schemes for our strategy FairExp on the evaluated datasets.

As observed in Figure 3, the fairness score improves as the weight for the fairness objective increases. However, for German Credit and COMPAS, the accuracy stays relatively stable even with increasing weight for fairness. The reason is that, for these two datasets, there are very few important and unbiased features. Therefore, removing additional features does not significantly affect the classification accuracy. The opposite is true for Adult where many features are required to achieve high accuracy. Therefore, we see a decline in accuracy when increasing the fairness weight because more and more features will be removed.

Table 6: Comparison of different classification models.

	Adult		COMPAS		German Credit	
Model	F1	ROD	F1	ROD	F1	ROD
LR	**62** ± 3	**91** ± 3	**55** ± 2	78 ± 10	80 ± 2	**89** ± 8
SVM	55 ± 9	70 ± 21	37 ± 7	**79** ± 8	**82** ± 2	85 ± 11
GB	55 ± 9	64 ± 20	41 ± 5	41 ± 10	80 ± 2	58 ± 20

Table 7: Scaling the number of rows for the Traffic dataset.

Number Rows	1k	10k	100k	1M
Runtime (seconds)	168 ± 12	463 ± 70	2984 ± 1098	20670 ± 2293

Table 8: Scaling the number of inadmissible features.

# Inadmissible Features	1	10	25	50
Runtime (s)	365 ± 41	376 ± 61	393 ± 36	340 ± 11

4.5.3 Model Selection. We analyze the influence of the classification model on the optimization problem. We experimented with 3 different models: gradient boosting (GB), logistic regression (LR), and support vector machines (SVM). Table 6 shows the accuracy and fairness for three models across the three benchmark datasets. In general, the scores vary across models. For instance, GB achieves a lower ROD score compared to the other two models. The reason is that GB on its own also selects features in its training phase to optimize accuracy. This focus on accuracy affects the fairness score negatively. Generally, it is well-known that different ML models fit different datasets differently well, which motivates the field of algorithm selection [19].

4.5.4 Scalability. In Section 4.5.1, we already analyzed the scalability across the number of constructed features. We now evaluate the scalability of our approach with respect to the number of instances and the number of inadmissible features. Table 7 shows that our approach scales linearly for an increasing number of instances.

Second, we experimented on a randomly generated dataset with 2000 rows and 198 boolean features, one target, and one sensitive feature. Then, we randomly pick an increasing number of inadmissible features. Table 8 contains the results of this experiment. It shows that the number of inadmissible features only marginally affects runtime because we have to conduct the same number of conditional independence tests. Computationally, we treat sensitive features and inadmissible features the same. Therefore, increasing

Table 9: Impact of Parallelization Strategies

Training Parallelism	CV Parallelism	Speculative Parallelism	Runtime (s) German Credit	Traffic
1	1	1	6359	391
27	1	1	6703	389
1	5	1	7082	339
1	1	27	1228	175

Table 10: Different ranking strategies on Adult

Ranking Strategies	F1	ROD
MDL	0.61	-0.707
reverse MDL	0.62	-0.716
Random	0.61	-1.456

Table 11: Capuchin and FairExp

Method	Adult F1	Adult ROD	COMPAS F1	COMPAS ROD
Original	**68** ± 1	17 ± 14	**66** ± 1	7 ± 10
CA	**62** ± 1	77 ± 3	64 ± 1	**94** ± 3
CA-FairExp	53 ± 5	**92** ± 6	65 ± 2	79 ± 15
CA-FairExp-Dropped	56 ± 1	**93** ± 5	**66** ± 2	**83** ± 20
FairExp	60 ± 2	**92** ± 5	**67** ± 2	80 ± 7

the number of sensitive features behaves the same way as increasing the number of inadmissible features.

4.5.5 Parallelization. Table 9 compares training, CV, and speculative parallelism. For German Credit, training and CV parallelism introduce too much overhead and therefore require even more runtime than running without parallelism. Only the parallelization approach that speculatively evaluates feature sets is 5 times faster than without parallelization by leveraging 27 CPU cores. However, when running on a larger dataset, e.g. Traffic with 10k examples, training and CV parallelization improve the runtime compared to no parallelization.

4.5.6 Ranking Strategies. In the forward pass of Algorithm 1, we rank the features first based on their DL. Table 10 compares the ranking strategies MDL, reverse MDL, and random. All of them achieve similar results. Therefore, we chose MDL as it prefers simple features over complex ones.

4.5.7 Combining horizontal and vertical bias reduction. We first bin the data, repair it using CA. Then, we apply FairExp on the repaired data (use the original data points as opposed to the binned ones for those that are retained by the repair algorithm). Table 11 shows the result of this experiment. FairExp improves the fairness because CA guarantees fairness on training but overfits. FairExp achieves higher fairness by optimizing for the cross-validation score. The results are similar when we additionally drop sensitive and inadmissible attributes (CA-FairExp-Dropped).

5 RELATED WORK

Our work is strongly related to research on algorithmic bias reduction and feature engineering.

Algorithmic Bias Reduction. Recent work for reducing bias includes pre-processing [2, 13, 18, 50], in-processing[29, 33, 37, 40, 43] and post-processing[22, 48] strategies. Pre-processing methods aim to remove the bias in the training data either by modifying the probability distributions [13, 18, 50], selecting a subset of features [20], or defining different feature weight strategies [2]. In-processing methods impose constraints on the algorithm's loss function [26] and optimize fairness by tuning the classifier's hyperparameters [37]. Post-processing methods aim to remove the bias via some transformation of the predictions [9, 22]. Our algorithm follows the pre-processing approach but does not only optimize for accuracy but also fairness.

Furthermore, bias reduction algorithms either follow an associational or a causal approach to measure fairness. Associational approaches check for inequalities in the algorithm's outcomes between groups of the sensitive feature, while causal strategies aim to identify, quantify, and remove the influence of the sensitive feature on the outcomes. Approaches based on causality frameworks [13, 42] have shown to avoid paradoxical conclusions and provide more principled reasoning to understand the influence of the sensitive features on an algorithm's outcome [42]. Our solution follows the spirit of the described causal algorithmic fairness frameworks but none of them employs FS and FC simultaneously to address the problem with a multi-objective approach.

Feature Engineering. Feature engineering has been broadly used to improve prediction performance, provide faster and more cost-effective predictors, and provide a better understanding of the underlying process generating the data [16, 17, 21, 27, 34]. Feature engineering approaches have been lately proposed for algorithmic fairness [35].

6 CONCLUSION

We proposed a novel pre-processing bias reduction algorithm based on automatic FC and selection that does not only optimize for accuracy but also fairness. Our system generates a large number of new features from the original feature set by applying human-understandable transformations. Our experiments show that our system achieves competitive results compared to SOA strategies. Unlike SOA, our method does not require the deletion of tuples and, therefore, it generalizes better to unseen data. Further research is needed to efficiently restrain the large number of feature candidates and explanation of constructed features.

Acknowledgments. The contribution of Felix Neutatz was funded by the German Ministry for Education and Research as BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref. 01IS18037A).

REFERENCES

[1] 2015. Multi-Objective Feature Subset Selection using Non-dominated Sorting Genetic Algorithm. *Journal of Applied Research and Technology* 13, 1 (2015), 145 – 159.
[2] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing Fair Ranking Schemes. In *SIGMOD*. 1259–1276.
[3] Solon Barocas et al. 2019. *Fairness and Machine Learning*. fairmlbook.org.
[4] Solon Barocas and Andrew D. Selbst. 2016. Big Data's Disparate Impact. *California Law Review* 104 (2016), 671.
[5] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.* 21, 2 (2010), 277–292.
[6] Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* 3, 1 (1974), 1–27.
[7] L. Elisa Celis and Vijay Keswani. 2019. Improved Adversarial Learning for Fair Classification. *CoRR* abs/1901.10443 (2019).
[8] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.
[9] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic Decision Making and the Cost of Fairness. In *SIGKDD*. 797–806.

[10] Montgomery County. 2020. Traffic Violations. data retrieved from Data.gov, https://data.montgomerycountymd.gov/api/views/4mse-ku6q.

[11] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated Experiments on Ad Privacy Settings. *Proc. Priv. Enhancing Technol.* 2015, 1 (2015), 92–112.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[13] Flávio du Pin Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. 2017. Optimized Pre-Processing for Discrimination Prevention. In *NeurIPS*. 3992–4001.

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

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *ITCS*. 214–226.

[16] Mahdi Esmailoghli and Ziawasch Abedjan. 2020. CAFE: Constraint-Aware Feature Extraction from Large Databases. In *CIDR*.

[17] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2021. COCOA: COrrelation COefficient-Aware Data Augmentation. In *EDBT*. 331–336.

[18] Michael Feldman et al. 2015. Certifying and Removing Disparate Impact. In *SIGKDD*. 259–268.

[19] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *NeurIPS*. 2962–2970.

[20] Sainyam Galhotra, Karthikeyan Shanmugam, Prasanna Sattigeri, and Kush R. Varshney. 2020. Fair Data Integration. *CoRR* abs/2006.06053 (2020).

[21] Isabelle Guyon and André Elisseeff. 2003. An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* 3 (2003), 1157–1182.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NeurIPS*. 3315–3323.

[23] Franziska Horn, Robert Pack, and Michael Rieger. 2019. The autofeat Python Library for Automated Feature Engineering and Selection. In *ECML/PKDD*. 111–120.

[24] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *CHI*. 159–166.

[25] Faisal Kamiran and Toon Calders. 2011. Data Pre-Processing Techniques for Classification without Discrimination. *Knowledge and Information Systems* 33 (10 2011). https://doi.org/10.1007/s10115-011-0463-8

[26] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-Aware Classifier with Prejudice Remover Regularizer. In *ECML/PKDD (Lecture Notes in Computer Science, Vol. 7524)*. 35–50.

[27] Gilad Katz, Eui Chul Richard Shin, and Dawn Song. 2016. ExploreKit: Automatic Feature Generation and Selection. In *ICDM*. 979–984.

[28] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for Occupations. In *CHI*. 3819–3828.

[29] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. Avoiding Discrimination through Causal Reasoning. In *NeurIPS*. 656–666.

[30] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. How We Analyzed the COMPAS Recidivism Algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm [Online; posted 23-May-2016].

[31] Dimitris Margaritis. 2012. Learning Bayesian Network Model Structure From Data. (03 2012).

[32] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondrej Certík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason Keith Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Stepán Roucka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony M. Scopatz. 2017. SymPy: symbolic computing in Python. *PeerJ Comput. Sci.* 3 (2017), e103.

[33] Razieh Nabi and Ilya Shpitser. 2018. Fair Inference on Outcomes. In *AAAI*. 1931–1940.

[34] Felix Neutatz, Felix Biessmann, and Ziawasch Abedjan. 2021. Enforcing Constraints for Machine Learning Systems via Declarative Feature Selection: An Experimental Study. In *SIGMOD*.

[35] Alejandro Noriega-Campero, Michiel A. Bakker, Bernardo Garcia-Bulle, and Alex 'Sandy' Pentland. 2019. Active Fairness in Algorithmic Decision Making. In *AIES*. 77–83.

[36] Judea Pearl. 2014. Comment: understanding Simpson's paradox. *The American Statistician* 68, 1 (2014), 8–13.

[37] Valerio Perrone, Michele Donini, Krishnaram Kenthapadi, and Cédric Archambeau. 2020. Fair Bayesian Optimization. *CoRR* abs/2006.05109 (2020).

[38] Pavel Pudil, Jana Novovicová, and Josef Kittler. 1994. Floating search methods in feature selection. *Pattern Recognit. Lett.* 15, 10 (1994), 1119–1125.

[39] Jorma Rissanen. 1978. Modeling by shortest data description. *Autom.* 14, 5 (1978), 465–471.

[40] Chris Russell, Matt J. Kusner, Joshua R. Loftus, and Ricardo Silva. 2017. When Worlds Collide: Integrating Different Counterfactual Assumptions in Fairness. In *NeurIPS*. 6414–6423.

[41] Babak Salimi, Johannes Gehrke, and Dan Suciu. 2018. Bias in OLAP Queries: Detection, Explanation, and Removal. In *SIGMOD*. 1021–1035.

[42] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional Fairness: Causal Database Repair for Algorithmic Fairness. In *SIGMOD*. 793–810.

[43] Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. 2020. FairPrep: Promoting Data to a First-Class Citizen in Studies on Fairness-Enhancing Interventions. In *EDBT*. 395–398.

[44] James E. Smith. 1998. Retrospective: A Study of Branch Prediction Strategies. In *25 Years of the International Symposia on Computer Architecture*. 22–23.

[45] Julia Stoyanovich, Bill Howe, and H. V. Jagadish. 2020. Responsible Data Management. *PVLDB* 13, 12 (2020), 3474–3488.

[46] Julia Stoyanovich, Bill Howe, H. V. Jagadish, and Gerome Miklau. 2018. Panel: A Debate on Data and Algorithmic Ethics. *PVLDB* 11, 12 (2018), 2165–2167.

[47] Eric V. Strobl, Kun Zhang, and Shyam Visweswaran. 2019. Approximate Kernel-Based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. *Journal of Causal Inference* 7, 1 (2019). https://doi.org/doi:10.1515/jci-2018-0017

[48] Blake E. Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. 2017. Learning Non-Discriminatory Predictors. In *COLT (Proceedings of Machine Learning Research, Vol. 65)*. 1920–1953.

[49] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *Big Data*. 570–575.

[50] Lu Zhang, Yongkai Wu, and Xintao Wu. 2017. A Causal Framework for Discovering and Removing Direct and Indirect Discrimination. In *IJCAI*. 3929–3935.