A Brief Overview of the State of the Art in Fairness

Andre Lustosa
alustos@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

Algorithmic fairness is a current and relevant research field in software engineering. Systems created by developers using machine learning for decision making may present different biases that can translate to discrimination based on characteristics such as age, sex, race, etc. Given the novelty of the field, most of current work baselines itself against previous traditional machine learning techniques. Research on fairness is then currently lacking a study on how the existing methods fare against each other.

In this study we reproduce 5 previous highly cited and reused works in the literature, and apply them to 7 different datasets with the goal of understanding the current state of the fairness arena, and to find an appropriate baseline for future work. The selection of methods was based in three criteria, citation count, generality, and proof of reuse.

The results of this study show that there are two algorithms out of the 5 that stand out in all tests. As such they are recommended as baselines to which any future work in this arena should be compared to.

KEYWORDS

datasets, fairness, reproduction, software engineering

ACM Reference Format:
Andre Lustosa. 2021. A Brief Overview of the State of the Art in Fairness. In *Proceedings of Raleigh '21: Sinless Software Engineering Final Report (Raleigh '21).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

It has become more and more common for software to make autonomous decisions in cases such as credit approval, hiring processes, criminal sentencing and many other situations. The ones responsible for developing this software have an ethical responsibility to guarantee that their software is free of biases, given that that kind of software directly impact human lives.

Unfortunately, there are many examples where the opposite is true, software showing biased behavior based on some protected attribute. As pointed out by Chakraborty et al. [11]:

- Amazon had to scrap an automated recruiting tool as it was found to be biased against women. [13].
- A widely used face recognition software was found to be biased against dark-skinned women. [1]
- Google Translate, shows gender bias when translating sentences between gendered and non gendered languages (e.g: "She is an engineer, He is a nurse" to Turkish and then back becomes "He is an engineer, She is a nurse"). [2]

Given how this is a new and emerging field on the concept of algorithmic bias, or bias existing within software products, most of the work in the literature has been developed comparing their findings against traditional ML techniques. So not much work has been done in comparing existing state-of-the-art solutions to fairness and their overall performance on the most commonly used fairness datasets. These approaches can vary in ways where we can classify them among three different categories.

- **Pre-processing** - When the technique is completely model agnostic and only modifies the data. [10, 11, 15]
- **In-processing** - When the technique is not usually model agnostic, being something applied directly to the learner. [7, 11, 17]
- **Post-processing** - When the technique is applied to the result obtained by the learner during prediction

Achieving fairness is a difficult task, because in most cases, there is a compromise on performance while achieving better results on fairness metrics. As such it is valuable to figure out which of the current existing fairness techniques and algorithms perform the best. The focus of this paper was to perform a brief overview on a few algorithms by replicating their results on the most commonly used fairness datasets, comparing their results and finding out if there are any particularities that differentiate techniques from the above mentioned categories.

Overall this paper makes the following contributions:

- The largest reproduction of fairness work to date.
- A detailed comparison on the different results achievable by different algorithm categories (excluding Post-processing).
- A summary showing the current state-of-the-art algorithms which should be used as baseline for future work.

A caveat that has to be made before we proceed, is that, in this reproduction work, we remained model agnostic. Using a Random Forest whenever allowed. We have also not delved into Post-processing algorithms, and the reasoning for that is that in general Post-processing algorithms require a large quantity of domain-specific knowledge, which makes these algorithms not as easily generalized for different problems (datasets and tasks).

2 BACKGROUND

Fairness in ML models, while a well-explored research topic in the ML community, it has only recently gained traction within the software engineering community. ICSE 2018 and ASE 2019

Raleigh '21, Dec 08, 2021, Raleigh, NC

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

conducted workshops for software fairness [3, 4]. Big software industries have also started taking this issue seriously. For example, IBM created a public GitHub repository, AI Fairness 360 [5](AIF360), where the most popular fairness metrics and algorithms can be found. Microsoft has also created a dedicated research group for fairness, accountability, transparency and ethics in AI [6].

From the above sources we have selected a few works that we consider were the most influential, and also the most data and model agnostic algorithms available. Given that in order to compare these algorithms, they need to be a generic fairness solution. As said before, we have not selected any algorithm from the Post-processing category due to it's limitations when reapplying those techniques to multiple datasets.

First we have selected the most popular and most cited works that made it to the AIF360 repository. This yielded us four algorithms to use in our comparison study. We have also selected the Distinguished Paper award winner work by Chakraborty et al [11]. We also note that Chakraborty et al's algorithm does not quite classify solely as a Pre-processing algorithm, albeit being the core part of the work, some In-processing happens through the means of situation testing which will be explained further. This situation testing ends up classify it as a Mixed-processing algorithm.

The reasoning for these choices is as follows. We have started our literature review by reviewing Chakraborty et al's work on Fair-SMOTE, where they state that, none of the popular bias mitigation algorithms were good baselines for their work due to previous evidence of degradation on the learner performance. However as shown on Chakraborty et al's work, their Fair-SMOTE algorithm also shown some degradation in performance when compared to default learners without bias mitigation. Hence this brings up the question.

> *How do these popular bias mitigation algorithms compare to Fair-SMOTE?*

So we have accordingly selected from AIF360 the most cited and relevant algorithms, which brought us down to a sample of four algorithms. Two of them being Pre-processing algorithms and two of them being In-Processing algorithms.

This selection took into account:

- Number of citations and venue of the original papers;
- Generality of the algorithm in regards to models and datasets;
- Number of publicly available reuses of the algorithm on the AIF360 repository through forked repositories or reproduction samples.

In this section then we will discuss the general motivation, theory and reasoning behind each of the four selected techniques.

2.1 Optimized pre-processing

This algorithm formulated by Calmon et al. [10] can be summarized as a statistical optimization problem. The idea is to apply a transformation function to an original dataset (X_i, Y_i), where X are the independent variables and Y is the target class, in order to transform it into a new dataset of tuples $(D_i, \hat{X}_i, \hat{Y}_i)$, which is then used to train the model. $\hat{X}_i$ being the transformed X_i without the D_i protected attribute, and $\hat{Y}_i$ is the transformed target class.

The goal of this transformation is to reduce disparities in the predicted outcome (Y_i) solely based on the D_i variable. Which theoretically will produce a less biased algorithm. It is important to note that this algorithm will provide the D_i variable for training the model, given that the transformation is already equalizing the predicted outcome for the unpriviledged and priviledged classes. As such this is an Pre-processing algorithm, for it does not interfere in any way with the model, and solely transforms the dataset prior to training as shown in Figure 1.

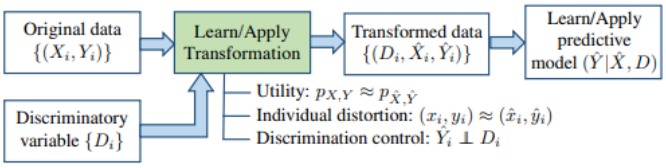

Figure 1: The proposed pipeline for predictive learning with discrimination prevention. Learn mode applies with training data and apply mode with novel test data. Note that test data also requires transformation before predictions can be obtained.

2.2 Reweighing

Algorithm 1 Reweighing

Require: (D, S, Class) ▷ *Dataset, Protected Attribute, Target Class*
Ensure: Classifier learned on reweighed D
  for $s \in \{b, w\}$ **do**
    for $c \in \{-, +\}$ **do**
      Let $W(s, c) := \frac{|\{X \in D | X(S)=s\}| \times |\{X \in D | X(Class)=c\}|}{|D| \times |\{X \in D | X(Class=c) \wedge X(S)=s}$
    end for
  end for
  $D_W := \{\}$
  for $X \in D$ **do**
    Add$(X, W(X(S), X(Class)))$ to D_W
  end for
  Train a classifier C on training set D_W, taking onto account the weights
  return Classifier C

The Reweighing algorithm, formulated by Kamiran et al. [15], is one of three algorithms proposed in this paper. All of the three proposed algorithms are techniques to alter the data in some form to allow for a less biased dataset. Particularly, the Reweighing algorithm can be summarized as a weight calculation for each sample in the dataset, in order to reduce the discrimination while maintaining overall positive class probablility, or maintaining accuracy on the model. As such, this algorithm calculates biases in the dataset by observing the difference in positive class probability for a protected attribute. If there is a bias, (e.g.: the probability is not equal between the two possible protected attribute values), lower weights will be assigned to samples that have been deprived or favored.

In this way their algorithm assigns a weight to every data point according to the step described in the previous paragraph, these weights are then used as biases during training. Most of available models are capable of accepting weights as a secondary input, but

for those that don't in his work Kamiran et al. [15] also proposes a secondary, Sampling, algorithm analogous to Reweighing.

2.3 Expontentiated Gradient Reduction

On their algorithm Agarwal et al. [7], propose a newly and altered gradient descent algorithm that is applied on a learner as it's loss function. The difference is that this algorithm is capable of doing cost-sensitive classification, which means they are capable of specifying different costs for different training examples. This abstraction is essential for the incorporation of fairness constraints. In particular their approach is equivalent to a weighted classification problem. The question then is how to reduce the dataset into such form during training. And for that they propose the Exponentiated Gradient Descent by converting their problem as a saddle point problem.

Converting their problem to a saddle point problem allows them to apply a gradient descent algorithm to minimize loss, and in this case loss is a complex function comprising of multiple goals (Fairness and Performance).

2.4 Adversarial Debiasing

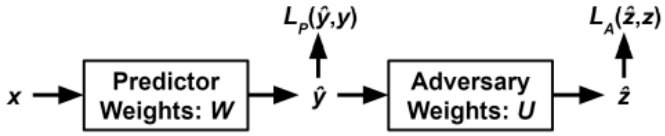

Figure 2: The architecture of the adversarial network.

In their work Zhang et al. [17], propose an adversarial learning architecture as their approach for solving the fairness problem. They begin with a model, which they call the *predictor*, the goal of the predictor is to predict Y (the target class) given X (the dependent variables). As shown in Figure 2, they assume a model trained by modifying weights W to minimize some loss $L_p(\hat{(}y), y)$, using a gradient-based method such as stochastic gradient descent.

The output layer of this *predictor* is then used as an input for a secondary network called *adversary*, whose goal is to predict Z (the protected attribute). In this architecture, when the *adversary* successfully predicts Z, the predictors loss function suffers, making the *predictor* adapt to not allow for this scenario. The goal then is to train a *predictor* that does not allow the *adversary* to correctly predict for Z, which in turn is by definition a non-biased prediction.

2.5 Fair-SMOTE

For their solution Chakraborty et al. [11], take a slighty different approach than the previously cited related work. Their approach stems from the SMOTE algorithm [12] hence the name Fair-SMOTE. As SMOTE, Fair-SMOTE focus on rebalancing the dataset, however instead of only worrying about class inbalance, Fair-SMOTE does a Pre-processing step on the dataset where the classes are balanced together with the proportion of classes between priviledged and unpriviledged sets of the protected attribute at hand.

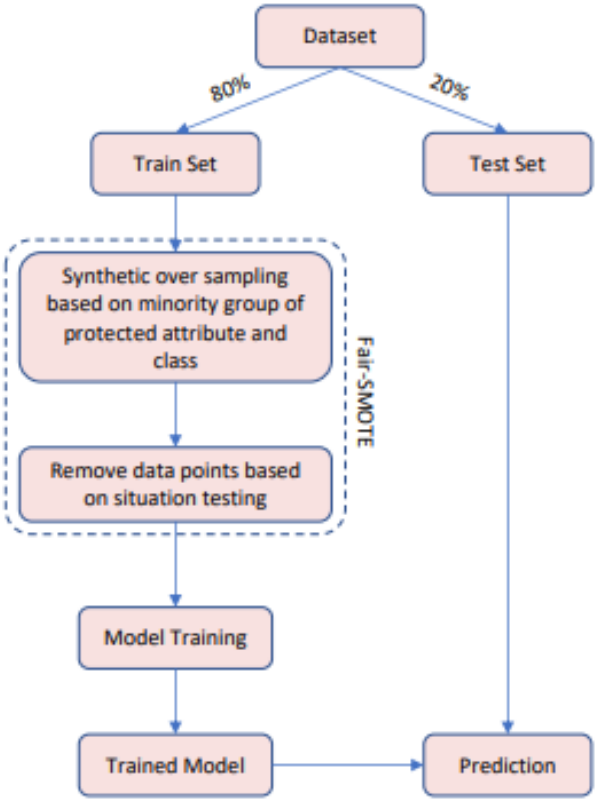

Figure 3: Diagram of Fair-SMOTE.

This is done through a synthetic generation of new data points for all classes and protected attributes except those that are unbalanced in excess. These datapoints are created close to it's parent point, ensuring that the new data belongs in the same distribution. The flowchart of this algorithm can be seen on Figure 3.

After the synthetic over sampling is completed, through the use of simple logistic regression models, Fair-SMOTE performs situation testing on data points, eliminating those that it deems biased. The logic for elimination stems from whether the class label will flip, if we change the protected attribute. This next step, interferes with a logistic-regression algorithm, but does not interferes with the final model to be trained. Hence why this model is not completely a Pre-processing model or an In-processing model.

It is important to note that this transformation is only applied to training data, and not the test data as seen on Figure 3

3 METHODOLOGY

3.1 Datasets

Our dataset selection is based on commonly used datasets in the fairness literature. Table 1 shows the selected datasets and their characteristics. The datasets were pre-treated (one-hot encoding and normalization of features) and are made available on the AIF360

Dataset	#Rows	#Features	Protected Attribute	Description
Adult Census Income	48,842	14	Sex, Race	Individual information from 1994 U.S. census. Goal is predicting income >$50,000.
Compas	7,214	28	Sex, Race	Contains criminal history of defendants. Goal predicting reoffending in future
Bank Marketing	45,211	16	Age	Contains marketing data of a Portuguese bank. Goal predicting term deposit.
Default Credit	30,000	23	Sex	Customer information for people from Taiwan. Goal is predicting default payment.
German Credit	1,000	20	Sex	Personal information about individuals & predicts good or bad credit.
Heart Health	297	14	Age	Patient information from Cleveland DB. Goal is predicting heart disease.
MEPS-15,16	35,428	1,831	Race	Surveys of families, individuals, medical providers, employers. Target is "Utilization"

Table 1: Datasets used in this study. Taken from the AIF360 Github Repository [5]

Fairness Metric	Ideal Value
$AOD = [(FPR_U - FPR_P) + (TPR_U - TPR_P)] * 0.5$	0
$EOD = [TPR_U - TPR_P]$	0
$SPD = P[\hat{Y} = 1 \mid PA = 0] - P[\hat{Y} = 1 \mid PA = 1]$	0
$DI = P[\hat{Y} = 1 \mid PA = 0] / P[\hat{Y} = 1 \mid PA = 1]$	1

Table 2: Fairness metrics used in this study

github repository [5]. All of our selected algorithms for this study have used at least one of these datasets.

3.2 Metrics

We have used two different types of metrics in this study: performance (classification) and fairness metrics. For the performance target, we use four commonly used metrics; accuracy, precision, recall and F1. All of these metrics being on the interval [0,1] and with an optimal value of 1. For the fairness metrics, we use the 4 metrics made available on Table 2.

They are explained as follows:

- **Average Odds Difference (AOD)**: Average of the difference in False Positive Rates (FPR) and True Positive Rates (TPR) for unprivileged and privileged groups; [8]
- **Equal Opportunity Difference (EOD)**: Difference of True Positive Rates(TPR) for unprivileged and privileged groups; [8]
- **Statistical Parity Difference (SPD)**: Difference between probability of unprivileged group (protected attribute PA = 0) getting a favorable prediction ($\hat{Y} = 1$) and probability of privileged group (protected attribute PA = 1) getting a favorable prediction ($\hat{Y} = 1$); [9]
- **Disparate Impact (DI):** Similar to SPD but instead of the difference of probabilities, the ratio is measured. [14]

3.3 Experimental Setup

In order to evaluate and compare the performance of these algorithms on the aforementioned datasets according to the pre-defined metrics, we have obtained the replications of each implementation, made available either on the AIF360 Github repository [5] or by Chakraborty et al. [11]. A wrapper code was then written to run each algorithm on each dataset with each protected attribute 20 times.

Given that 2 datasets contain 2 protected attributes, a total of 9 different results were obtained on this study. Given the replication nature of the study, none of the original code has been changed from what the authors of these works have offered, unless there were minor fixes in order to accept the dataset format provided by AIF360. As such, no special configuration has been set for any of the selected algorithms for any of the runs.

As such, this study has performed a total of 900 runs to obtain the results described in the following section.

3.4 Statistical Analysis

In order to properly compare the performance of these algorithms, we have applied the Scott-Knott non-parametric significance test [16]. This is a technique that applies an hierarchical clustering algorithm to split a series of treatments into equivalence groups. The performance of algorithms within a same group is equivalent. Therefore we are able to observe a best group of techniques.

4 RESULTS

The objective of this work is to compare how well do popular fairness methods fare against the state-of-the-art method FairSMOTE [11]. Not only that, but also to compare and survey possible latent differences between Pre-processing and In-processing algorithms. The following tables for results on each dataset should be read with the following in mind:

- Results reported in numerical form are the median values across 20 runs;
- Scott-Knott ranks are represented in shades of green across the rows of each table. The darkest green is the top rank, and the no color being the worst rank.
- Each metric and each dataset has it's own separate ranking.

4.1 Compas

The Compas dataset has two protected attributes, race and sex. As we can see on the tables on Figures 4 and 5, in terms of overall performance, the adversarial debiasing takes the cake. Followed closely by FairSMOTE on both attributes. However when speaking on fairness both the Exponential Gradient Reduction and the Reweighing algorithm are the top performers, outperforming most other algorithms in all metrics by a large margin.

One thing that must be noted is that Chakraborty et al's [11] statements hold true. Their algorithm do not lose in performance when we look at the F1 score, at least on this dataset. But at the same time they are not the best in fairness.

In general terms, the best algorithm in this category is the Exponential Gradient Reduction, followed closely by the Reweighing. It is interesting to note that the Reweighing technique has a minimal effect on the Disparate Index metric, and the Exponential Gradient Reduction has the best metric, while for the other fairness metrics they are a even match.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.68	0.64	0.66	0.65	0.64
PREC	0.69	0.66	0.67	0.64	0.67
RECALL	0.74	0.72	0.74	0.61	0.68
F1	0.71	0.69	0.70	0.62	0.67
AOD	0.24	0.05	0.30	0.23	0.01
EOD	0.21	0.01	0.65	0.41	0.04
SPD	0.27	0.07	0.22	0.11	0.08
DI	0.67	0.90	0.34	0.26	0.12

Figure 4: Compas (Sex) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.68	0.64	0.65	0.65	0.63
PREC	0.69	0.66	0.68	0.63	0.67
RECALL	0.76	0.72	0.70	0.64	0.67
F1	0.72	0.69	0.69	0.64	0.67
AOD	0.13	0.03	0.27	0.26	0.01
EOD	0.11	0.01	0.61	0.44	0.03
SPD	0.15	0.05	0.25	0.12	0.06
DI	0.80	0.93	0.40	0.25	0.10

Figure 5: Compas (Race) results.

4.2 Adult

The Adult dataset also contains two protected attributes, race and sex. As we can see on the tables on Figures 6 and 7, regarding the performance metrics, all of the methods sit on similar footing with the exception of Optimized preprocessing. With each having a particular metric they favor the most.

However when looking at the fairness metrics, again we are surprised by the lack of performance by Fair-SMOTE when compared to other popular algorithms in fairness. In this case, Adversarial Debiasing takes the win with one exception on the Race protected attribute where Fair-SMOTE outperformed every other algorithm on the Disparate Index.

Trends that continue to hold are, the Reweighing technique continues to have good results, apart from DI. And the Optimized preprocessing with few exceptions continues to heavily underperform.

4.3 Bank

The Bank dataset only contains one protected attribute, age. As seen on the table on Figure 8, in terms of performance, all algorithms are basically top rank, or second best. The only exception being the Optimized preprocessing which was unable to converge in this dataset.

The same trend happens in Fairness, with all algorithms showing very similar results. However, apart from Optimized preprocessing that did not complete a single run on this dataset, Fair-SMOTE was the worst ranked in fairness.

4.4 Heart

The Heart dataset only contains one protected attribute, age. As seen on the table on Figure 9, similarly to the Bank dataset all algorithms performed extremely well. The Optimized preprocessing was also not capable of converging on this dataset, and this trend will hold for the following datasets, henceforth we will not discuss the Optimized preprocessing results for these datasets.

One interesting point is that in this case in terms of fairness, Fair-SMOTE outperformed on almost all metrics with the exception of EOD. And in this case Reweighing had the second best DI score.

4.5 Default

The Default dataset contains one protected attribute, sex. As we can see on the table on Figure 10, similarly to both the previous datasets, most algorithms had really good metrics for performance and fairness. However, the Adversarial Debiasing was not able to achieve significant scores on this dataset, we conjecture this to be the fault of the adversary in the network having overcome the predictor completely.

In this dataset however Fair-SMOTE was outperformed in terms of fairness in almost all metrics, whe compared to the Exponential Gradient Reduction and the Reweighing algorithm.

4.6 MEPS

The MEPS algorithm also only contains one protected attribute, race. As seen on Figure 11, again the Adversarial Debiasing was not able to achieve significant performance or fairness scores. However, the other algorithms had almost identical results in all metrics with

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.82	0.83	0.79	0.72	0.77
PREC	0.73	0.74	0.55	0.45	0.52
RECALL	0.40	0.48	0.71	0.74	0.71
F1	0.51	0.58	0.62	0.56	0.60
AOD	0.04	0.07	0.08	0.13	0.03
EOD	0.05	0.11	0.38	0.46	0.10
SPD	0.09	0.12	0.29	0.31	0.15
DI	0.45	0.40	0.75	0.67	0.41

Figure 6: Adult (Sex) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.82	0.84	0.79	0.74	0.78
PREC	0.75	0.72	0.54	0.47	0.53
RECALL	0.38	0.52	0.74	0.74	0.74
F1	0.50	0.60	0.62	0.57	0.61
AOD	0.01	0.04	0.11	0.12	0.00
EOD	0.02	0.07	0.35	0.40	0.02
SPD	0.04	0.07	0.22	0.27	0.08
DI	0.71	0.61	0.63	0.67	0.21

Figure 7: Adult (Race) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.71	0.80	0.80	0.00	0.77
PREC	0.78	0.78	0.77	0.00	0.74
RECALL	0.55	0.81	0.82	0.00	0.79
F1	0.63	0.79	0.79	0.00	0.76
AOD	0.12	0.11	0.37	1.00	0.03
EOD	0.20	0.11	0.50	1.00	0.03
SPD	0.04	0.28	0.10	1.00	0.20
DI	0.86	1.58	0.19	0.00	0.40

Figure 8: Bank (Age) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.68	0.81	0.84	0.00	0.76
PREC	0.68	0.80	0.85	0.00	0.72
RECALL	0.64	0.77	0.80	0.00	0.82
F1	0.64	0.78	0.82	0.00	0.75
AOD	0.25	0.17	0.15	1.00	0.11
EOD	0.16	0.12	0.54	1.00	0.10
SPD	0.36	0.35	0.35	1.00	0.37
DI	0.39	0.42	0.65	0.00	0.53

Figure 9: Heart (Age) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.78	0.80	0.80	0.00	0.77
PREC	0.00	0.78	0.77	0.00	0.74
RECALL	0.00	0.81	0.82	0.00	0.79
F1	0.00	0.79	0.79	0.00	0.76
AOD	1.00	0.11	0.37	1.00	0.03
EOD	1.00	0.11	0.50	1.00	0.03
SPD	1.00	0.28	0.10	1.00	0.20
DI	0.00	1.58	0.19	0.00	0.40

Figure 10: Default (Sex) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.78	0.87	0.86	0.00	0.76
PREC	0.09	0.66	0.63	0.00	0.40
RECALL	0.08	0.40	0.47	0.00	0.75
F1	0.08	0.50	0.54	0.00	0.52
AOD	1.00	0.03	0.05	1.00	0.01
EOD	1.00	0.05	0.14	1.00	0.03
SPD	1.00	0.03	0.08	1.00	0.05
DI	0.00	0.74	0.61	0.00	0.14

Figure 11: Meps (Race) results.

the exception of a large win in Recall by Reweighing and a hard loss by Reweighing on DI.

Interestingly enough for this dataset, Fair-SMOTE performed equally as well as Exponential Gradient Reduction and Reweighing. With Reweighing keeping its trend of providing little effect on DI on most datasets.

4.7 German

Finally the German dataset, which only contains one protected attribute, sex. On the table on Figure 12 We can see that both in performance and in fairness for most metrics there is a very large diference in performance between the Exponential Gradient Reduction algorithm and the others.

On Fairness however the Reweighing algorithm was as performant as the Exponential Gradient Reduction, with the recurring exception of DI.

4.8 Overall

As we look at the table on Figure 13, we can see the overall results for each algorithm across all datasets. This should be taken together with the following information in mind.

The median runtime of each algorithm on all datasets was:

- Reweighing: 6.75s
- Optimized preprocessing: 15.25s
- Fair-SMOTE: 27.25s
- Exponential Gradient Reduction: 90s
- Adversarial Debiasing: 150.5s

From here we can see that, tied together are the Exponential Gradient Reduction and the Reweighing algorithm achieving first rank 39 times, followed by Fair-SMOTE achieving it 30 times. Finally we have Adversarial debiasing achieving first rank 24 times and Optimized preprocessing achieving first rank 2 times.

When separating in performance and fairness the scenario is a little different.

- Reweighing: 15 Performance | 24 Fairness
- Optimized preprocessing: 2 Performance | 0 Fairness
- Fair-SMOTE: 23 Performance | 7 fairness
- Exponential Gradient Reduction: 20 Performance | 19 Fairness
- Adversarial Debiasing: 13 Performance | 11 Fairness

From this we can see that there is a tendency for In-Processing algorithms to have better overall performance score than Pre-processing algorithms. And it also shows that although more expensive in terms of computational costs, the Exponential Gradient Reduction is the most balanced across all the algorithms.

We must note as well that, although Fair-SMOTE only achieved top rank in fairness a total of 7 times, it was consistently second rank. Which together with its great performance, shows that Fair-SMOTE is a very strong algorithm for fairness, being almost 3 times as fast as the Exponential Gradient Reduction.

Reweighing while extremely good in fairness metrics, tends to lack in performance as previously stated by Chakraborty et al. [11]. Although inexpensive it is not an algorithm that can be recommended in face of this study.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	0.63	0.69	0.58	0.00	0.57
PREC	0.72	0.71	0.79	0.00	0.79
RECALL	0.76	0.94	0.55	0.00	0.54
F1	0.72	0.81	0.65	0.00	0.63
AOD	0.18	0.01	0.23	1.00	0.01
EOD	0.18	0.01	0.50	1.00	0.01
SPD	0.17	0.02	0.19	1.00	0.08
DI	5.51	0.98	0.38	0.00	0.17

Figure 12: German (Sex) results.

Method	Adversarial Debiasing	EGReduction + RF	FairSMOTE + RF	Optimized preprocessing + RF	Reweighing + RF
ACC	4	7	4	0	2
PREC	5	5	5	0	4
RECALL	2	3	7	2	5
F1	2	5	7	0	4
AOD	2	4	2	0	9
EOD	3	5	0	0	8
SPD	4	5	3	0	6
DI	2	5	2	0	1

Figure 13: This table has the counts on how many times each algorithm was considered a top ranker in each metric.

The Optimized preprocessing algorithm seems very dependent on certain characteristics of the data. Otherwise the algorithm is incapable of running at all. Which in turn may mean that it is not generalizable enough, contrary to what was stated by Calmon et al. [10]. Henceforth we do not recommend this algorithm moving forward.

Finally the Adversarial Debiasing, although very powerfull when it works well, is extremely expensive to run. We should take into consideration that these runtimes were achieved using the tensorflow-gpu package on a very powerful graphics card. One trial run was made on the CPU solely and this algorithm took over 500 seconds to complete in a median size dataset (Default). It also has problems due to it's adversarial nature which can bring bad results even after an expensive computational time. So this algorithm also cannot be recommended in face of this study.

5 THREATS TO VALIDITY

- **Sampling bias** - We have selected algorithms based on citation counts and popularity on github. And we have only done experiments on five algorithms. This means that the conclusion on differences between pre and in-processing algorithms needs to be taken with this in mind. Not only that but there may exist many other algorithms capable of outperforming the ones we selected here. Since our criteria was mostly academic recognition and popularity between users.
- **External validity** - The conclusions obtained here are only valid for the datasets included in the research, and under the same conditions as explained in our Experimental Setup section. The results of this study might be valid for other datasets and under different conditions, however the proof of this remains as future work.

6 CONCLUSION

This study explored a statement made by Chakraborty et al. [11] on their paper Bias in "Machine Learning Software: Why? How? What to Do?", where the authors stated that no comparison to other existing fairness algorithms, specifically the ones used in this study, was necessary. Due to the fact that those algorithms represent a large degradation on learner performance. We compared these algorithms on their performance on four metrics and also on four distinct fairness metrics.

As seen in our results section, the results show that the comparison was granted. There are algorithms that will often outperform Fair-SMOTE while maintaining or even providing better fairness metrics to learners. And as a matter of fact, an algorithm different than Fair-SMOTE, the Exponential Gradient Reduction, provided the best returns in both performance and fairness, granted at an increased computational cost. Even so, after reviewing these four state-of-the-art fairness algorithms we can confidently say that we recommend that either Fair-SMOTE or the Exponential Gradient Reduction should be selected as baselines for any future work in this arena.

Future work from this paper would include exploring different state-of-the-art algorithms under the same conditions to add data-points to the tables, exploring more datasets in fairness literature.

And improving this brief overview of the state-of-the-art in fairness to a full fledged systematic literature review.

REFERENCES

[1] 2021. https://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212

[2] 2021. https://nypost.com/2017/11/30/google-translates-algorithm-has-a-gender-bias/

[3] 2021. http://fairware.cs.umass.edu/

[4] 2021. https://2019.ase-conferences.org/home/

[5] 2021. aif360: a comprehensive set of fairness metrics for datasets and machine learning models, explanations for these metrics, and algorithms to mitigate bias in datasets and models. https://github.com/Trusted-AI/AIF360

[6] 2021. fate: fairness, accountability, transparency, and ethics in ai. https://www.microsoft.com/en-us/research/theme/fate/

[7] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[8] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).

[9] Toon Calders and Sicco Verwer. 2010. Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery* 21, 2 (2010), 277–292.

[10] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems* 30 (2017).

[11] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in Machine Learning Software: Why? How? What to do? *arXiv preprint arXiv:2105.12195* (2021).

[12] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[13] Jeffrey Dastin. 2021. Amazon scraps secret AI recruiting tool that showed bias against women. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

[14] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[15] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[16] Nikolaos Mittas and Lefteris Angelis. 2012. Ranking and clustering software cost estimation models through a multiple comparisons algorithm. *IEEE Transactions on software engineering* 39, 4 (2012), 537–551.

[17] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.