FACULTY INNOVATION PROPOSAL: Sony Research Award Program

Prof. Tim Menzies, IEEE Fellow
Com.Sci. NC State, USA, http://menzies.us,
timm@ieee.org, +1-304-376-2859

1 Title

M4ME: Mitigating for Malicious Explanations

2 Focused Research Theme

Fairness in AI

3 Abstract

We seek to support an AI economy where models, learned by AI, are available for use across the web (perhaps for some fee for service). Sometimes, AI models have discriminatory properties[1] so this AI economy needs monitoring tools that can detect unfair models. Current methods for assessing the fairness of a model assume that verification tools can access extensive details about the model. Yet there are valid privacy, pragmatic and proprietary reasons why this may not be possible. For example, model owners want to hide details, least their competitors reverse engineer their model. Also, the data used to build a model may reveal sensitive information about individuals. Furthermore, owners of the models (inside the model store) want to keep those models details in-house, at least until they recoup their development cost via renting out their model. Hence we seek methods that can:

Reliably assessing how fair a black box model is, given limited labeled data.

Currently AI explanation tools can succinctly study complex model, but those same methods allow malicious actors to hide certain discriminatory properties of their model (for examples of this, see the next page). To mitigate this problem we must make it harder (or, indeed, impossible) for malicious actors to learn how to deceive us. To that end, we propose *M4ME*, an algorithm for measuring then mitigate unfairness in black-box models.The core idea behind *M4ME* is to

Reduce the information available to the malicious actor.

In summary: we build our explanations via semi-supervised learning that only needed limited interaction with the model. Our conjecture, to be tested here, is that such limited interactions also limit the ability of malicious actors to learn how to deceive us

4 Differentiation from the Current State-of-the-art

The nearest work to this proposal comes from Ji et al. [27] who measure fairness in black-box models via Bayes sampling (on the available data) to extrapolate a much bigger data set. We differ from that approach as follows. *Firstly*, the meta-review of the Ji et al. paper from NIPS'20[2] commented that "(their paper) is conceptually and mathematically sound. The significance of the contribution (an auditor tool only, instead of an auditor plus a mitigation tool) is however at the low side". To say that another way, a criticism of that prior work is that:

After unfairness measurement should come unfairness mitigation.

Accordingly, we offer not just a way to measure unfairness, but also ways to mitigation methods.

Secondly, Ji et al. rely on parametric assumptions about the data and the learner that build a model from that data. While such assumptions might be true, they cannot be checked in the black-box case. Hence, we use non-parametric methods for our data extrapolation (specifically, we will use the data manifolds seen in the leaf nodes of Figure 2b.)

[1] Sadly, there are too many examples of this. There are too many examples of AI software exhibiting unfair/biased behavior based on privileged attributes like sex, race, age, marital status. For example, Amazon had to scrap an automated recruiting tool as it was found to be biased against women [13]. Also, a recidivism assessment models used by the criminal justice system was found to be more likely to falsely label black defendants as future criminals (twice as often as white defendants) [9].

[2] https://proceedings.neurips.cc/paper/2020/hash/d83de59e10227072a9c034ce10029c39-Abstract.html

5 Methods

Slack et al. [36] have recently shown that explanation algorithms are vulnerable to malicious human operators trying to hide discriminatory models. In their work, they discuss explanations generated via stochastic methods. For example, as shown in Figure 1a, the model-agnostic LIME explanation algorithm [34] generates random instances near to some example. After building some linear models from that generated data, LIME can explain what factors can change the classification of a particular example.

The problem here is shown in Figure 1b. Here, Slack et al. [36] show the raw data (in **red**) and data synthesized by LIME (in **blue**). While the **blue** dots are near the **red** dots, Slack et al. show that they are distinguishable via automatic classifiers. That is. the model can distinguish (a) queries-about-explanation from (b) normal-queries seen in day-to-day operation. This means that:

> *Malicious actors can lie about discriminatory properties of their model by offering model1 to queries-about-explanation and model2 to normal-queries from day-to-day operation.*

Figure 1: About LIME.

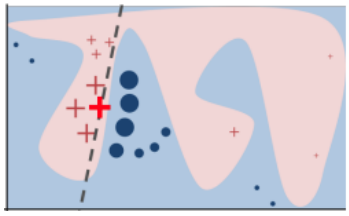


Figure 1a: *LIME samples around the red cross to learn delta between blue,red classes.*

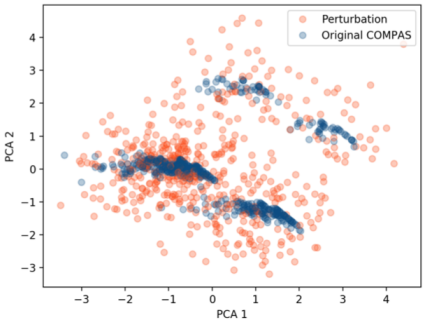


Figure 1b: *Red/blue = original/invented data. From [36].*

One response to the issues raised by Slack et al. is to make LIME's artificial examples indistinguishable from the real data (i.e. to ensure that all the **blue** dots fall at the same location as the **red** dots). To that end, Ji et al. [27] applied extensive and elaborate Monte Carlo sampling to learn everything they can about the distributions in the data. But their sampling methods made numerous normality assumptions about their domain, and many domains do *not* correspond to those assumptions. Also, their methods only detected unfair models. They did not go to the next step of offering mitigation methods. Further, despite their best efforts, their methods still made enough queries to the model such that the Figure 1b results might hold; i.e. a model could distinguish the special queries (the **blue** dots) being made by Ji et al. This, in turn, means that that that model could then offer misinformation to the Ji et al. queries.

Rather than patch the Ji et al. methods, we take a different approach. If the problem is that the model in Figure 1b can distinguish the fairness queries, then *make fewer queries*. The rest of this proposal offers *M4ME*, a semi-supervised learning method that replaces the tools of Ji et al. *M4ME* severely limits the number of the **blue** queries in Figure 1b such that it becomes impossible to learn a model that distinguishes the **blue** from the **red**.

To explain *M4ME*, we will answer the following questions:

- How to measure unfairness? (see §5.1)
- What is semi-supervised learning? (see §5.2)
- How to compute fairness using semi-supervised learning? (see §5.3)
- How to mitigate unfairness, if it is detected? (see §5.4)

5.1 How to measure unfairness?

The standard framework [16] for measuring unfairness is to:

- Recognize the *protected* social grouping (gender, different racial groups, medical status, etc);
- Contrast the performance of a learner across these group;
- Declare a model "unfair" if its performance is different for different protected social groups.

Within that framework, many measures have been proposed, the most common of which are the fairness metrics of Table 1. The problem with these measures is that they require extensive knowledge of the performance of the model on test data (specifically, all the true and false positive and negative rates seen when the model is executed). Note that to access that data, we need to add more dots to Figure 1b.

To measure the fairness, while also thwarting malicious actors, we need to be able compute the metrics of Table 1 use very few queries. For that purpose, we use semi-supervised learning.

Table 1: Definition of the performance and fairness metrics used in this study. TP,TN,FP,FN denotes true positives, true negatives, false positives, false negatives, respectively.

Performance Metric	Ideal Value	Fairness Metric	Ideal Value
Recall = TP/P = TP/(TP+FN)	1	**Average Odds Difference (AOD)**: Average of difference in False Positive Rates(FPR) and True Positive Rates (TPR) for unprivileged and privileged groups [14]. TPR = TP/(TP + FN), FPR = FP/(FP + TN), $AOD = [(FPR_U - FPR_P) + (TPR_U - TPR_P)] * 0.5$	0
False alarm = FP/N = FP/(FP+TN)	0	**Equal Opportunity Difference (EOD)**: Difference of True Positive Rates (TPR) for unprivileged/ privileged groups [14]. $EOD = TPR_U - TPR_P$	0
Accuracy = $\frac{(TP+TN)}{(TP+FP+TN+FN)}$	1	**Statistical Parity Difference (SPD)**: Difference between probability of unprivileged group (privileged attribute PA = 0) gets favorable prediction ($\hat{Y}$ = 1) & probability of privileged group (privileged attribute PA = 1) gets favorable prediction ($\hat{Y}$ = 1) [17]. $SPD = P[\hat{Y} = 1|PA = 0] - P[\hat{Y} = 1|PA = 1]$	0
Precision = TP/(TP+FP)	1	**Disparate Impact (DI)**: Like SPD, but it me assures the ratios (not the difference) in probabilities [24]. $DI = P[\hat{Y} = 1|PA = 0]/P[\hat{Y} = 1|PA = 1]$	1
F1 = $2 * Prec * Recall/(Prec + Recall)$	1		

5.2 What is semi-supervised learning?

Semi-supervised learners (SSL) make conclusions after minimal queries to a model. For this proposal, the essential feature of SSL is that they would add very few **blue** dots to Figure 1b.

To explain our use of SSL, we make the following assumptions. First, we assume that data miners are trying to learn some function f from a set of examples of dependent outputs Y and independent inputs X where

$$y_i = f(x_i)$$

That is, the function can transform some example x_i from X into y_i from Y.

Next we assume that some malicious actor is operating a model containing some discriminatory aspects. That malicious actor has to hide the model behind a firewall (to prevent us from inspecting the f structure). But for that model to have its malicious effect, it must be used. That is, we can also assume that there must be some way an outside client can push in some X values and receive back some X values. Under these assumptions:

- We may know nearly nothing about f;
- We only know the Y values seen after we run the model;
- But we have some knowledge of the X values.

To defend that last point, we note that to run the model we need to know the number and range of the variables in X input (since if were otherwise, then we could not even call the model).

Using that publicly available knowledge about X, semi-supervised learners can explore a space of possible X values, while occasionally make a few called to f in order to obtain some Y values. *Semi-supervised learners* [20] train their models by combining a small amount of labeled data (Y values) with a large amount of unlabeled data (X values). *Semi-supervised explanation algorithms* use semi-supervised learning to generate very small explanations [27, 33, 40].

Semi-supervised algorithms [20] utilize *manifold*, *continuity*, and *clustering* assumptions. The *manifold* assumption is that data lies approximately on a manifold of a much lower dimension than the input space. Under this manifold assumption, higher-dimensional data is approximated in a much lower dimension space, with little or no performance loss [32]. When data is spread over just a few underlying dimensions, then there are fewer ways that examples can differ. Hence, there is more *continuity* between nearby examples and we do not need to reason separately about each example. Rather, we can *cluster* similar examples and reason about one item per cluster.

Using those assumptions, there are many ways to extrapolate from a small number of labels to a larger set of data [43]. For example, RMAP [33] (see Figure 2) recursively bi-clusters the data down to

Figure 2: RMAP [33] partitions x values using the FASTMAP [22] random projections.

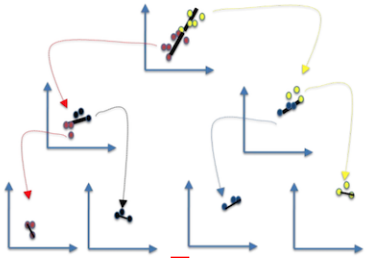


Figure 2a.

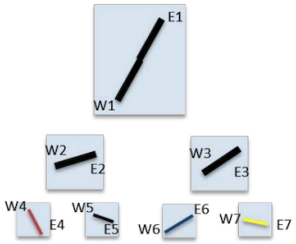


Figure 2b.

NOTES: M is any example (selected at random). E (east) is an example furthest from M and W (west) is an example furthest from E. Note that E, W can be found in time $O(2N)$. If $c = dist(E, W)$ then other examples have distances a, b to E, W, respectively and distance $x = (a^2 + c^2 - b^2)/(2c)$ on a line from E to W. By splitting data on median x, the examples can be then bi-clustered (and so-on, recursively, see Figure 2b).

$n = \sqrt{N}$ of the data. Next, all examples in each leaf clustered are labelled by (i) evaluating the centroid of each cluster; then (ii) propagating that label to all other items in that leaf.

As to other SSL approaches, *self training* algorithms [38] incrementally guesses new labels from a learner trained on all labels seen to date. Further, *GMM with expectation-maximization* algorithms [26] use a Gaussian mixture model to clusters the data (and use those clusters to label the data). Furthermore, *label propagation algorithms* [42] guess labels using a majority vote across the labels seen in nearby examples (or clusters). Label propagation algorithms never update their old labels. *Label spreading* algorithms [41], on the other hand, update old labels using with feedback from subsequent labelling. Label spreading algorithm iterates on a similarity matrix between example and normalizes the edge weights by computing the normalized graph of the Laplacian.

Of all these SSL. we will based *M4ME* on RMAP since:

- It is simplest;
- It has proven effective in our past work [33];
- It is fastest. As discussed in Figure 2, RMAP recursively clusters data in time $Olog(2N)$.
- Its recursive bi-clustering can also be used as a non-parametric alternative to the parametric methods of Ji et al. [27]. Recall that Ji et al. built their approximations of the data using potentially unrealistic assumptions about Gaussians. RMAP, on the other hand, can generated approximations to the data by extrapolating between examples in leaves of the recursively clustered data. Note that this extrapolation process requires no parametric assumptions.

5.3 How to compute fairness using semi-supervised learning?

To assessing how fair a black box model is, given limited labeled data, we propose that:

- Do not query f (i.e. do not add blue dots to Figure 1b);
- Evaluate the fairness measures of Table 1 using the X, Y values generated via semi-supervised learning.

20 years ago, this strategy might seemed foolhardy. "Surely," it might have been said, "measurements from 'made-up' data are far less informative (perhaps, even misleading) than exploring the information in the actual data". However, decades of experience suggest otherwise since the lesson of semi-supervised learning is that large set datas can be approximated via relative few queries [26, 33, 38, 41–43]

The rest of this sections describes two parts of *M4ME* we call *X-reason*ing and *Y-reason*ing. Just to give an overview of the effects of *X-reason*ing and *Y-reason*ing, consider a database of on 10,000 examples :

- *X-reason*ing can prune 90% of data, which making all calls to the model f.
- After that, recursive bi-clustering on the remaining 1,000 examples to $\sqrt{N}$, would return a cluster tree with of depth $d = 5$.
- *Y-reason*ing over this tree would find the best cluster after $2d = 10$ evaluations.

That is, fairness could be assessed via just 10 **blue** dots to Figure 1b. We conjecture (and this would need to be tested as part of this work) that this is too few examples to learn a classifier that can distinguish our queries-for-explanation from any other kind of query.

5.3.1 Details

It turns out that RMAP is not good enough (yet) since it adds too many **blue** dots to Figure 1b. Despite reducing the number of labelled instances, our experiments show that RMAP still queries f sufficient times to make it possible to learn a classifier that can distinguish RMAPs queries from anything else.

M4ME is an experiment in reducing the number of queries made by RMAP. To do so, *M4ME* exploits user knowledge to restrain the X, Y space exploration. *M4ME* assumes that when humans search a model, they are exploring that model for some *reason*. We say that the *X-reasons* and *Y-reasons* denote regions of the X, Y space that the users most care about. Both kinds of reasons can be used to reduce the number of queries.

X-reasons constrain the space of X input. Suppose users want to know about particular kinds of inputs. It terms of $y_i = f(x_i)$ this means that humans are asking a model to explore some constrained subset $X' \subseteq X$ of the space of possible input examples. We call X' the *X-reason* for the human's search.

To give a concrete example of this kind of *X-reason*, we know of software process models with 120 binary X inputs. When we asked humans "true, false, don't care" about those options, we typically find:

- 20 "true,false" biases, where the user cares about some model features;
- And 100 'don't cares", where the user does not care, or cannot decide, about some option.

That is, our users only want to *X-reason* about $\frac{1}{2^{20}} \approx \frac{1}{1{,}000{,}000}$ of the possibilities. *M4ME* exploits these *X-reasons* to prune the tree of Figure 2b. Using the standard entropy calculations, the algorithm sorts the nodes of Figure 2b via $e \times d$ where

e : is the weighted sum of the entropies of the variables in each of that node's subtree (here, we assume that the numeric variables are discretized by, say, the chi-merged algorithm [31] or the Fayyad-Irani discretizer [23]).

d : is the distance of that node from the root of the tree.

The node that minimizes $e \times (d+1)$ is the one that can remove most examples. For that node, we ask the user their opinion on the (say) $n = 2$ variables with highest entropy. That answer is used to prune one sub-tree and any example in the other sub-tree that contradicts that value. The whole process then repeats till some user-specific value. Our preliminary studies have found that stopping after a 90% pruning is useful- and that hyper-parameter would require more research.

As to *Y-reason*ing, for many applications, users do not want to explore *all* the data. Instead, they just want to find what leads to the *best* (or *worst*) case. The definition of what is "best" is often domain-specific might involve trading off between competing concerns. Hence, *M4ME* move beyond simple classification (or regression) to assessment methods that try to optimize (e.g.) all the five goals shown left-hand-side of Table 1[3].

We call these the *Y-reasons* and they represent user preferences over the Y labels. Starting at the root of the tree found above, we evaluate the two most distant points, then apply some multi-objective domination predicate (see Table 2) to determine which half we prefer. The other half is then deleted. This process recurses into the surviving sub-tree. Note that for a tree of depth d, this search navigates to a leaf after $2d$ evaluations.

Table 2: Notes on domination predicates for multi-goal reasoning.

When dealing with 1 goal, a simple $\leq$ function can rank goal values of E (east) and W (west). But for multiple-goal reasoning, examples must be ranked across many goals values.
Binary domination says E is better than W if E has at least one better goal value (and zero worse goal values) than W.
For more three or more goals, binary domination has trouble distinguishing examples [37], in which case Zitler's *continuous domination* predicate [44] is recommended [35].
Continuous domination extends binary domination by summing the actual difference in goal scores.
For individuals x, y have n goals variables (each of which has been normalized 0..1, min..max, to x', y'). Zilter says X is better than Y if the mean loss moving X to Y is less than the mean loss moving Y to X; i.e. $better(x,y) = Loss(x',y') < Loss(y',x')$ where $x' = norm(x)$, and $y' = norm(y)$ and $Loss(a,b) = -\sum_i^n k^{w_i * (a'_i - b'_i)/n}$ where k is some constant (usually 2.7183) and $w_i = -1$ if we want to minimize goal i (otherwise, $w_i = 1$).

This process returns one optimal cluster, that we classify as "best" and other non-best clusters we classify as "rest". A standard classification algorithm can then be applied to these best and rest examples, after which point all the fairness metrics for the protected attributes can be computed.

[3] We do not presumes that these are the only goals that users will ever want to explore. Our methods treat these goals as an input parameter; i.e. our rig would generalize to other kinds of goals.

Table 3: Data sets used in many papers on fairness. As the field of fairness testing evolves, we expect this table to grow much larger. Hence, as part of this work, we would monitor the data sets used in this research arena (and we would test all our methods on all that growing space of data).

Dataset	#Rows	#Features	Protected Attribute	Description
Adult Census Income [1]	48,842	14	Sex, Race	Individual information from 1994 U.S. census. Goal is predicting income >$50,000.
Compas [8]	7,214	28	Sex,Race	Contains criminal history of defendants. Goal is predicting re-offending in future
German Credit [2]	1,000	20	Sex	Personal information about individuals & predicts good or bad credit.
Default Credit [10]	30,000	23	Sex	Customer information for people from Taiwan. Goal is predicting default payment.
Heart Health [3]	297	14	Age	Patient information from Cleveland DB. Goal is predicting heart disease.
Bank Marketing [11]	45,211	16	Age	Contains marketing data of a Portuguese bank. Goal is predicting term deposit.
Home Credit [12]	37,511	240	Sex	Loan applications for individuals. Goal is predicting application accept/reject.
Student Performance [6]	1,044	33	Sex	Student achievement of two Portuguese schools. Target is final year grade.
MEPS-15,16 [7]	35,428	1,831	Race	Surveys of families, individuals, medical providers, employers. Target is "Utilization".

5.4 How to mitigate unfairness, if it is detected?

As said above, our goals are two-fold:

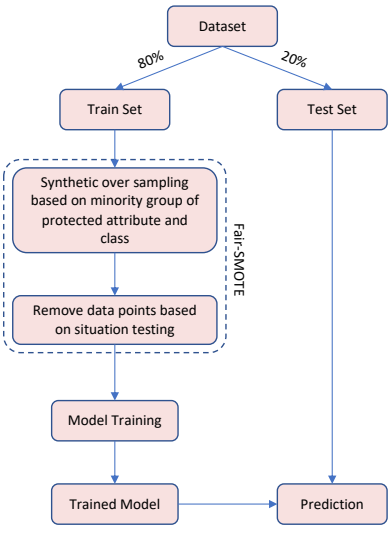
1. Reliably assessing how fair a black box model is, given limited labeled data. For this goal, we use the *X-reason*ing and *Y-reason*ing described above.
2. And if unfairness is detected, then mitigated that. This mitigation process is discussed in this section.

In the last section, we build a surrogate data set via SSL. Here, we propose *balancing methods* to remove unfairness. While these methods come from prior work [19], that previous work assumed access to all the data. In this work, we test if *balancing* works on data artifically generated via SSL (using the above methods).

To explain *balancing*, we first start with the common suggestion about reducing unfairness; i.e. "just remove the protected attribute". This turns out not to be so useful. In some cases, the effects of protected attributes are "essential"; i.e. "gender" can have an influence on an obstetric diagnosis (e.g. instances with"gender=male" cannot get pregnant). Hence, a repeated observation from the fairness testing literature is that improvements to the fariness metrics also means a deterioration in performance metrics (e.g. see experiments with instance reweighing [28], prejudice remover regularization [30], adversarial debiasing [39], equality of opportunity [25], and a reject option classifier [29] and other techniques [18, 25, 28–30, 39]). Consequently, a pessimistic truism in the software fairness testing community is that, as said by Berk et al. [15]: *"It is impossible to achieve fairness and high performance simultaneously (except in trivial cases)."*

In 2021, PI Menzies and Chakraborty and Majumder and [19] found they could reverse the pessimism of Berk et al. using their Fair-SMOTE tool of Figure 3[4]. It turns out, that the "Berk effect" (where improvements in either performance or fairness mean degrading the other) can be avoided, if researchers take greater care with how they *remove* and *add* instances to the training data.

Figure 3: Fair-SMOTE. From [19]. Previously, this algorithm has been tested on all the available data. This proposal would be the first to assess Fair-SMOTE in the context of semi-supervised learning.



As to *removing examples*, Fair-SMOTE carefully prunes discriminatory training data, as follows. Situation testing [4, 5] is a legal tactic that defines "discrimination" as similar individuals getting different

[4] That paper won a distinguished paper award at the 2021 ACM SIGSOFT Joint European SE Conference on the Foundations of SE.

	Recall	False alarm	Precision	Accuracy	F1 Score	AOD	EOD	SPD	DI	Total
SMOTE [21] vs Fair-SMOTE [19]										
Win (for Fair-SMOTE)	4	4	1	6	3	33	33	34	32	150
Tie	25	27	29	28	30	2	3	2	2	148
Loss	7	5	6	2	3	1	0	0	2	26
Win + Tie	29	31	30	34	33	35	36	36	34	298/324
Optimized Pre-processing [18] vs Fair-SMOTE [19]										
Win (for Fair-SMOTE)	10	7	4	3	12	1	2	2	3	44
Tie	21	22	26	30	20	34	33	32	31	249
Loss	5	7	6	3	4	1	1	2	2	31
Win + Tie	31	29	30	33	32	35	35	34	34	293/324

Table 4: Fair-SMOTE performs as well, or better, than alternative methods. Results from 50 repeats of Figure 3 on the data of Table 3 where the learner was logistic regression. Pink cells denote results of non-parametric significance and effect size tests (bootstrap and CliffsDelta) where Fair-SMOTE perform statistically better than anything else. From [19].

outcomes. Using a logistic regression model, Fair-SMOTE trains a preliminary model to makes predictions for all instances. The privileged attribute values for every row of training data are then flipped (e.g. male to female, white to non-white). If that change alters predictions, then that row is removed. For the data sets of Table 3, this removed 8% of the examples (median value). Note that in Fair-SMOTE, situation testing follows example creation (described below), so any discriminatory examples generated via example mutation are removed.

As to *adding examples*, Fair-SMOTE takes care to rebalance not just class frequencies (as done with SMOTE [21]) but all ranges of protected attributes. For example, suppose some original training had 20% of data with a criminal record (holding $\frac{4}{5}$ths male and $\frac{1}{5}$ths female) and 80% non-criminal (holding $\frac{6}{20}$ths male and $\frac{4}{10}$ths female). Traditional rebalancing methods (like SMOTE [21]) would generate data with 50% criminal (holding $\frac{4}{5}$,$\frac{1}{5}$th men,women) and 50% non-criminal (holding (holding $\frac{6}{10}$,$\frac{4}{10}$th men,women). Fair-SMOTE's rebalancing, on the other hand, would generate data with 50% criminal (half of which would be men and half of which would be women) and 50% non-criminal (half of which would be men and half of which would be women).[5]

Table 4 compares Fair-SMOTE to traditional SMOTE [21] as well as a prior state-of-the-art algorithm. Optimized Pre-processing (OP) by Calmon et al. [18] tries to find, then fix, unfairness. OP is a data-driven optimization framework for probabilistically transforming data in order to reduce algorithmic discrimination. OP treats bias mitigation as a convex optimization problem with goals to preserve performance and achieve fairness. We chose this work as a baseline because, like Fair-SMOTE, it is also a data pre-processing strategy. Note that contrary to the pessimism of Berk et al.:

- Compared to SMOTE, Fair-SMOTE achieves similar performance and better fairness scores.
- Compared to OP, Fair-SMOTE achieves better performance scores and similar fairness scores.
- More importantly, there is no evidence in Table 4 of the "Berk effect" where improvements in either of performance or fairness mean degrading the other.

This concludes the technical details of this proposal.

5.5 Summary of Novel Features

The novel features of the above are:

- A new semi-supervised learner called *M4ME* that uses *X-reasons* and *Y-reasons* to reduce the number of queries made to a black-box model.
- A new unfairness detection mechanism that reasons over the X, Y values computed by *M4ME*.
- A new unfairness mitigation method based on Fair-SMOTE. External users of a black-box model cannot balancing the raw data and relearning the model (since the details of that model are hidden away behind firewalls). But external to the model, for models that are found to be unfair, they can balance the data found by *M4ME* then relearn their own fairer model.

[5] In our experience, changing class frequencies in this way can confuse some algorithms such as the Naive Bayes class likelihood calculation of $P(H) \times \prod_i P(E_i|H)$. This is particularly true especially in the case of only a few attributes (i.e. i is small). But once the feature size grows to more than half a dozen attributes, then the contribution of $P(H)$ is less important than the $\prod_i P(E_i|H)$ term (which means class rebalancing no longer confuses Naive Bayes).

References Cited

[1] "Uci:adult data set," 1994. [Online]. Available: http://mlr.cs.umass.edu/ml/datasets/Adult

[2] "Uci:statlog (german credit data) data set," 2000. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[3] "Uci:heart disease data set," 2001. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[4] "Situation testing for employment discrimination in the united states of america," 2007. [Online]. Available: https://www.cairn.info/revue-horizons-strategiques-2007-3-page-17.htm

[5] "Proving discrimination cases – the role of situation testing," 2009. [Online]. Available: https://www.migpolgroup.com/_old/portfolio/proving-discrimination-cases-the-role-of-situation-testing/

[6] "Student performance data set," 2014. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Student+Performance

[7] "Medical expenditure panel survey," 2015. [Online]. Available: https://meps.ahrq.gov/mepsweb/

[8] "propublica/compas-analysis," 2015. [Online]. Available: https://github.com/propublica/compas-analysis

[9] "Machine bias," *www.propublica.org*, May 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[10] "Uci:default of credit card clients data set," 2016. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[11] "Bank marketing uci," 2017. [Online]. Available: https://www.kaggle.com/c/bank-marketing-uci

[12] "Thome credit default risk," 2017. [Online]. Available: https://www.kaggle.com/c/home-credit-default-risk

[13] "Amazon scraps secret ai recruiting tool that showed bias against women," Oct 2018. [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

[14] R. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. Natesan Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, R. Kush, and Y. Zhang, "Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," 10 2018.

[15] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," 2017.

[16] Y. Brun and A. Meliou, "Software fairness," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 754–759.

[17] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Min. Knowl. Discov.*, vol. 21, no. 2, p. 277–292, Sep. 2010. [Online]. Available: https://doi.org/10.1007/s10618-010-0190-x

[18] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3992–4001. [Online]. Available: http://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf

[19] J. Chakraborty, S. Majumder, and T. Menzies, "Bias in machine learning software: Why? how? what to do?" *arXiv preprint arXiv:2105.12195*, 2021.

[20] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006. [Online]. Available: http://dblp.uni-trier.de/db/books/collections/CSZ2006.html

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, Jun 2002. [Online]. Available: http://dx.doi.org/10.1613/jair.953

[22] C. Faloutsos and K.-I. Lin, "Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, 1995, pp. 163–174.

[23] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," *NASA JPL Archives*, 1993.

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

[25] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," 2016. [Online]. Available: https://arxiv.org/abs/1610.02413

[26] J. Hui, "Machine learning —expectation-maximization algorithm (em)," 2019. [Online]. Available: https://jonathan-hui.medium.com/machine-learning-expectation-maximization-algorithm-em-2e954cb76959

[27] D. Ji, P. Smyth, and M. Steyvers, "Can i trust my fairness metric? assessing fairness with unlabeled data and bayesian inference," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/d83de59e10227072a9c034ce10029c39-Abstract.html

[28] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, Oct 2012. [Online]. Available: https://doi.org/10.1007/s10115-011-0463-8

[29] F. Kamiran, S. Mansha, A. Karim, and X. Zhang, "Exploiting reject option in classification for social discrimination control," *Inf. Sci.*, vol. 425, no. C, p. 18–33, Jan. 2018. [Online]. Available: https://doi.org/10.1016/j.ins.2017.09.064

[30] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases*, P. A. Flach, T. De Bie, and N. Cristianini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 35–50.

[31] R. Kerber, "Chimerge: Discretization of numeric attributes," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, ser. AAAI'92. AAAI Press, 1992, p. 123–128.

[32] E. Levina and P. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou, Eds., vol. 17. MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper/2004/file/74934548253bcab8490ebd74afed7031-Paper.pdf

[33] A. Lustosa and T. Menzies, "Preference discovery in large product lines," *CoRR*, vol. abs/2106.03792, 2021. [Online]. Available: https://arxiv.org/abs/2106.03792

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[35] A. S. Sayyad, T. Menzies, and H. Ammar, "On the value of user preferences in search-based software engineering: A case study in software product lines," in *2013 35Th international conference on software engineering (ICSE)*. IEEE, 2013, pp. 492–501.

[36] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods," in *3rd AAAI/ACM Conference on AI, Ethics, and Society*, 2020.

[37] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *EMO*, 2006.

[38] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ser. ACL '95. USA: Association for Computational Linguistics, 1995, p. 189–196. [Online]. Available: https://doi.org/10.3115/981658.981684

[39] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," 2018. [Online]. Available: http://www.aies-conference.com/wp-content/papers/main/AIES_2018_paper_162.pdf

[40] T. Zhang, J. Li, M. Han, W. Zhou, P. Yu *et al.*, "Fairness in semi-supervised learning: Unlabeled data help to reduce discrimination," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[41] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16. MIT Press, 2004. [Online]. Available: https://proceedings.neurips.cc/paper/2003/file/87682805257e619d49b8e0dfdc14affa-Paper.pdf

[42] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Tech. Rep., 2002.

[43] X. J. Zhu, "Semi-supervised learning literature survey," 2005.

[44] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2004, pp. 832–842.