Data Science
Computer Science

Capstone Report - Spring 2020

Improving Fairness in Machine Learning Predictions

Yuhao Ding

supervised by
Professor Siyao Guo

Abstract

As machine learning methods are increasingly applied to real-world scenarios, it is crucial to make sure the models we used are fair. Historic biases in the datasets would easily make models discriminate against certain groups of people in terms of race, gender, etc. In this work, we present a novel approach to reduce bias that could be easily applied to machine learning applications. We find that by training a post-correction model that focuses learning and correcting bias patterns, the bias could be reduced by around 80%, with accuracy only limitedly harmed (<1%).

1 Introduction

With the development of modern artificial intelligence techniques, many decision systems have switched from human-power to machine-powered. With the help of historical data, machine models could learn from the data and make future predictions. However, historical biases occur in those data, and a wrongly used algorithm may aggregate the bias inside, for example, by over-relying on certain sensitive attributes such as race and gender [1]. In recent researches, fairness has gained an increasing attention. Especially in fields such as credit scoring or criminal justice, a biased model may create enormous disadvantages for certain groups of people. One of the most well-known example is a research done by a group of researchers who found that in one Florida county, the decision system they used to evaluate recidivism is racially biased [2]. Other examples include the Google translate service that is shown to be gender biased and exhibits strong male defaults, which most likely results from the biased training data [3]. The not-uncommon bias problems that exhibit in those real-life examples has made the fairness problems crucial.

Due to the increasing importance of the fairness field, publications related to fairness have increased dramatically since 2010 [4]. Early efforts simply stop the algorithm from using the sensitive attributes such as race or gender. However, researches show that this naive method is useless because these sensitive features can be inferred from other attributes and thus being reconstructed [4]. We will also show this in our work. To solve this problem, some works thus pre-process the data to make race not inferrable from the inputs. These methods, though intuitive in reducing the bias, are being debated as misleading and inherently harmful. As one research shows that a race-unaware college admission systems show a worse performance than a race-aware system, using the data of the students' eventual performance [5]. In our work, we also acknowledge this and keep the original data unchanged (without masking any sensitive features).

Another crucial element in fairness is the metric used to evaluate bias. As there are different interpretation of biases, a variety of fairness metrics exist. Each metric is meant for different emphasis. For example, parity-based metrics such as statistically parity ensures all groups (such as male and female groups) have the same probability of being classified as positive. This does not consider the potential difference between groups. In our experiments, we consider a confusion matrix-based metrics, *Equalized Odds*, which considers the inherent difference between each groups by ensuring that each group have the same probability as being correctly classified. A more detailed description of fairness definitions can be found in the next sections.

Currently there are three types of fairness-enhance mechanisms: pre-process, in-process and post-process, which is represented in Figure 1. Pre-process happens before model training and usually modify the training data. In-process mechanisms modify the training algorithm such as the objective function. Post-process methods, which we adopt in this work, directly modify the prediction. Currently, there are no concluding works that deterministically select a processing mechanism [1]. In fact, Hamilton shows that these methods' performance varies across different datasets [6]. We choose post-process mechanisms because pre-process mechanisms modify the original data, which includes the testing data which make it hard to explain, and in-process mechanisms are difficult, especially for algorithms such as random forest if we don't have a solid mathematically deduction of the ideal objective function.

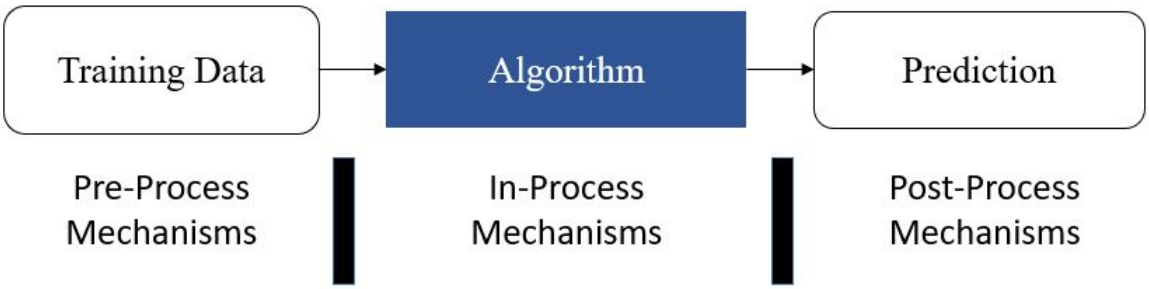

Figure 1: Three Types of Fairness-Enhancement Methods

Our main contribution in this work is:

- **We provide a novel post-process method for fairness enhancement**. Specifically, we trained a post-correction model to correct the output of the prediction algorithm to ensure fairness. In our experiments, we find a 80% increase in fairness method which bring bias score to a low threshold level, and 1% harm to the original accuracy.
- **We systematically tested the performance and stability across two different datasets**. Through these experiments, we understand the model's performance and generalizability. We believe that our model is generalizable and could be fine-tuned on different dataset. For the Adult Income Dataset we mainly used in this work, our model shows great stability.
- **We analyze the results and provided possible explanations of the current performance**. Our methods, unlike other statistical methods, does not require revealing the ground truth labels during testing, and can be trained and tuned during training and validation. We provide several intuitive conclusions based on the empirical results.

We believe our contribution could immediately bring more insights in the fairness field, as this could verify an important hypothesis: **bias patterns are learn-able**. We present this model architecture of using machine learning methods to de-bias machine learning models, as a new approach to tackle the fairness problem.

2 Related Work

Nowadays, an increasing number of decisions have been made by automated artificial intelligence models [1]. In predictions such as household income, credit score, or defendant's recidivism rate, machine learning models may yield bias because historically biased human decisions are used as training data [7]. Angwin et al. use statistical models to show a commonly used risk model to predict recidivism probability in the criminal justice system are heavily biased. They indicate that although the race features are not used in the model, black defendants are more than 20% higher in probability to be incorrectly classified to re-offend [2]. This racial bias favors white defendants and may significantly disrupt the life of a black defendant. Dressel and Farid conduct a study that let humans predict recidivism rates without knowing the defendants' race and these humans' predictions show similar unfairness to black defendants [8]. This further supports that the biases are historically rooted. Therefore, solving the fairness problem is not as easy as excluding certain features. Our work is motivated by this to systematically reduce bias in machine learning models.

There are two legal definitions of unfairness. The first is disparate treatment, which a member in a protected class (race, gender, etc.) is discriminated against because of their memberships

[9]. This is direct discrimination and is unlikely to happen when the sensitive attributes are not used in the prediction. Another is disparate impact, which a group is discriminated indirectly because their sensitive features could be inferred from other features [9].

Based on the definitions, there are several fairness measures invented, such as disparate impact, demographic parity, equalized odds, etc., different measures will have different advantages and disadvantages [1]. We address some metrics that have been widely used in recent literature for binary predictions, some of which will also be used in our work.

- *Demographic parity:* This metric is similar to the legal definitions of disparate impact, this metric requires the proportion of positive predictions are similar across different groups [10]. For example, if 80% of the male are predicted as high income, the percentage of the female being predicted as high income should be similar.

- *Equalized Odds*: As proposed by Hardt et al., a model satisfies equalized odds if both the true positive rate (TPR) and the false negative rate (FNR) are the same across different groups [11]. More intuitively, Chouldechova describes this as achieving "error rate balance" [12], which means the error rate for different groups (for example, male and female) are similar.

- *Equal Opportunity*: Also known as prediction parity, this metric asks for similar true positive rates (TPRs). This is similar to *Equalized Odds* but uses TRPs rates only.

- *Individual Fairness*: This methods requires "similar individuals to be treated similarly" [1]. To define similar individuals, this metric requires a distance metric so that similarity can be measured. Because of this complexity, this metric will not be used in this work.

However, Kleinberg et al. show that different fairness measures cannot be satisfied simultaneously, there is a trade-off inherently behind those metrics [13]. This is also shown in other researches [14, 15, 16]. In fact, as recommended by Pleiss et al., a practical way is to choose only one of the fairness measures according to the application needs [16]. We acknowledge this in our work, thus our main mechanism aims to improve *Equalized Odds*, but we would still provide the result analysis on how our methods might impact other fairness measures.

With these definitions invented, current fairness-enhancing methods could be classified as pre-process, in-process and post-process mechanisms [1]. Pre-process mechanisms mainly change the training data before feeding them to a machine learning model. This comes from the idea that biased training data is the fundamental cause of unfairness. For example, Feldman et al. change attributes to make sure it is impossible to predict the protected attributes (race, gender, etc.), which makes the model impossible to indirectly infer the protected attributes. Similarly, Calmon et al. propose an optimization model to obtain a data transforming method that could both increase fairness and minimize distortion [17]. In-process mechanisms directly modify the prediction algorithm [1]. This could be done by changing the objective function, such as adding a regularization term to enhance fairness[18]. Kamiran et al. adjust the decision tree split criterion to simultaneously minimize the information gain of the protected attribute, which makes predicting the protected attribute as unlikely as possible [19]. Post-process mechanisms, which we adopt in this work, directly change the output of the classifier of the model [1]. Therefore the learned model is used as a black-box model. Hardt et al. propose a statistical method to flip decisions to increase *Equalized Odds* scores. Similarly, Corbett et al. suggest giving different thresholds for each group to increase the probability of a group being predicted as positive [20].

Our work stands upon Hardt et al.'s and Woodworth et al.'s work to use post-processing mechanisms to correct *Equalized Odds* [11, 21], but our approach differs as we use another machine learning model to correct the original outputs. More detailedly, we train a random forest model to predict the labels that are most likely incorrectly labelled, and put a rule-based method to ensure fairness while correcting the outputs of the original model. This approach, to the best

of our knowledge, is novel in the fairness-enhancing field. Some similar research has been done by other scholars. Luong et al. uses a KNN based method to search for unfair labels that lie in an "unfair" situation, and change their labels. We share the same intuition to find problematic points that may bring unfairness [22]. Our work differs as Luong et al.'s method is a pre-process mechanism that uses clustering algorithms, and our work is a post-process mechanism that uses random forest to directly classify incorrect, instead of "unfair", predictions, and we used a rule-based method to ensure fairness. Nina et al. find that ensemble methods could increase fairness as a biased base model could be avoided by using many models so that the biases cancel with each other. Our work uses this idea as we train a random forest, an ensemble model, to post-process model outputs [23]. But we are inherently different as Nina et al.'s works on the original model itself, and our method leaves the original model unchanged, as we add a post-process model to correct the output of the original model. Lehia et al.'s work, which based on Kamiran et al., propose a post-process method to select some predictions to be flipped if they are uncertain, i.e. the output score of a data point is close to the threshold [24, 18]. Our model is similar to their research as we both intend to find points that are possible to be incorrect, but we used a random forest model instead of using the output scores. Kim et al. propose a *multiaccuracy* post-process model to make sure the prediction error of all *identifiable subgroups* are similarly low, based on the validation set [25]. Our approach is similar as we both try to reduce prediction error in all subgroups. However, Kim et al.'s approach choose outputs to flip based on statistics calculation, and our method trains a model to do that. Ultimately, our method invents a novel approach to use another machine learning model correct fairness of the original machine learning model. Our work points to a interesting direction that unfairness brought by a model could also be learn-able. This work aims to show and analyze this idea.

3 Solution

In this work, we proposed a novel post-process model for making fair predictions while ensuring accuracy in models. The intuition is that algorithms' bias patterns are learn-able. Thus, we trained a processing model to correct the behaviour of the previous algorithm. Quantitatively, our post-correction framework could reduce the unfairness by 80%, with the accuracy nearly unharmed (<1%).

3.1 Model Overview

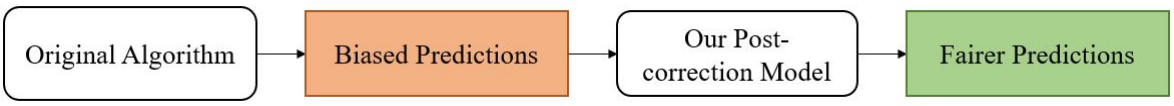

Figure 2: Overview of our methodology: our post-correction model only start training after the original algorithm finished training and made prediction

The nature of our post-processing model is that it only begin processing after the original algorithm finished processing. The overall architecture is in Figure 2. The original algorithm remains untouched and can be replaced by any other algorithms. Our Post-corrector model could see the prediction made by the original algorithm. The prediction goal of the post-correction model is correcting the wrong outputs of the previous algorithm. For example, if the previous algorithm predicts a data point as positive (1), but the ground truth label is negative (1). The post-correction model should predict true (1), which indicates that the data point should be changed to ensure correctness. An example of the data point is shown in Figure 3.

Original Algorithm's Prediction	Post-Correction Model Output	Final Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Explanation of flipping predicted labels.

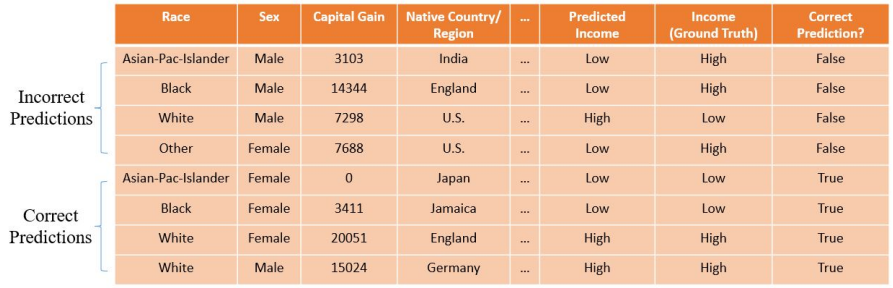

Figure 3: Example data used for training the Post-Correction model. The data used the ground truth label and the prediction made by the original algorithm. The new label simply points out whether the original prediction needs to be flipped.

To be more specific, from Figure 3, we could see that if original algorithm made correct predictions (equivalent to the true positive and true negative predictions), these data points will get a new label *False (0)*, which indicates that this prediction need not to be changed. However, if the original algorithm made incorrect predictions (equivalent to false positive and false negative predictions), these data points gets *True (1)* label indicating that these decisions need to be flipped.

A explanation of flipping decision is shown in Table 1. If the original algorithm predicts a, and post-correction model predicts b. The final output will be the *exclusive or (XOR)* of a and b.

3.2 Model Training, Validation and Testing

During training process, we first train the original algorithm with the training data. Then with the predictions on the training data, we extract the data points that the model wrongly predicted and label them as 1 (indicates need to be flipped). We then randomly extract another set of data points that is correctly predicted by the original algorithm and labelled them as 0 (do not need to be flipped). The proportion of random extracted correct data points to wrongly data points p is a parameter that we need to fine-tuned. This parameter controls the tendency that the post-correction model flips the original data. After obtained this new dataset, we feed the data to the post-correction model. The post-correction model does a new train-test split in this new dataset and train. Figure 4 shows the overall logic of training the model, as explained above.

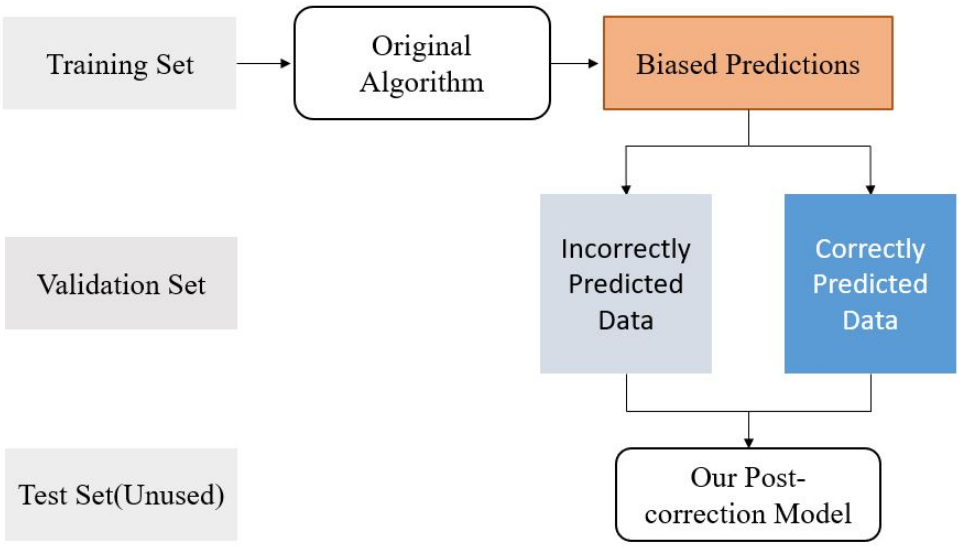

Figure 4: The process of model training. After the original algorithm is trained, we extract the new dataset indicating whether the original prediction needs to be flipped and train the post correction model.

After the model is being trained, we took the trained original algorithm and post-correction model for validation. The validation process begins with feeding the data into the original algorithm and get the predicted data. Then all the predicted data are put in the post-correction model. The post-correction model will then make a prediction indicating whether the predicted data point might be wrong. If the prediction indicates that the prediction is wrong, the model will correct the prediction accordingly. Figure 5 shows the overall logic of validating the model.

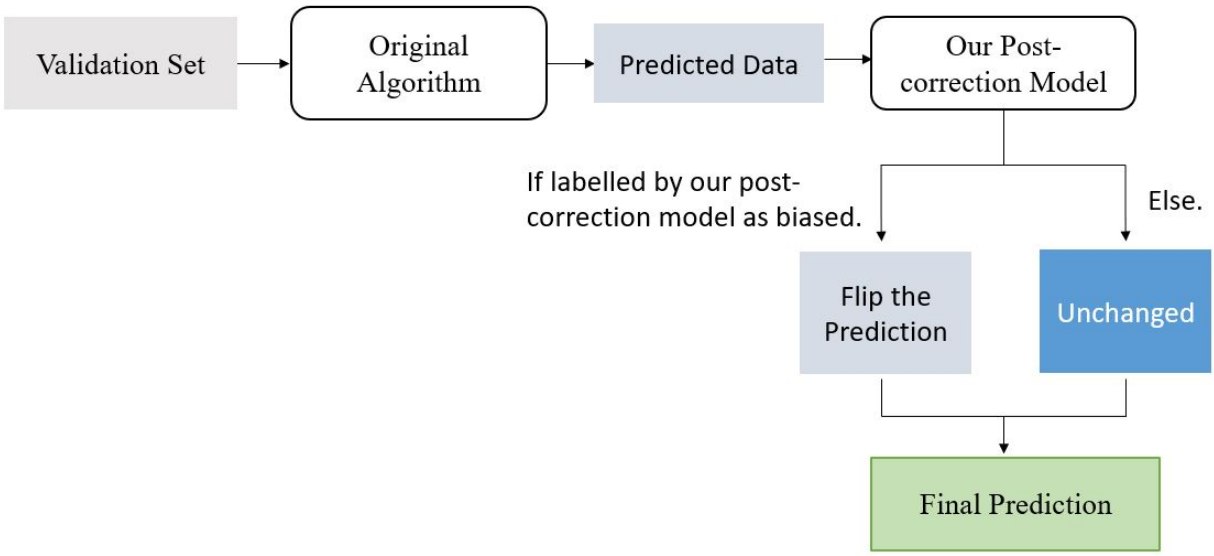

Figure 5: The process of model validation and testing. The prediction by the original algorithm is fed into the post-correction model, if the post-correction model predicted the data point as biased, we flip the prediction of that data point.

The testing process follows the validation process, except that the dataset used is the test set.

4 Results and Discussion

In our experiments, we choose our original algorithm as random forest model, as random forest provides a reasonably good result and very commonly used on our task. For the post-correction model, we experiment with random forest and multi-layer perceptron neural network, our results show that using another random forest post-correction model could greatly enhance fairness while limitedly hurt accuracy. With multi-later perceptron, the result is a bit worse but still we get an improvement. The details of the experiment results are presented in this section.

We experimented our methodology mainly on the Adult Income Dataset and later tested on another German Credits Risk Dataset. As mentioned previously, we put our focus on two criterion: accuracy and fairness. While accuracy is a commonly used criterion, we begin by defining fairness.

4.1 Fairness Assessment

We begin with defining fairness metric, which is crucial in our experiments. In our work, we choose *Equalized Odds* as our metric for fairness because it is suitable and very commonly used for a supervised learning task. In our problem, we tried to predict on a goal Y from features set X based on some labelled training data, and there is a certain protected attribute S, which in our case gender, that we do not want to discriminate on. Our definition of fairness is defined as follows.

Equalized Odds is designed by Hardt et. al, which computes the difference between the false positive rates (FPR), and the difference between the true positive rates (TPR) of the two groups [11]. In our work we calculate the average of the two differences to be the final fairness indicator.

Suppose the protected attribute is S. The prediction is $\hat{Y}$, and the true label is Y. Our metric can be denoted as:

$$
\begin{aligned}
2\Delta EO =& \sum_{y=0}^{1}[|P(\hat{y}=1|S=1,Y=y) - P(\hat{y}=1|S=0,Y=y)|] \\
=& P(\hat{y}=1|S=1,Y=1) - P(\hat{y}=1|S=0,Y=1) \\
&+ P(\hat{y}=1|S=1,Y=0) - P(\hat{y}=1|S=0,Y=0)
\end{aligned}
$$

*The reason that it is $2\Delta EO$ is that we calculate the average rather than the sum.

Thus, the calculation involved calculation with the confusion matrix so some researchers called them "confusion matrix based" approach. To facilitate our calculation, we use the open source tool kit Aequitas to calculate fairness [26].

4.2 Baseline Model

We used two baseline models in our experiments: Original baseline and Random-Flip baseline. Original baseline basically means that no post-correction is made, and the original algorithm is directly evaluated for its fairness and accuracy. Random-Flip baseline randomly flipped the data points with the same amount of data points flipped by our algorithm. This could evaluate how our flipping process works. A detailed explanation is as followed.

Original Baseline: After our testing data is fed into the original algorithm and we get the predicted output, no post-processing is done on these outputs. We directly evaluate the predicted outputs of its accuracy and fairness.

Random-Flip Baseline: After we get the predicted output of the original algorithm, we feed the data to our post-correction algorithm. Suppose if k number of data points is flipped using our post-correction model's output, we randomly flip k number of data points here instead. If our post-correction model is effective, then it should perform better than the Random-Flip Baseline, which show that the post-correction model will point a good direction to which data to be flipped.

4.3 Random Forest as Post-Correction model

Random forest is introduced by Breiman in 2001 [27], it is a extremely effective model as a general-purpose classification and regression method [28]. It is based on decision trees and aggregates (by averaging) several results of decision trees. However, the complexity of the models make it hard to be fully explained, especially in large scale datasets. Some researchers believe that random forest may have a flavor of deep network architectures [29]. From our understanding, the random forest model has presented a high-dimension and complex patterns for classification, which may make it hard to replicate with other machine learning models. Therefore, we also present a result we made using neural network (Multiple-layer perceptron) as a comparison. We find that, the model perform worse with neural networks, which validates with our understanding.

Our first experiments use random forest as our post-correction model. As our original algorithm is also random forest, we find these two algorithms adapts well in our task. In other words, the post-correction random forest performs well in reducing the bias for the first model.

Figure 6 and Figure 7 shows the fairness and accuracy after our model is applied.

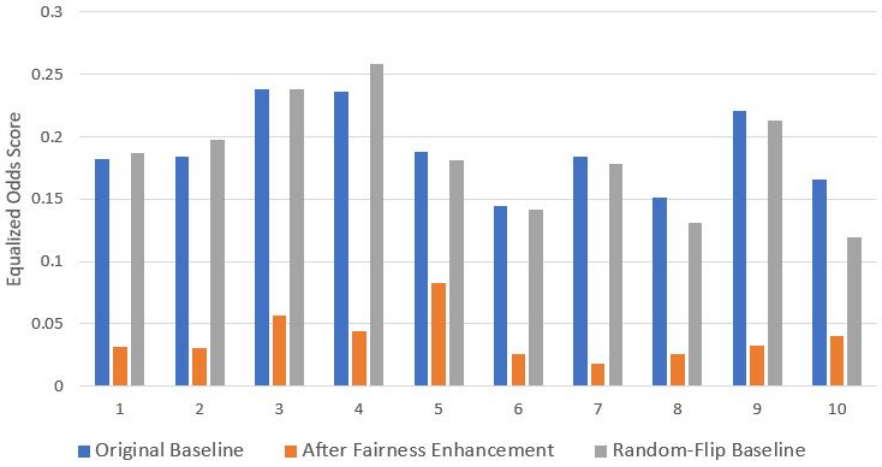

Figure 6: Fairness improvement (the lower the fairer) after using the random forest post-correction model. We can see that fairness (evaluate by Equalized Odds) has been greatly enhanced.

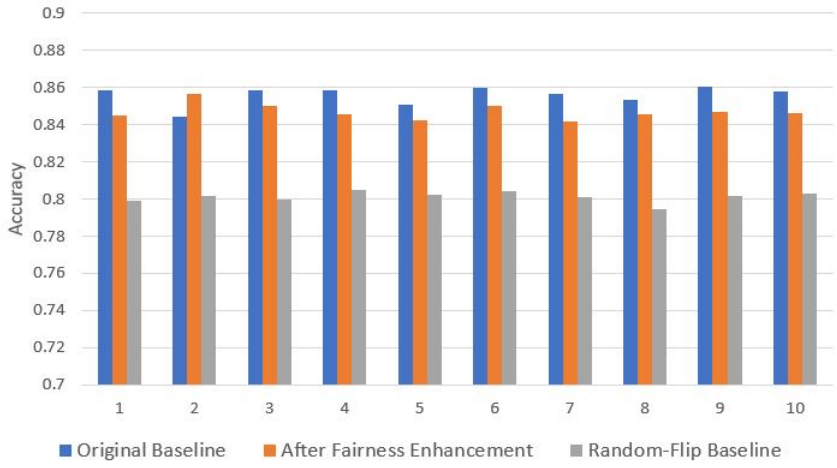

Figure 7: Accuracy change after using the random forest post-correction model. We could see that our model only limited hurt accuracy.

We could see that fairness has been greatly enhanced, as the fairness assessment score (defined by *Equalized Odds*) has been reduced from an average of 18.9% to 3.9%. Accuracy has been hurt only by limited amount. Table 2 shows the percentage improvement. Though accuracy has been hurt by an average of 1%, fairness has been improved for an average of nearly 80%.

	Original	Our Post-correction Model	Percentage Improvement
Accuracy	85.5%	84.6 %	-1.0%
Fairness	18.9%	3.9 %	79.5%

Table 2: Overall performance of our post-correction models based on 100 random trials.

Many hyper-parameters in the random forest model we used are standard settings [1], however, an important hyper-parameter we introduced that need to be tuned is p, which represents the proportion of random extracted correct data points to wrongly predicted data points when we feed them to the post-correction model. The intuition is that this parameter controls the tendency that the post-correction model flips the original prediction. A higher p makes the model less likely to correct the original model, and a lower p, on the contrary, makes the model more likely to make a correction. For example, in the case if $p = 3$, suppose there are 3000 data points in the training set that are wrongly predicted, we random sample an $3000 * 3 = 9000$ correctly predicted data points and combined these 12000 data points as the new training set for the post-correction model.

Figure 8 and Figure 9 shows the fairness and accuracy change when tuning the parameter p, we could see this parameter is crucial to the final fairness and accuracy of the model. Specifically, we have the following conclusions.

- **If p is tuned too large ($>= 3.5$):** This means that in the training set fed to the post-correction model, more data are originally correctly classified data. We find that less data points will be corrected by the post-correction model. In cases where $p >= 4$, less than 50 data points will be corrected, this the fairness score will only change very subtle. In this case, the accuracy is also not effected.

- **If p is tuned too small ($<= 2.5$):** This means that in the training set fed to the post-correction model, more data are originally wrongly classified data. Thus significantly more

[1] n_estimators = 100, min_samples_split = 10, max_depth = 10, bootstrap=True

data points will be corrected by the post-correction model. In our experiments, around 30% of the total validation data will be corrected. In this method, there is a significant increase in false positive rate, leading to a higher bias score and a much lower accuracy score. This is understand-able as the original accuracy is around 85%, thus too many data are being corrected.

- **Appropriate p (around 3):** At around $p = 3$, we see the best accuracy-fairness trade-off point. Fairness score is around 5%, and accuracy is harmed within 1 %. Eventually, we decided to adopt $p = 3$ as the experience value. Indeed, another approach is to used some rule-based methods to determine the optimal p (this involves how to determine the best trade-off point of accuracy and fairness). As this work does not involve on this topic, thus we adopt a reasonably good value $p = 3$.

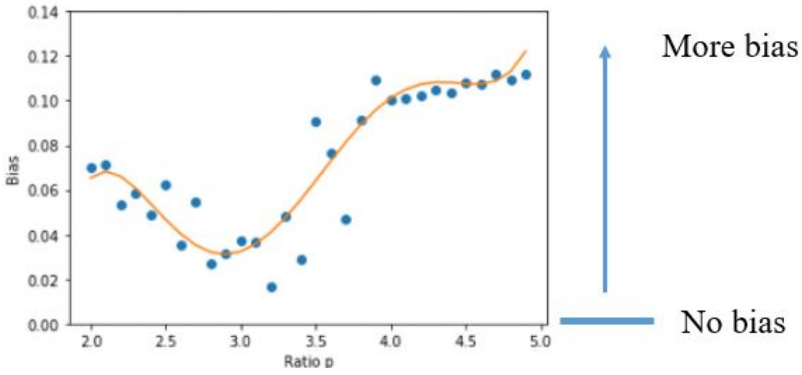

Figure 8: Fairness score with respect to parameter p.

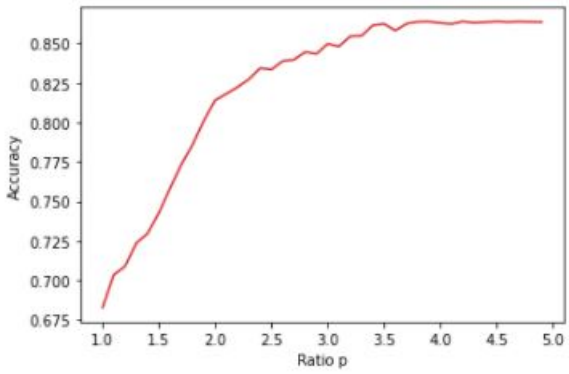

Figure 9: Accuracy score with respect to parameter p.

Here we do our experiments using random forest model, we find that the model works well in reducing the bias. The results could definitely be improved with more careful fine-tuning or model building, which could be done in future work.

4.4 Multi-layer Perceptron as Post-Correction model

As we explained before, random forest model has a nature of complexity in the model itself, we want to see that if neural network works in this problem. In this section, we experiment with using multi-layer perceptron as our post-correction model, and we arrive at a simple conclusion: we find that multi-layer perceptron is not suitable for the task.

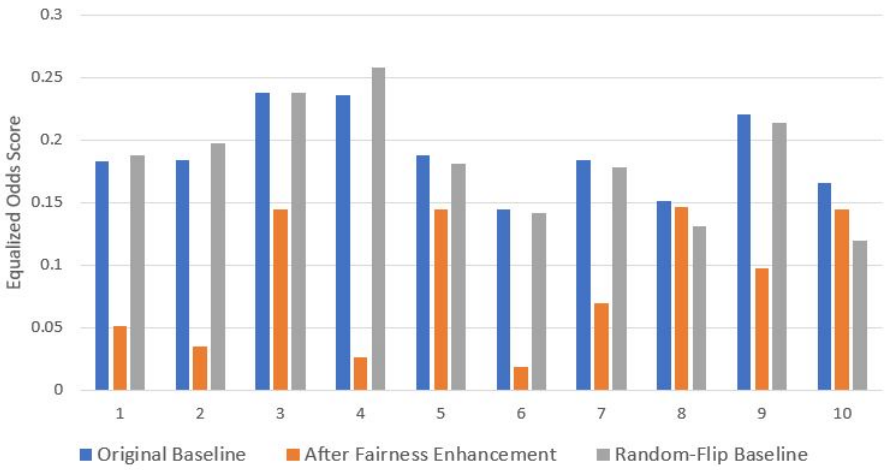

Figure 10: Multi-layer perceptron led to fairer model, but not so good as the previous random forest post-correction model, and it is very unstable.

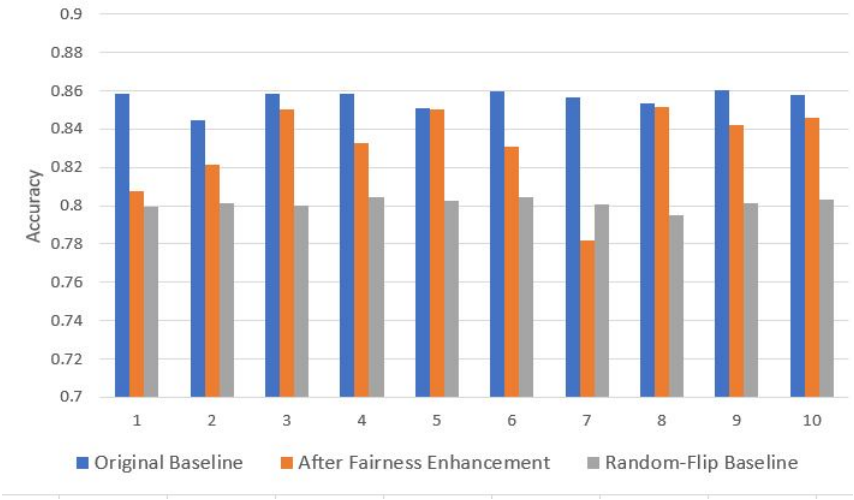

Figure 11: Multi-layer perceptron did not perform better in accuracy as well.

We hypothesize that the result is due to the fact that the multi-layer perceptron could be highly susceptible to highly imbalanced dataset. In our case, as only a small number of data's labels need to be changed, the post-correction model need to make highly imbalanced decisions, thus the model does not perform well.

4.5 Summarize of results

Table 3 shows a summarize of all the main results we achieved. We could see that using random forest model could greatly improve fairness of the model while only limited hurt accuracy. The variance in accuracy and fairness is also pretty small. The multi-layer perceptron, though has some effects in reducing the bias, does not perform stable enough (higher variance). We therefore conclude that random forest as a post-correction model in our case is more effective.

Post-correction Model	Accuracy Score	Fairness Score	Std in Accuracy	Std in Fairness
Original (None)	**85.5%**	18.9%	0.0047	0.0310
Random-Flip Baseline	80.3%	18.5%	0.0027	0.0427
Random Forest	84.6%	**3.9%**	0.0043	0.0178
Multi-Layer Perceptron	83.2%	8.8%	0.0513	0.0217

Table 3: Summarize of Results (10 random trials).

4.6 Experiments on another dataset

We also applied the same techniques on another commonly used dataset: German Credit Risk Dataset by UCI Machine Learning Repository [30]. German Credit Dataset is a very small dataset with only 1000 data instances. It is also very complicated, thus we used the pre-processed version available on Kaggle [2]. Eventually, we have 9 attributes and the goal is to predict whether a data point is good or bad risk. Attributes include age, job, housing, savings account, etc. The protected attribute is sex. We use the same experiment settings as above, and used random forest as the post-correction model. For the original algorithm, we also used random forest.

Result is shown in Table 4. From the results, we could see that as the dataset are inherently very different, the results are very different. Our method do have a improvement in fairness scores. The improvement is not much, and the model does not work as stable as previously. We believe that this mainly due to the fact that there are too less data available, which make our post-correction model greatly underfit.

Post-correction Model	Accuracy Score	Fairness Score	Std in Accuracy	Std in Fairness
Original (None)	**76.52%**	6.9%	0.0294	0.0347
Random Forest	74.8%	**4.3%**	0.0271	0.0319

Table 4: Results in German Credit Dataset (10 random trials).

Specifically, in one example, the model makes 18 flips, in which 8 are correct and 10 are incorrect flips. However, these flips successfully bring down the difference of False Positive rate to <2%. Still, the model is greatly unstable because one data point may account for nearly 1% in this dataset (suppose the test set is 20%, then there are only 200 data points in the testing set). If our model need to be used in these datasets, more fine-tune process need to be done. This also explains why the variance is huge, as small changes in a few predictions may greatly impact overall accuracy and fairness.

5 Discussion

As our result is more stable in the Adult Income Dataset, we focus this discussion on this dataset only. We'll begin with exploring the Adult Income Dataset.

5.1 Probing the dataset

5.1.1 Data distributions

As shown in Figure 12 and Figure 13, the dataset is greatly imbalanced in both sex distribution and income distribution. Specifically, we have less data with female and less data with high income.

Though this is not a big problem, the bigger problem that caused bias is shown in Figure 14. We could see that male has more portion of higher income data points, and female has less

[2]https://www.kaggle.com/uciml/german-credit

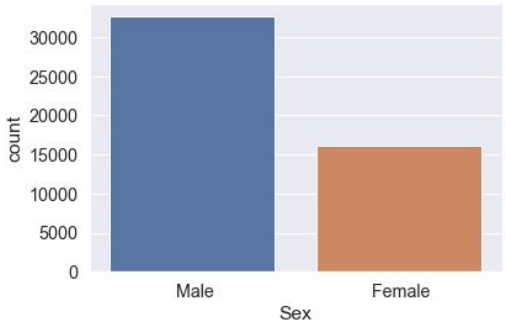

Figure 12: Sex distribution.

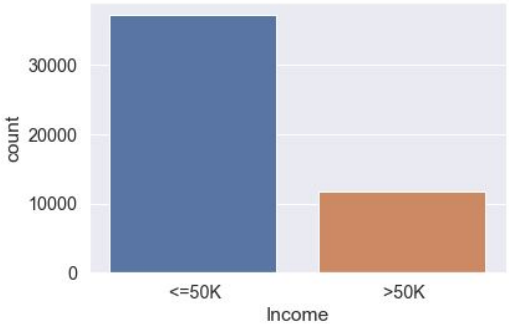

Figure 13: Income distribution.

portion. They may create a problem if there are more unseen data, then the trained model with these data will provide a natural disadvantage for female data points.

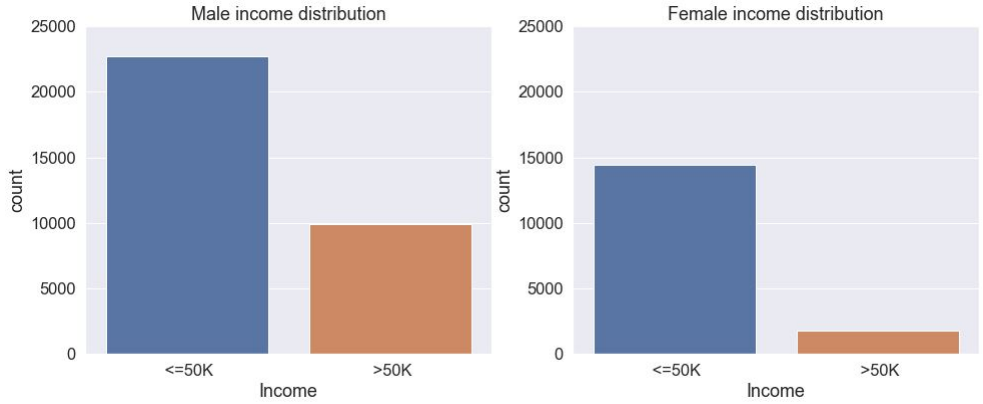

Figure 14: Imbalanced data with male and female income.

5.1.2 Indirect bias

Could the problem be solved if we just simply removed the sex attributes? As we explained earlier, there are indirect biases. Here, the sex attributes could be inferred from other attributes. Here for example, in Figure 15, we visualize some of the occupation's gender distribution. For this, we could easily see that gender attributes could be easily inferred, as the distribution of the male and female data points are radically different.

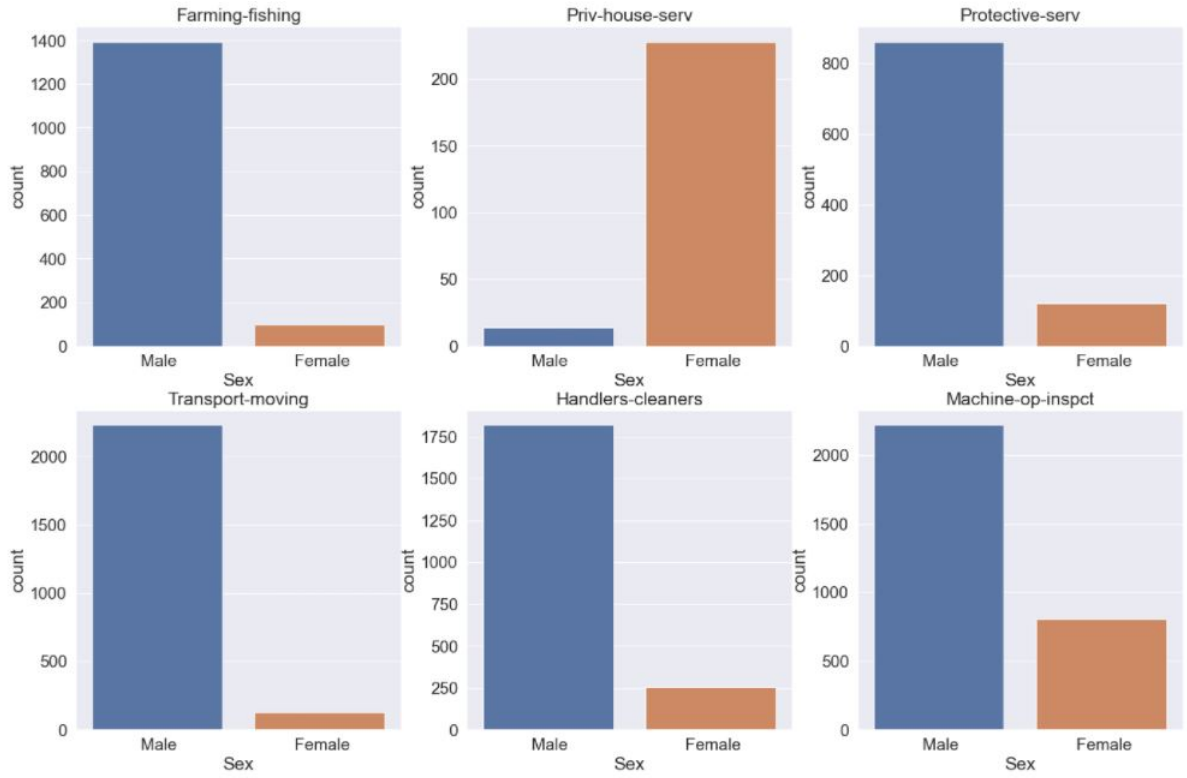

Figure 15: Indirect bias examples: gender could be easily inferred from other attributes, such as occupation.

With our experiments, we find such indirect bias is very common. In fact, we trained a random forest model to predict gender, and the model gets 86% prediction accuracy, which further verified our thoughts. Therefore, in our experiments, we have retained all sex and gender attributes. And our result, as we have shown, are not affected when sensitive features are used as input.

5.2 Explore post-process models

To understand our post-correction model, we first take a deep look into the post-correction process. As we mentioned earlier, our post correction model takes a training set consists of wrongly predicted attributes, along with p times of random sampled correct attributes as input. The model is trained with the purpose of being able to identify biased data points.

As using neural network does not yield a better result, in this section we only look at the results for using random forest and the post-correction model. In our analysis, we have several findings that meet our intuition, and also have several thoughts. We outlined them here.

Accuracy-Fairness trade-off: Accuracy-fairness trade-off is one of the widely acknowledged phenomenon in the fairness field. We expect this in our results, too. The intuition is simple, biases occurs in the dataset, to remove it, we must. in some way, change the performance of the original model. Given that the original model is fine-tuned to get the most optimal result, we will definitely lose some accuracy. In our model, we find that the post-correction model has a validation accuracy of nearly 75% and actually testing accuracy of around 45%. Therefore, around 55% of the correction are wrong, from our understanding, this is reasonable as many data points that are predicted to be biased could be correct (for example, a low education man could really be hard-working and get a high income, but the model ma find that the original algorithm predicted him to be high income only because of his gender). This shows a typical accuracy-fairness trade-off.

Attributes Used in the Post-correction model: Our intuition is that the model will use attributes that are more likely to cause bias (which includes the protected attributes themselves). Figure 16 shows the most used attributes that in the first four layers of the random forest. We find that Sex attributes stably occur in around 50% of the cases, which verified our intuition. We can also see that the mostly used attributes are capital gain, education number and age (The top 1 attribute "native country: Yugoslavia" is used purely because it only has a few data points, which makes using it very perfect, because you can get a perfect information gain. We exclude this in this discussion.) We believe this is because if the model wants to predict which data is biased, the model has to look at the major features that contributes to this decision. Still, visualizing a random forest is still naive, but we can still safely conclude that the sensitive attributes is frequently used in the post-correction model.

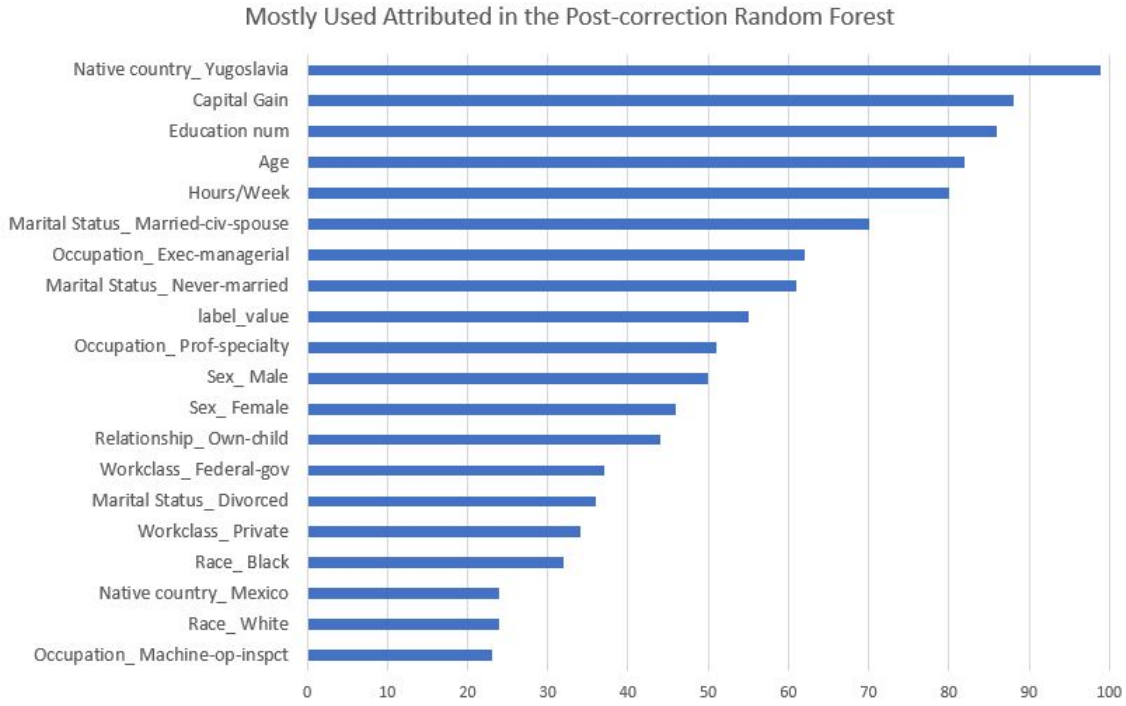

Figure 16: Most used attributes in the post-correction random forest.

How does fairness improved? We have previously shown that the fairness is greatly enhanced using our model, the next question is how. Our understanding is based on the intuition that as the original algorithm has a bias towards female (female data points are discriminated and a more likely to be falsely predicted). Our post-correction model will therefore get more female data inputs, and has a stronger tendency to correct female data points. Thus, the bias is corrected in this process. We verify this by the finding that of all the data points that are corrected from high income to low income by our post-processing model, about 91% are male data points. For the data points that are corrected from low income to high income, the percentage of male data points dropped to 72%. This shows that the model has a stronger tendency to correct male data points with high income, and female data points with low income.

6 Conclusion

Fairness is an increasingly important field in real-life scenarios, especially when machine learning models are deployed in more areas. In this work, with the Adult Income Dataset, we find that

significant bias problems. These biases not only come from the sensitive attributes (such as Sex, Race, etc.), but also come from other attributes that could infer those sensitive attributes. The harder part is that the bias occurs in the training data themselves. If we make changes, we might greatly harm the accuracy as well. A very delicate and clean way need to be devised.

In this work, we provide a novel approach in reducing the bias problem in the dataset. We utilized several intuitions in this work, the most important intuition is that bias patterns could be detected and machine learning models could also learn this pattern. Thus, our post-correction model mainly functions to identify bias and correct the predictions made by the original algorithm. From our empirical results, the model could reduce bias by 80% and only hurts accuracy within 1%. This finding is novel and very intuitive. Previous approaches mainly use statistical methods and require revealing of ground truth labels. Our approach is entirely black-box and very efficient, which can be easily deployed in any real-life scenarios.

Specifically, our approach involves training a post-correction model using random forest. The post-correction learn from the wrongly predicted data points in the original algorithm. During testing time, the post-correction model looks at all the data points and decides of which predictions to be flipped. Eventually, after this flipping process, we get the eventual predictions. The eventual prediction may sacrifice some accuracy, but will be greatly enhanced in fairness.

Our model is novel in several ways. First, our model does not require using ground truth labels during testing time, and the post-correction model utilized a current commonly-used machine learning model random forest, which make it much easier to be used. Second, from the best of our knowledge, this is the first work that used purely machine learning model as a post-process mechanism. Previous approaches are majorly statistical based. Third, our finding has validated an important hypothesis, that bias patterns is learn-able. In our experiments, through comparison with baselines models that do random flipping, we could easily see that our model flip the much more data points that are indeed biased, which results in fairer predictions.

We believe that this work has pointed out to numerous future directions. For example, some other models may be found that fits better in the post-correction process, or our model could be fine-tuned to fit better. In our work we only work on the overall structure and do not focus on model fine-tuning. Second, there could be more understanding on the post-correction model, in this work, intuitive conclusions are made based on empirical results, which could be further investigated. Third, as our model is efficient and easy to use, if we find ways to further enhance its stability, we could also work on to deploy it in real-world scenarios.

7 Acknowledgements

I would like to thank to Professor Siyao Guo, Professor Gus Xia and Professor He He who all have provided support for this project. These advices have been crucial and insightful. Special thanks should be given to Professor Siyao Guo who has generously helped me throughout my undergraduate years and this project.

I also extend my thanks to my parents and friends who have supported me throughout my study.

References

[1] D. Pessach and E. Shmueli, "Algorithmic fairness," *arXiv preprint arXiv:2001.09784*, 2020.

[2] S. M. Julia Angwin, Jeff Larson and L. Kirchner. (2016) Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[3] M. O. Prates, P. H. Avelar, and L. C. Lamb, "Assessing gender bias in machine translation: a case study with google translate," *Neural Computing and Applications*, pp. 1–19, 2019.

[4] S. Caton and C. Haas, "Fairness in machine learning: A survey," *arXiv preprint arXiv:2010.04053*, 2020.

[5] J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan, "Algorithmic fairness," in *Aea papers and proceedings*, vol. 108, 2018, pp. 22–27.

[6] E. Hamilton, "Benchmarking four approaches to fairness-aware machine learning," Ph.D. dissertation, 2017.

[7] B. A. Williams, C. F. Brooks, and Y. Shmargad, "How algorithms discriminate based on data they lack: Challenges, solutions, and policy implications," *Journal of Information Policy*, vol. 8, pp. 78–115, 2018.

[8] J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," *Science advances*, vol. 4, no. 1, p. eaao5580, 2018.

[9] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[10] 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.

[11] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[12] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.

[13] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.

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

[15] T. Miconi, "The impossibility of" fairness": a generalized impossibility result for decisions," *arXiv preprint arXiv:1707.01195*, 2017.

[16] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," *arXiv preprint arXiv:1709.02012*, 2017.

[17] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3995–4004.

[18] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 35–50.

[19] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 869–874.

[20] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, 2017, pp. 797–806.

[21] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, "Learning non-discriminatory predictors," in *Conference on Learning Theory*. PMLR, 2017, pp. 1920–1953.

[22] B. T. Luong, S. Ruggieri, and F. Turini, "k-nn as an implementation of situation testing for discrimination discovery and prevention," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 502–510.

[23] N. Grgić-Hlača, M. B. Zafar, K. P. Gummadi, and A. Weller, "On fairness, diversity and randomness in algorithmic decision making," *arXiv preprint arXiv:1706.10208*, 2017.

[24] P. K. Lohia, K. N. Ramamurthy, M. Bhide, D. Saha, K. R. Varshney, and R. Puri, "Bias mitigation post-processing for individual and group fairness," in *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2019, pp. 2847–2851.

[25] M. P. Kim, A. Ghorbani, and J. Zou, "Multiaccuracy: Black-box post-processing for fairness in classification," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 247–254.

[26] P. Saleiro, B. Kuester, A. Stevens, A. Anisfeld, L. Hinkson, J. London, and R. Ghani, "Aequitas: A bias and fairness audit toolkit," *arXiv preprint arXiv:1811.05577*, 2018.

[27] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[29] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[30] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml