Increasing Fairness in Predictions Using Bias Parity Score Based Loss Function Regularization

Bhanu Jain, Manfred Huber, Ramez Elmasri
Department of Computer Science and Engineering, The University of Texas at Arlington, TX
bhanu.jain@uta.edu, {huber.elmasri}@cse.uta.edu

Abstract
Increasing utilization of machine learning based decision support systems emphasizes the need for resulting predictions to be both accurate and fair to all stakeholders. In this work we present a novel approach to increase a Neural Network model’s fairness during training. We introduce a family of fairness enhancing regularization components that we use in conjunction with the traditional binary-cross-entropy based accuracy loss. These loss functions are based on Bias Parity Score (BPS), a score that helps quantify bias in the models with a single number. In the current work we investigate the behavior and effect of these regularization components on bias. We deploy them in the context of a recidivism prediction task as well as on a census-based adult income dataset. The results demonstrate that with a good choice of fairness loss function we can reduce the trained model’s bias without deteriorating accuracy even in unbalanced datasets.

Introduction
The use of automated decision support and decision-making systems (ADM) (Hardt, Price, and Srebro 2016) in applications with direct impact on people’s lives has increasingly become a fact of life, e.g. in criminal justice (Kleinberg, Mullainathan, and Raghavan 2016; Jain et al. 2020b; Dressel and Farid 2018), medical diagnosis (Kleinberg, Mullainathan, and Raghavan 2016; Ahsen, Ayvaci, and Raghunathan 2019), insurance (Baudry and Robert 2019), credit card fraud detection (Dal Pozzolo et al. 2014), electronic health record data (Gianfrancesco et al. 2018), credit scoring (Huang, Chen, and Wang 2007) and many more diverse domains. This, in turn, has led to an urgent need for study and scrutiny of the bias-magnifying effects of machine learning and Artificial Intelligence algorithms. Appropriately, much research is being done currently to mitigate bias in AI-based decision support systems (Ahsen, Ayvaci, and Raghunathan 2016; Kleinberg, Mullainathan, and Raghvan 2016; Noriega-Campero et al. 2019; Feldman 2015; Oneto, Donini, and Pontil 2020; Zemel et al. 2013).

Bias in Decision Support Systems. As our increasingly digitized world generates and collects more data, decision makers are increasingly using AI based decision support systems. With this, the need to keep the decisions of these systems fair for people of diverse backgrounds becomes essential. Groups of interest are often characterized by sensitive attributes such as race, gender, affluence level, weight, and age to name a few. While machine learning based decision support systems do often consider these attributes explicitly, biases in the data sets, coupled with the used performance measures can nevertheless lead to significant discrepancies in the system’s decisions. For example, as many minorities have traditionally not participated in many domains such as loans, education, employment in high paying jobs, receipt of health care, resulting datasets are often highly unbalanced. Similarly, some domains like homeland security, refugee status determination, incarceration, parole, loan repayment etc., may be already riddled with bias against certain subpopulations. Thus, human bias infiltrates the datasets used for AI based prediction systems which further amplifies it. This mandates reducing bias in AI datasets.

Contributions. We propose to use Bias Parity Score (BPS) measures to characterize fairness and develop a family of corresponding loss function regularizers for Neural Networks to enhance fairness. The goal is to permit the system to actively pursue fair solutions during training while maintaining as high a performance as possible. We apply the approach in the context of several fairness measures and investigate multiple loss function formulations to study the performance. In these experiments we show that, if used with appropriate settings, the technique measurably reduces race-based bias in recidivism prediction, and demonstrate on the gender-based Adult Income dataset that it can outperform state-of-the-art techniques even if they are targeted more directly on the aspect of bias and fairness measured.

Notation
In this paper we present the proposed approach largely in the context of class prediction tasks with the subpopulations indicated by a binary sensitive attribute. However, the approach can be extended to multi-class sensitive attributes as well as to regression tasks. For the description we use the following notation to describe the classification problem and to capture group performance and prediction bias.

Each element of the dataset, \(D = \{(X, A, Y)\}_i\), is represented by an attribute vector \((X, A)\), where \(X \in \mathbb{R}^2\) represents its features and \(A \in \{0, 1\}\) is the sensitive attribute. \(C = c(X, A) \in \{0, 1\}\) indicates the predicted value and \(Y \in \{0, 1\}\) is the true value of the target variable.

To obtain fairness characteristics, we use parity over statistical performance measures, \(m\), where \(m(A = a)\) indicates the measure for the subpopulation where \(A = a\).
Since this paper will introduce the proposed framework in the context of binary classification, the fairness related performance measures used will center around elements of the confusion matrix. In the notation used here, these metrics and the corresponding measures are:

**False positive rate (FPR):** \( P(C = 1 \mid Y = 0) \).
\[ m_{FPR}(A = a) = P(C = 1 \mid Y = 0, A = a) \]

**False negative rate (FNR):** \( P(C = 0 \mid Y = 1) \).
\[ m_{FNR}(A = a) = P(C = 0 \mid Y = 1, A = a) \]

**True positive rate (TPR):** \( P(C = 1 \mid Y = 1) \).
\[ m_{TPR}(A = a) = P(C = 1 \mid Y = 1, A = a) \]

**True negative rate (TNR):** \( P(C = 0 \mid Y = 0) \).
\[ m_{TNR}(A = a) = P(C = 0 \mid Y = 0, A = a) \]

Strict parity in these or similar measures implies that they are identical for both groups, i.e. \( m(A = 0) = m(A = 1) \).

### Related Work

Some recent works (Ntoutsi et al. 2020; Krasanakis et al. 2018) summarize bias mitigation techniques into three categories: i) preprocessing input data (Calders, Kamiran, and Pechenizky 2009; Feldman et al. 2015; Feldman 2015), ii) in-processing or training under fairness constraints that focuses on the algorithm (Celis et al. 2019; Zafar et al. 2017; Iosifidis and Ntoutsi 2020) and, iii) postprocessing that seeks to improve the model (Hardt, Price, and Srebro 2016; Mishler, Kennedy, and Chouldechova 2021). Along similar lines, (Orphanou et al. 2021) sums up mitigating algorithmic bias as detection of bias, fairness management, and explainability management. Their “fairness management” is the same as the aforementioned “bias mitigation techniques” but they added “Auditing” (Fairness formalization).

The first bias mitigation technique involving preprocessing data is built on the premise that disparate impact in training data results in disparate impact in the classifier. Therefore, these techniques are comprised of massaging data labels and reweighting tuples. (Calders, Kamiran, and Pechenizky 2009) asserts that these massaging and reweighting techniques yield a classifier that is less biased than without such a process. It also notes that while massaging labels is intrusive and can have legal implications (Barocas and Selbst 2016), reweighting does not have these drawbacks.

The second technique of in-processing fairness under constraint (Celis et al. 2019; Zafar et al. 2017) chooses impact metrics (Iosifidis and Ntoutsi 2020; Zafar et al. 2017; Calders et al. 2013), followed by modifying the imposed constraints during classifier training (Zafar et al. 2017; Calders et al. 2013; Iosifidis and Ntoutsi 2020; Oneto, Donini, and Pontil 2020). Similarly, (Iosifidis and Ntoutsi 2020) changes the training data distribution and monitors the discriminatory behavior of the learner to adjust the decision boundary to prevent discriminatory learning. (Islam, Pan, and Foulds 2021) studied the behavior of fair learning algorithms to improve fairness without loss in performance and showed promising results via hyper-parameter selection.

The third technique involves postprocessing to obey fairness constraints. Given the target and the sensitive attribute, (Hardt, Price, and Srebro 2016) shows how to adjust the predictor to eliminate discrimination as per their definition.

Our work uses supervised learning without balancing the dataset and falls under in-processing fairness under constraints. We use the original, potentially skewed data to avoid issues from massaging labels and potential complications arising in the context of overlapping sensitive attributes. Some work employing unsupervised clustering also seek to maintain the original balance in the dataset (Abbasi, Bhaskara, and Venkatasubramanian 2021; Abraham, Sundaram, and others 2019; Chierichetti et al. 2018).

### Definitions of Fairness

Several concepts of fairness require one or more demographic or statistical properties to be constant across subpopulations. Demographic parity (or statistical parity) mandates that decision rates are independent of each sensitive attribute (Noriega-Campero et al. 2019; Louizos et al. 2015; Calders, Kamiran, and Pechenizky 2009; Zafar et al. 2015). For binary classification this implies \( P(C = 1 | A = 0) = P(C = 1 | A = 1) \). This, however, makes an equality assumption between subpopulations which might not hold. To address this, (Hardt, Price, and Srebro 2016) focus on error rate balance where fairness requires subpopulations to have equal FPR, FNR, or both. Another common condition is equality of odds, demanding equal TPR and TNR. While perfect parity would be desirable, it often is not achievable and thus measures of the degree of parity have to be used (Jain et al. 2020a). Refer to (Mehrabi et al. 2019; Chouldechova and Roth 2018; Verma and Rubin 2018) for a more complete survey of computational fairness metrics.

Here we use a quantitative parity measure, Bias Parity Score (BPS), as our concept of fairness. This, applied to metrics including FPR, FNR, TPR, TNR, and prediction rate, allows us to approximate a number of different fairness criteria, including Demographic parity and error rate balance.

As in most work, we use only one sensitive attribute at a time in the experiments. (Foulds et al. 2020), by contrast explicitly considers a framework of intersectionality. To address similar scenarios, we briefly discuss how the BPS measure can be expanded to multi-valued attributes.

### Approach

This paper provides a framework for deep learning systems to learn fairer models for known sensitive attributes. It uses a quantitative measure of fairness, namely Bias Parity Score (BPS) which evaluates the degree to which a common measure in the subpopulations is the same. Based on this we then derive a family of corresponding differentiable loss functions as regularizers to the original task performance.

### Bias Parity Score (BPS)

Prediction bias is the differential treatment of “similarly situated” individuals. It manifests itself in unfairly benefiting some groups (Jiang and Nachum 2020; Angwin et al. 2016; Dressel and Farid 2018; Zeng, Ustun, and Rudin 2017; Jain et al. 2019; 2020b; Ozkan 2017). Bias in recidivism, for example, may be observed in a higher FPR and lower FNR for one subgroup, implying more frequent incorrect predictions to reoffend and thus denied parole.

To capture this, the relevant property can be encoded as a measure and parity between subgroups can be defined as
equality for this measure. In this work we capture the similarity of the measures in the form of Bias Parity Score (BPS). Given that \( m_s(A = 0) \) and \( m_s(A = 1) \) are the values of a statistical measure, \( s \), for the sensitive and non-sensitive subpopulations, respectively, BPS can be computed as:

\[
BPS_s = 100 \frac{\min(m_s(A = 1), m_s(A = 0))}{\max(m_s(A = 1), m_s(A = 0))}. \tag{1}
\]

where the multiplication factor of 100 leads to a percentage measure, making it easier to read. A BPS of 100 represents perfect parity while a BPS of 0 represents maximal bias. This formulation yields a symmetric measure for two groups that can be applied to any property across subgroups.

We here defined BPS in terms of a binary sensitive attribute. While this is the case we will study in this paper, this fairness measure can easily be expanded to a multi-valued sensitive attribute \( A \in \{a_1, \ldots, a_k\} \) in a form such as:

\[
BPS_s = \frac{100}{k} \frac{\min(m_s(A = a_1), m_s(A = a_2), \ldots, m_s(A = a_k))}{\max(m_s(A = a_1), m_s(A = a_2), \ldots, m_s(A = a_k))} \tag{2}
\]

where \( m_s() \) is the value of the underlying measure for the entire population. BPS here thus measures fairness as the average bias of all classes compared to the population.

Even though, BPS can be used for many statistical measures (Jain et al. 2020a), we will evaluate it with FPR, FNR, TPR and TNR, and in the Adult Income domain with prediction rate to facilitate comparisons with previous approaches.

**BPS-Based Fairness Loss Functions**

While the BPS score represents a measure of fairness, it does not lend itself directly to training a deep learning system since it is not generally differentiable. To address this, we need to translate the underlying measure into a differentiable form and combine it into a differentiable version of the BPS score. For FPR, FNR, TPR, and TNR we first have to define continuous approximations. For this, we build our Neural Network classifiers with a logistic (or Softmax) output, leading, when trained for accuracy using binary cross-entropy, to the output, \( y \), being interpretable as the probability of the positive class, \( y = P(Y = 1 | X, A) \). Using this continuous output, we can define a continuous measure approximation \( m_c() \) for FPR, FNR, TPR, and TNR:

\[
\begin{align*}
mc_{\text{FPR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 0} y_t}{\sum_{(X,Y), A_k = k, Y = 0} y_t + \sum_{(X,Y), A_k = k, Y = 1} (1 - y_t)} \\
mc_{\text{FNR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 1} (1 - y_t)}{\sum_{(X,Y), A_k = k, Y = 0} y_t + \sum_{(X,Y), A_k = k, Y = 1} (1 - y_t)} \\
mc_{\text{TPR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 1} y_t}{\sum_{(X,Y), A_k = k, Y = 0} y_t + \sum_{(X,Y), A_k = k, Y = 1} (1 - y_t)} \\
mc_{\text{TNR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 0} y_t}{\sum_{(X,Y), A_k = k, Y = 0} y_t + \sum_{(X,Y), A_k = k, Y = 1} (1 - y_t)}
\end{align*}
\]

This is not equal to \( m_s() \) as it is sensitive to the probability. To reduce this discrepancy, a second approximation, \( ms_s() \), uses a sigmoid function, \( S(x) = \frac{1}{1 + e^{-x}} \), to reduce the impact of intermediate prediction probabilities:

\[
\begin{align*}
ms_{\text{FPR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 0} S(y_t - 0.5)}{\sum_{(X,Y), A_k = k, Y = 0} S(y_t - 0.5) + \sum_{(X,Y), A_k = k, Y = 1} S(0.5 - y_t)} \\
ms_{\text{FNR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 1} S(0.5 - y_t)}{\sum_{(X,Y), A_k = k, Y = 0} S(y_t - 0.5) + \sum_{(X,Y), A_k = k, Y = 1} S(0.5 - y_t)} \\
ms_{\text{TPR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 1} S(y_t - 0.5)}{\sum_{(X,Y), A_k = k, Y = 0} S(y_t - 0.5) + \sum_{(X,Y), A_k = k, Y = 1} S(0.5 - y_t)} \\
ms_{\text{TNR}}(A = k) &= \frac{\sum_{(X,Y), A_k = k, Y = 0} S(0.5 - y_t)}{\sum_{(X,Y), A_k = k, Y = 0} S(y_t - 0.5) + \sum_{(X,Y), A_k = k, Y = 1} S(0.5 - y_t)}
\end{align*}
\]

Once measures are defined, a continuous approximation for BPS fairness for both measures can be defined as:

\[
\begin{align*}
BPS_{cs} &= \min(m_c(A = 1), m_c(A = 0)) \\
BPS_{cs} &= \max(m_s(A = 1), m_s(A = 0)) \\
\end{align*}
\]

These, can be inverted into loss functions that can be used during training and further expanded by allowing to weigh small versus large biases by raising it to the \( k^{th} \) power.

\[
\begin{align*}
LFC(s, k) &= (1 - BPS_{cs})^k \\
LFS(s, k) &= (1 - BPS_{cs})^k \\
\end{align*}
\]

These loss functions are continuous and differentiable in all but one point, namely where numerator and denominator are equal. This, however, can be easily addressed.

**Fairness Regularization for NN Training**

The approach is aimed at training Neural Network classifiers to obtain more fair results while preserving accuracy. The latter task is commonly encoded using a binary cross-entropy loss, \( L_{BCE} \). Starting from this, we use the fairness loss as regularization terms resulting in loss functions \( LFC \) and \( LFS \) for continuous and sigmoided fairness losses:

\[
\begin{align*}
LFC(\alpha, \tilde{k}) &= L_{BCE} + \sum_{s,k} \alpha_i LFC(s, k_i) \\
LFS(\alpha, \tilde{k}) &= L_{BCE} + \sum_{s,k} \alpha_i LFS(s, k_i)
\end{align*}
\]

where \( \alpha_i \) is a weight vector and \( k \) is a vector of powers for each of the fairness losses, and \( s \) is the vector of loss metrics, \( <FPR, FNR, TPR, TNR> \). Setting an \( \alpha_i \) to 0 effectively removes the corresponding fairness criterion.

During Neural Network classifier training, different values for \( \alpha, \tilde{k} \), and the choice of sigmoided vs continuous loss modulate the underlying fairness characteristics.

Figure 1 shows an overview of the basic model selection process for the proposed approach. Based on selected fairness criteria and hyperparameter ranges, an architecture is trained in a grid search and the best model is selected.

![Figure 1: Fair classifier design based on fairness criteria, loss function, hyperparameter range, and network architecture.](image-url)
Experiments

To study the applicability of the proposed use of fairness losses as regularization terms, we conducted experiments in the recidivism and income prediction domains.

Recidivism Data: This uses data from “Criminal Recidivism in a Large Cohort of Offenders Released from Prison in Florida, 2004-2008 (ICPSR 27781)” (United States Department of Justice, Office of Justice Programs, Bureau of Justice Statistics 2010) (Bhati and Roman 2014). It contains 156,702 records with a 41:59 recidivist to non-recidivist ratio. This ratio for our two race-based subpopulations is 34:66 for Caucasians and 46:54 for African Americans, making it very unbalanced and leading to significant bias. The data covers six crime categories and provides demographic features including crime committed, age, time served, gender, etc. We employed one-hot encoding for categorical features and trained to predict recidivism within 3 years.

Income Data: This uses the common “Adult Income Data Set” (Kohavi and others 1996) from the UCI repository (Dua and Graff 2017) that was extracted from 1994 Census data. The dataset has 48,842 records, each representing an individual over 16 earning more than $100. The dataset holds socioeconomic status, education, and job information and is thus largely demographic. Gender is the sensitive attribute used here with the goal of predicting a salary higher than $50 K. This dataset helped verify effects in other domains and permitted comparisons with state-of-the-art techniques.

Performance on Recidivism Data

Constructing Neural Networks For the recidivism data we trained networks with 2 hidden layers with 41 units each and a logistic output unit. Each hidden layer used ReLU activation and 10% dropout (Srivastava et al. 2014), followed by Batch Normalization (Santurkar et al. 2018). We tuned hyperparameters to select a batch size of 256 and 100 epochs and trained using the Adam (Kingma and Ba 2014) optimizer.

Evaluation Study To evaluate fairness regularization, we conducted experiments with 6 measures for continuous and sigmoided loss, employed 4 different exponents for the continuous case, and ran experiments for 10 different weight settings. We used FPR, FNR, TPR, and TNR individually as well as FPR and FNR or TPR and TNR simultaneously with equal weights. A grid search varied weights $\alpha$ between 0.1 and 1, and for continuous loss used powers of 1 to 4. The goal was to compare the effects on system behavior in terms of accuracy, fairness, and stability. We used Monte Carlo cross validation with 10 iterations and reported the means.

Results and Discussion We investigated the effect of the loss functions on residual bias measured by Bias Parity Score for FPR, FNR, TPR, TNR, and Accuracy. In addition, we recorded Accuracy as well as the values for BCE loss and for each fairness loss used in the respective experiment.

Applicability and Effect of Regularization Weight: We first analyzed the experiments for the six continuous fairness measures in the linear case. Figure 2 shows the average accuracy (solid green line), BPS scores (solid lines), and loss function values (corresponding dashed lines) as a function of the regularization weight, $\alpha$. Only results for $FNR$, $FPR$, and $FNR+FPR$-based regularization are shown. Behavior for $TNR$, $TPR$, and $TNR+TPR$ was similar.

These graphs show that introducing fairness regularization immediately increases the corresponding BPS score, and thus fairness, while only gradually decreasing accuracy. This demonstrates the viability of the technique.

However, the experiments also indicate considerations for regularization weights. In particular, while the experiments with $LF_c(FNR,1)$ and $LF_c(FNR,1) + LF_c(FPR,1)$ show a relatively steady increase in fairness, the case of $LF_c(FPR,1)$ shows that after an initial strong increase, the active fairness measure, $BPS_{FPR}$, starts to decrease once $\alpha$ exceeds 0.2. At the same time the regularization loss, $LF_c(FPR,1)$, continues to decrease, showing a decoupling between the core fairness measure and the loss function.

Sigmoided Loss Function: One way to address this decoupling are the proposed sigmoided fairness loss functions. Figure 3 shows the corresponding results using the sigmoided version of the fairness loss $LF_s(FNR,1)$, $LF_s(FPR,1)$, and $LF_s(FNR,1) + LF_s(FPR,1)$. Again, BPS values, Accuracy, and loss function values are shown.

While these results as expected show less decoupling between loss and fairness, indicate higher levels of fairness earlier, and maintain fairness more reliably, they also show a stronger degradation in accuracy and, when looking at regularization loss for higher weights and corresponding variances, also exhibit strong signs of instabilities. This implies that while there are advantages, optimizing them is significantly harder, leading to less stable convergence.

Effect of Loss Function Power: Another way to modify the effect of regularization losses is to increase the loss function power. This reduces the impact of small amounts of bias near a fair solution while increasing importance if fairness losses are high. The goal is to reduce the small scale adjustments near a solution most responsible for decoupling. Figure 4 shows the effect of different powers for continuous loss $LF_c(FPR,k)$, the case where strong decoupling occurred.

These graphs show that as the power increases, improvement in fairness becomes smoother and decoupling of loss and fairness occurs later. However, a larger weight is needed to optimize, with power 4 not reaching the best fairness.

Performance on Adult Income Data

To show the applicability beyond recidivism and to compare with state of the art approaches, we applied it to the commonly used Adult Income Dataset. As existing works focus on different fairness measures, this paper separately compares against works that used pRule and Accuracy to measure fairness and performance (Du et al. 2021; Krasanakis et al. 2018; Zafar et al. 2017; Zhang, Lemoine, and Mitchell 2018; Kamishima et al. 2012), against works that use Discrimination Score ($DS$) (Calders and Verwer 2010; Rajabi and Garibay 2021) and accuracy, and against works that used absolute values of FPR and FNR for the two subpopulations (Beutel et al. 2017; Zhang, Lemoine, and Mitchell 2018).

For these comparisons, an appropriate measure for the BPS score is needed. While FPR and FNR together can be used directly for the last group, a different metric is required...
Figure 2: BPS Measures, Accuracy, and Loss function values as a function of regularization weight, $\alpha$, for the continuous, linear case using $LF_c(FPR,1)$ (left), $LF_c(FNR,1)$ (middle), and $LF_c(FPR,1) + LF_c(FNR,1)$ (right).

Figure 3: BPS Measures, Accuracy, and Loss function values as a function of regularization weight, $\alpha$, for sigmoided loss functions $LF_s(FPR,1)$ (left), $LF_s(FNR,1)$ (middle), and $LF_s(FPR,1) + LF_s(FNR,1)$ (right).

for the first two since pRule and Discrimination Score are related to statistic parity and based on prediction rate, $STP = P(C = 1|X)$, with pRule being the ratio between subpopulations while Discrimination Score models the positive difference, $DS = |P(C = 1|A = 1) - P(C = 1|A = 0)|$. To address these definitions, positivity rate $STP$ is used here as the underlying measure for the BPS based regularization.

Neural Network Architectures We experimented with hyperparameters to find the network with the highest baseline accuracy, leading to a network with two leaky ReLU hidden layers with 108 and 324 neurons, respectively, that yielded a 84.75% baseline accuracy. For this network, a grid search similar to the recidivism experiments was used to obtain the best loss function parameters for both $LF_c(STP,k)$ and the combined $LF_c(FPR,k)$ and $LF_c(FNR,k)$.

Results and Comparison Varying loss function parameters yielded similar observations as shown in Figure 5 for $BPS_{STP}$. As previously, increasing the power reduced decoupling. Studies with sigmoided loss similarly demonstrated the same behavior as for the recidivism data, demonstrating the framework’s generality. The best fairness results for $BPS_{STP}$ were achieved for $\alpha = 0.84$ and $k = 4$, while the best results for the FPR and FNR combination resulted from $\alpha_1 = 0.1$, $\alpha_2 = 0.125$ and $k = 3$. These hyperparameters were used in the subsequent comparisons.

Comparison with State of The Art Results: We compared our approach to a number of recent techniques that used one or more fairness criteria and performance metrics.

As shown in this table, we achieved the highest accuracy while maintaining a pRule of approximately 100%.

To demonstrate the versatility of the approach, we also compared against two techniques that use a different fairness measure in the form of Discrimination Score. Here we compare to (Gong, Liu, and Liu 2021) which also utilizes fairness regularization and to (Rajabi and Garibay 2021) who utilize a Generative Adversarial Network in a pre-processing approach to generate fair data. The results are in Table 2.

As per the results our approach outperforms others.

Finally, in Table 3, we compare results of our approach with those that used equality of absolute values of FPR and FNR. (Beutel et al. 2017) uses an adversarial training technique to eliminate the sensitive attribute information.
Figure 4: BPS Measures, Accuracy, and Loss function values as a function of regularization weight, $\alpha$, for different powers of the continuous loss functions $LF_{c(FPR,1)}$ (left), $LF_{c(FPR,2)}$ (center-left), $LF_{c(FPR,3)}$ (center-right), and $LF_{c(FPR,4)}$ (right).

Figure 5: BPS Measures, Accuracy, and Loss function values as a function of regularization weight, $\alpha$, for Architecture 2 on Adult income data for $LF_{c(STP,1)}$ (left), $LF_{c(STP,2)}$ (right), $LF_{c(STP,3)}$ (eft), and $LF_{c(STP,4)}$ (right).

Table 2: Adult Income Discrimination Score Comparison

<table>
<thead>
<tr>
<th>Fairness Technique</th>
<th>DS pRule acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Gong, Liu, and Liu 2021)</td>
<td>0.03 - 76%</td>
</tr>
<tr>
<td>(Rajabi and Garibay 2021)</td>
<td>0.082 - 77%</td>
</tr>
<tr>
<td><strong>Current work (0.84 $\ast$ $LF_{c(STM,4)}$)</strong></td>
<td><strong>0.001 100% 83%</strong></td>
</tr>
</tbody>
</table>

Table 3: Adult Income FPR and FNR Equality Comparison without and with fairness correction

<table>
<thead>
<tr>
<th></th>
<th>FPR</th>
<th>FNR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Beutel 17</strong></td>
<td>0.1200</td>
<td>0.1778</td>
</tr>
<tr>
<td></td>
<td>0.1200</td>
<td>0.1778</td>
</tr>
<tr>
<td><strong>Zhang 18</strong></td>
<td>0.0017</td>
<td>0.0081</td>
</tr>
<tr>
<td></td>
<td>0.0017</td>
<td>0.0081</td>
</tr>
<tr>
<td><strong>Current work</strong></td>
<td>0.0319</td>
<td>0.0610</td>
</tr>
<tr>
<td></td>
<td>0.0319</td>
<td>0.0610</td>
</tr>
</tbody>
</table>

The FPR and FNR using our technique were approximately equal across genders at 0.061 versus 0.0708 and 0.4785 versus 0.4862, respectively, corresponding to $BPS_{FPR}$ and $BPS_{FNR}$ of 93.3 and 98.4, which outperformed the other techniques on the fairness criteria. While the lowest FNR and FPR scores (indicated in bold for each of the scores and genders) were achieved by either baseline models or other approaches, these were not able to balance them across genders and resulted in models with higher bias.

All of these results underline the flexibility and power of the use of proposed BPS-based regularization in Deep Neural Networks to reduce bias while maintaining high performance. In particular, they demonstrate that with the right choice of hyper-parameters it can match or even outperform state of the art techniques in a wide range of domains and in the context of different fairness and performance criteria, even if those techniques are more specifically tailored to the used criteria. The current approach can thus provide a high-performance tool to address fairness in prediction tasks.

**Conclusions**

We proposed Bias Parity Score-based fairness metrics and an approach to translate them into corresponding loss functions to be used as regularization terms while training deep Neural Networks to actively improve fairness between sub-populations. For this, we introduced a family of fairness loss functions and conducted experiments on recidivism prediction to investigate the applicability and behavior of the approach for different hyperparameters. We demonstrated that this work does not depend on changing input or output labels to make fair recommendations while not forsaking accuracy.

Additional comparative experiment results from the Adult Income dataset with specialized state-of-the-art approaches showed that the our regularization framework is flexible and with an appropriate BPS-based fairness loss function can compete with and outperform these methods.
References


Dua, D., and Graff, C. 2017. UCI machine learning repository.


