

Article

How Optimal Transport Can Tackle Gender Biases in Multi-Class Neural Network Classifiers for Job Recommendations

Fanny Jourdan ^{1,2} , Titon Tshiongo Kaninku ^{1,3}, Nicholas Asher ² , Jean-Michel Loubes ¹ 
and Laurent Risser ^{1,*} 

¹ CNRS, Institut de Mathématiques de Toulouse, Université de Toulouse, CEDEX 9, F-31062 Toulouse, France

² CNRS, Institut de Recherche en Informatique de Toulouse, Université de Toulouse, CEDEX 9, F-31062 Toulouse, France

³ AKKODIS Group-Hauts de France, Marcq-en-Baroeul, F-59700 Lille, France

* Correspondence: laurent.risser@math.univ-toulouse.fr

† Current Address: Institut de Mathématiques de Toulouse, Université Paul Sabatier, 118 Route de Narbonne, CEDEX 9, F-31062 Toulouse, France.

Abstract: Automatic recommendation systems based on deep neural networks have become extremely popular during the last decade. Some of these systems can, however, be used in applications that are ranked as High Risk by the European Commission in the AI act—for instance, online job candidate recommendations. When used in the European Union, commercial AI systems in such applications will be required to have proper statistical properties with regard to the potential discrimination they could engender. This motivated our contribution. We present a novel optimal transport strategy to mitigate undesirable algorithmic biases in multi-class neural network classification. Our strategy is model agnostic and can be used on any multi-class classification neural network model. To anticipate the certification of recommendation systems using textual data, we used it on the Bios dataset, for which the learning task consists of predicting the occupation of female and male individuals, based on their LinkedIn biography. The results showed that our approach can reduce undesired algorithmic biases in this context to lower levels than a standard strategy.



Citation: Jourdan, F.; Kaninku, T.T.; Asher, N.; Loubes, J.-M.; Risser, L. How Optimal Transport Can Tackle Gender Biases in Multi-Class Neural Network Classifiers for Job Recommendations. *Algorithms* **2023**, *16*, 174. <https://doi.org/10.3390/a16030174>

Academic Editor: Frank Werner

Received: 17 February 2023

Revised: 14 March 2023

Accepted: 17 March 2023

Published: 22 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: fairness; algorithmic bias; neural networks; NLP; recommender systems; multi-class classification; certification

1. Introduction

The field of Artificial Intelligence (AI) has experienced remarkable growth over the past decade, particularly in Natural Language Processing (NLP). Current state-of-the-art NLP applications, such as translation or text-based recommendations, rely heavily on Deep Neural Networks (DNNs), which use transformer architectures [1]. Transformer architectures are composed of several blocks that each contain an attention sublayer and a feed-forward sublayer. The two most-widely used transformer neural network architectures for these tasks are Bidirectional Encoder Representations from Transformers (BERT) [2] and Generative Pre-trained Transformer (GPT) [3]. There are numerous variants of these models. Compared with their predecessors such as LSTM models [4], they exhibit significantly higher performance in NLP applications. However, due to the large number of parameters and non-linearities involved, they are even less interpretable than more classic models and are typically just treated as black box decision systems. We developed the methodology in this paper with the aim of controlling undesirable algorithmic biases in recommendation systems that exploit textual information from personal profiles on social networks, such as job or housing offers. Throughout this paper, we emphasise the importance of ethical considerations in the development and deployment of these applications, as they can significantly impact users' lives.

For a long time, many believed that machine learning algorithms could not be discriminatory since they lack human emotions. This view is, however, outdated now, as different studies have shown that an algorithm can learn and even amplify biases from a biased dataset [5]. In this paper, we use the term *algorithmic biases* to refer to automatic decisions made by a machine learning algorithm that are not neutral, fair, or equitable for a particular subgroup of people (or statistical observations in general). This group is distinguished by a *sensitive variable*, such as gender, age, or ethnic origin. The field of study and prevention of these specific algorithmic biases is called *fair learning*. Ensuring fairness is essential to ensure an ethical application of algorithms in society. Ethical concerns have become increasingly important in recent years, and the deployment of a discriminatory algorithm is no longer acceptable. Many regulations already address ethical issues related to AI. In the area of privacy, the General Data Protection Regulation (GDPR), adopted by the European Parliament in 2016, allows, for instance, the French Commission on Informatics and Liberty (NCIL) and other independent administrative authorities in France to impose severe penalties on companies that do not manage customer data transparently [6]. The GDPR is an example of how public authorities are progressively developing legal frameworks and taking actions to mitigate threats. More recently, the so-called *AI act* (<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022R2065&qid=1679413388765> accessed on 14 March 2023) of the European Commission defined a list of *High Risk* applications of AI, most of them being related to a strong impact on human life. For instance, job candidate recommendation systems are ranked as *High Risk*. Importantly, when sold in or from the European Union, such AI systems will need to have appropriate statistical properties with respect to any potential discrimination they may cause (see Articles 9.7, 10.2, 10.3, and 71.3).

Motivated by the future certification of AI systems based on black box neural networks against discrimination, our article expands on the work of [7] to address algorithmic biases observed in NLP-based multi-class classification. The main methodological novelties of this paper are: the extension of [7] to multi-class classification and a demonstration of how to apply it to NLP data in an application ranked as *High Risk* by the *AI act*. The bias mitigation model proposed in this paper involves incorporating a regularisation term, in addition to a standard loss, when optimising the parameters of a neural network. This regularisation term mitigates algorithmic bias by enforcing the similarity of prediction or error distributions for two groups distinguished by a predefined binary sensitive variable (e.g., males and females), measured using the 2-Wasserstein distance between the distributions. Note that [7] is the first paper that demonstrated how to calculate pseudo-gradients of this distance in a mini-batch context, enabling the use of this method to train deep neural networks with reduced algorithmic bias.

To extend [7] to multi-class classification with deep neural networks, we need to address a key problem: estimating the 2-Wasserstein distance between multidimensional distributions (where the dimension equals the number of output classes) requires numerous neural network predictions, leading to slow training. In order to solve this problem, we redefine the regularisation term to apply it to predicted classes of interest, making the bias mitigation problem numerically feasible. Our secondary main contribution from an end-user perspective is to demonstrate how to mitigate algorithmic bias in a text classification problem using modern transformer-based neural network architectures such as RoBERTa small [8]. It is important to note that our regularisation strategy is model-agnostic and could be applied to other text classification models, such as those based on LSTM architectures. We evaluated our method using the *Bios* dataset [9], which includes over 400,000 LinkedIn biographies, each with an occupation and gender label. This dataset is commonly used to train automatic recommendation models for employers to select suitable candidates for a job and quantify algorithmic biases in the trained models. The *Bios* dataset is a key resource for the scientific community studying algorithmic bias in NLP.

2. Definitions and Related Work

Measuring algorithmic biases in machine learning: Different popular metrics exist to measure algorithmic biases in the machine learning literature. In this paper, we used the True Positive Rate gap (TPRg) [9], which is one of the classic fairness metrics for NLP. Other metrics such as the Statistical Parity [10] or Equalised Odds [11] are also very popular. Over 20 different fairness metrics were compared in [12]. Very important for us, each metric shows specific algorithmic bias properties, and not all of them are compatible with each other [13–15]. For instance, the True Positive Rate gap quantifies the difference between the portion of positive predictions ($\hat{Y} = 1$ using common ML notation) in two groups, by only considering the observations that should be classified as positive ($Y = 1$ using common ML notation). Another popular metric such as the disparate impact will also quantify this difference, but for all observations. This makes their practical interpretation different.

Impact of AI biases in society: The use of Artificial Intelligence (AI) in decision-making systems has become increasingly widespread in recent years and, with it, concerns about the potential for discriminating biases to affect the outcomes of these decisions. We review below different key studies that have explored the impacts of such biases in AI on society. One such study [16] focused on the criminal justice system and found that AI algorithms can produce biased outcomes, particularly when trained on non-representative datasets. This can result in higher incarceration rates for certain groups, such as racial minorities, and perpetuate systemic racism in the criminal justice system. Another study by [17] explored how gender differences and biases can affect the development and use of artificial intelligence in the field of bio-medicine and healthcare. The paper discussed the potential consequences of these differences and biases, including unequal access to healthcare and inaccurate medical diagnoses. Another important area in which algorithmic biases can impact society is the case of online advertisements. Ad targeting based on demographic factors such as race, gender, and age rather than interests or behaviours can perpetuate negative stereotypes and result in discrimination by limiting access to job or housing announcements for certain groups. For example, Facebook's ad delivery algorithms, by optimising for maximum engagement, can lead to biased outcomes, which result in the amplification of certain groups or messages over others. This can lead to discrimination against certain groups, as advertisers may target their ads to specific demographics or exclude certain groups from seeing their housing and employment advertising, as highlighted by studies such as [18,19].

Bias mitigation in NLP: Bias in NLP systems has received significant attention in recent years, with researchers and practitioners exploring various methods for mitigating bias in NLP models. In this subsection, we review some of the existing work on bias mitigation in NLP. The first approach to mitigate bias is to apply *pre-processing* techniques to the data used to train the model. Some researchers have proposed methods for removing or neutralising sensitive attributes from the training data, such as gender or race, in order to reduce the likelihood that the model will learn to make decisions based on these attributes. We can reduce the bias directly in the text of the training dataset. For example, in the case of gender bias, like the study in this paper, the most-classic technique is to remove explicit gender indicators [9]. This technique is the one we used to compare our proposed strategy to another one commonly used in industry. This technique is indeed simple to implement and makes it possible to reduce the bias, but in a partial and not very localised manner. Other classical techniques can be used, such as identifying biased data in word embeddings, which represent words in a vector space. Reference [20] demonstrated that these embeddings reflect societal biases. There are also methods to show how these embeddings can be unbiased by aligning them with a set of neutral reference vectors [21,22]. These de-biasing methods have, however, strong limitations, as explained in [23], where the authors showed that, although the de-biasing embedding methods can reduce the visibility of gender bias, they do not completely eliminate it.

A second approach is to use *post-processing* de-biasing methods. These methods are model-agnostic and, therefore, not specific to NLP since they modify the results of previously trained classifiers in order to achieve fairer results. References [11,24] investigated this for binary classification, and Reference [25] proposed a method for multiclass classification.

The last approach to mitigate biases in AI is to use fairness-aware algorithms, which are specifically designed to take into account the potential for bias and to learn from the data in a way that reduces the risk of making biased decisions. These are the *in-processing* methods, which generally do not depend on the type of data input either. The method we propose in this paper is one of them. To achieve this, we can use adversarial learning by adjusting the discriminator. Adversarial learning involves training a model to make predictions while also training a second model to identify and correct any biases in the first model's predictions. By incorporating this technique into the training process, References [26,27] demonstrated that it is possible to reduce the amount of bias present in machine learning models. Another technique is to constrain the predictions with a regularisation technique, such as [28], but this technique was only used on a logistic regression classifier. On the other hand, Reference [29] mitigated fairness specifically in neural networks. Finally, References [30,31] used fairness metrics constraints and solved the training problem subject to those constraints. All these *in-processing* methods apply in the case of binary classification. There is indeed an *in-processing* paper that proposes a method for multiclass classification for a computer vision task [32], but this paper focused on the regularisation of the mean bias amplification and, therefore, did not deal with the classic fairness metrics.

Research implications: We want to emphasise that the *pre-processing* and *post-processing* methods are complementary to *in-processing* methods. Especially when using neural network models, which simultaneously project the input data into an optimal representation (the so-called *latent space* or *feature space*) and use this optimal data representation for their predictions, we believe that *in-processing* methods are those that should be the most-efficient ones. They can indeed both constrain the neural networks to learn fair data representations and fair decision rules based on these data representations. Our paper hence focused on an *in-processing* method. In this context, our methodology tackles an issue that was still not addressed in the fair learning literature, as far as the authors know: we tackled algorithmic biases on multi-class neural network classifiers and not on binary classifiers or on non-neural network classifiers. We believe that the potential of such a strategy is high for the future certification of commercial AI systems. The key methodological contribution of our work is to show how to extend [7] to multi-class classification for regularised mini-batch training of neural networks. As described in Section 3.3, extending this optimal transport regularisation strategy to multi-class classification requires tackling an important technical lock related to the algorithmic cost of the procedure, which is the heart of this methodological contribution. Note that this regularisation strategy applies to any type of multi-class classification neural network model. Thus, the method is particularly flexible and can be applied in various industrial classification problems. From a practical perspective, our secondary contribution is to showcase using the proposed technique for the future certification of neural network application ranked as High Risk by the European Commission. We used in this paper an NLP application and thoroughly describe the procedure to correct strong biases.

3. Methodology

The bias mitigation technique proposed in this paper extends the regularisation strategy of [7] to multi-class classification. In this section, we first introduce our notation, then describe the regularisation strategy of [7] for binary classifiers, and then, extend it to multi-class classifiers. This extension is the methodological contribution of our manuscript.

3.1. General Notations

Input and output observations: Let $(x_i, y_i)_{i=1, \dots, n}$ be the training observations, where $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}^K$ are the input and output observations, respectively. The value p represents the inputs dimension or, equivalently, the number of input variables. It can for instance represent a number of pixels if x_i is an image or a number of words in a text if x_i is a word embedding. The value K represents the output dimensions. In a binary classification context, i.e., if $K = 1$, the fact that $y_i = 0$ or $y_i = 1$ specifies the class of the observation i . In a multi-class classification context, i.e., if $K > 1$, a common strategy consists of using one-hot vectors to encode the class c of observation i : all values y_i^k , $k \in \{1, \dots, K\}$ are equal to 0, except the value y_i^c , which is equal to 1. We use this convention all along this manuscript.

Prediction model: A classifier f_θ with parameters θ is trained so that the predictions $\hat{y}_i \in \{0, 1\}^K$ it indirectly makes based on the outputs $f_\theta(x_i) \in [0, 1]^K$ are *on average* as close as possible to the true output observations y_i in the training set. The link between the model outputs $f_\theta(x_i)$ and the prediction $\hat{y}_i$ depends on the classification context: In binary classification, $f_\theta(x_i)$ is the predicted probability that $\hat{y}_i = 1$, so it is common to use $\hat{y}_i = 1_{f_\theta(x_i) > 0.5}$. Now, by using one-hot-encoded output vectors in multi-class classification, an output $f_\theta(x_i) = (f_\theta^1(x_i), f_\theta^2(x_i), \dots, f_\theta^K(x_i))$ represents the predicted probabilities that the observation i is in the different classes $k \in \{1, 2, \dots, K\}$. As a consequence, $\sum_k f_\theta^k(x_i) = 1$. More interesting for us, the predicted class is the one having the highest probability, so $\hat{y}_i$ is a vector of size K with null values everywhere, except at the index $\arg \max_k f_\theta^k(x_i)$, where its value is 1.

Loss and empirical risk: In order to train the classification model, the empirical risk $\mathcal{R}$ is minimised with respect to the model parameters θ :

$$\mathcal{R}(\theta) := \mathbb{E}[\ell(\hat{Y} := f_\theta(X), Y)], \quad (1)$$

or, empirically, $R(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(\hat{y}_i := f_\theta(x_i), y_i)$, where the loss function ℓ represents the price paid for the inaccuracy of the predictions. This optimisation problem is almost systematically solved by using variants of stochastic (or mini-batch) gradient descent [33] in the machine learning literature.

Sensitive variable: An important variable in the field of *fair learning* is the so-called *sensitive variable*, which we denote S . This variable is often binary and distinguishes two groups of observations $S_i \in \{0, 1\}$. For instance, $S_i = 0$ or $S_i = 1$ can indicate that the person represented in observation i is either a male or a female. A widely used strategy to quantify that a prediction model is fair with respect to the variable S is to compare the predictions it makes on observations in the groups $S = 0$ and $S = 1$, using a pertinent *fairness metric* (see the references of Section 2). From a mathematical point of view, this means that the difference between the distributions $(X, Y, \hat{Y})_{S=0}$ and $(X, Y, \hat{Y})_{S=1}$, quantified by the fairness metric, should be below a given threshold. Consider for instance a binary prediction case where $\hat{Y}_i = 1$ means that the individual i has access to a bank loan, $\hat{Y}_i = 0$ means that the bank loan is refused, and that S_i equal to 0 or 1 refers to the fact that the individual i is a male or a female. In this case, one can use the difference between the empirical probabilities of obtaining the bank loan for males and females, as a fairness metric, i.e., $P(\hat{Y} = 1 | S = 1) - P(\hat{Y} = 1 | S = 0)$. More advanced metrics may also take into account the input observation X , the true outputs Y , or the prediction model outputs $f_\theta(X)$ instead of their binarised version $\hat{Y}$.

3.2. W2reg Approach for Binary Classification

3.2.1. Regularisation Strategy

We now give an overview of the *W2reg* approach, described in [7], to temper algorithmic biases of binary neural network classifiers. The goal of *W2reg* is to ensure that the treated binary classifier f_θ generates predictions $\hat{Y}$ for which the distributions in groups $S = 0$ and $S = 1$ do not deviate too much from pure equality. To achieve this, the similarity metric used in [7] is the 2-Wasserstein distance between the distribution of the predictions in the two groups:

$$\mathcal{W}_2^2(\mu_{\theta,0}, \mu_{\theta,1}) = \int_0^1 \left(\mathcal{H}_{\theta,0}^{-1}(\tau) - \mathcal{H}_{\theta,1}^{-1}(\tau) \right)^2 d\tau. \quad (2)$$

where $\mu_{\theta,s}$ is the probability distribution of the predictions made by f_θ in group $S = s$ and $\mathcal{H}_{\theta,s}^{-1}$ is the inverse of the corresponding cumulative distribution function. Note that $\mu_{\theta,s}$ is mathematically equivalent to the histogram of the model outputs $f_\theta(X)$ for an infinity of observations in the group $S = s$, after normalisation, so that the histogram integral is 1. We remark that this metric is also based on the model outputs $f_\theta(X) \in [0, 1]$ and not the discrete predictions $\hat{Y} \in \{0, 1\}$ (see Section 3.1—the *prediction model* for the formal relation), so the probability distributions $\mu_{\theta,s}$ are continuous. Ensuring that this metric remains low makes it possible to control the level of fairness of the neural network model f_θ with respect to S . As specifically modelled by Equation (2), this is performed by penalising the average squared difference between the quantiles of the predictions in the two groups. In order to train a neural network that simultaneously makes accurate and fair decisions, the strategy of [7] then consists of optimising the parameters θ of the model f_θ such that

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \left\{ \mathcal{R}(\theta) + \lambda \mathcal{W}_2^2(\mu_{\theta,0}, \mu_{\theta,1}) \right\}, \quad (3)$$

where Θ is the space of the neural network parameters (e.g., the values of the weights, the bias terms, and the convolution filters in a CNN). As usual, when training a neural network, the parameters θ are optimised using a gradient descent approach, where the gradient is approximated at each gradient descent step by using a mini-batch of observations.

3.2.2. Gradient Estimation

We computed the gradient of Equation (3) using the standard backpropagation strategy [34]. For the empirical risk part of Equation (3), this requires computing the derivatives of the losses $\ell(f_\theta(x_i), y_i)$ with respect to the neural network outputs $f_\theta(x_i)$, something routinely performed by packages such as PyTorch, TensorFlow, or Keras, for all mainstream losses. For the 2-Wasserstein part of Equation (3), the authors of [7] proposed to use a mathematical strategy to compute the pseudo-derivatives of $\mathcal{W}_2^2(\mu_{\theta,0}, \mu_{\theta,1})$ with respect to the neural network outputs $f_\theta(x_i)$. Specifically, to compute the pseudo-derivative of a discrete and empirical approximation of $\mathcal{W}_2^2(\mu_{\theta,0}, \mu_{\theta,1})$ with respect to a mini-batch output $f_\theta(x_i)$, the following equation was used:

$$\Delta_\tau \left[1_{s_i=0} \frac{f_\theta(x_i) - \text{cor}_1(f_\theta(x_i))}{n_0(H_0^{j_i+1} - H_0^{j_i})} - 1_{s_i=1} \frac{\text{cor}_0(f_\theta(x_i)) - f_\theta(x_i)}{n_1(H_1^{j_i+1} - H_1^{j_i})} \right], \quad (4)$$

where n_s is the number of observations in class $S = s$ and H_s^j are discrete versions of the cumulative distribution functions $\mathcal{H}_{\theta,s}$ defined on a discrete grid of possible output values:

$$\eta^j = \min_i (f_\theta(x_i)) + j\Delta_\eta, \quad j = 1, \dots, J_\eta, \quad (5)$$

where $\Delta_\eta = J_\eta^{-1}(\max_i(f_\theta(x_i)) - \min_i(f_\theta(x_i)))$ and J_η is the number of discretisation steps. We denote $H_s^j = H_s(\eta^j)$, and j_i is defined such that $\eta^{j_i} \leq f_\theta(x_i) < \eta^{j_i+1}$. Finally, $cor_s(f_\theta(x_i)) = H_s^{-1}(H_{[1-s]}(f_\theta(x_i)))$.

3.2.3. Distinction between Mini-Batch Observations and the Observations for H_0 and H_1

As shown in Equation (4), computing the pseudo-derivatives of the 2-Wasserstein distance $W_2^2(\mu_{\theta,0}^n, \mu_{\theta,1}^n)$ with respect to the model predictions $f_\theta(x_i)$ requires computing the discrete cumulative distribution functions H_s , with $s \in \{0, 1\}$. Computing H_s would ideally require computing $f_\theta(x_i)$ for all n observations x_i of the training set, which would be a computational bottleneck. To solve this issue, Reference [7] proposed approximating H_s at each mini-batch iteration, where Equation (4) is computed, using a subset of all training observations. This observation subset is composed of m randomly drawn observations in group $S = 0$, m other randomly drawn observations in group $S = 1$, and the mini-batch observations. This guarantees that there are at least m observations to compute either H_0 or H_1 and that the impact of each mini-batch observation is represented in H_0 and H_1 . Note that these additional $2m$ predictions do not require backpropagating any gradient information, so their computational burden is limited in terms of memory resources. Although it is also reasonable in terms of computational resources, the amount of $2m$ additional predictions should remain relatively small to avoid significantly slowing down the gradient descent. In previous experiences on images, $m = 16$ or $m = 32$ often appeared as reasonable, as this allowed mitigating undesirable algorithmic biases and slowed down the whole training procedure by a factor of less than 2. Finally, preserving the amount of such additional predictions to something reasonable at each gradient descent step is at the heart of our methodological contribution when extending *W2reg* to multi-class classification.

3.3. Extended *W2reg* for Multi-Class Classification

As discussed in Section 1, our work was motivated by the need for bias mitigation strategies in NLP applications where the neural network predicts that an input text belongs to a class among more K output classes, where $K > 2$. We show in this section how to take advantage of the properties of [7] to address this practical problem. We recall that the regularisation strategy of [7] is model-agnostic, so the fact that we treated NLP data will only be discussed in the Results Section. In terms of methodology, the main issue to tackle is that the model outputs $f_\theta(x_i)$ are in dimension $K > 2$ and not one-dimensional, which would require comparing multivariate point clouds following the optimal transport principles, which were modelled by Equation (2) for 1D outputs. As we will see below, this generates algorithmic problems to keep the computational burden reasonable and to preserve the representativity of the pertinent information. Solving them requires extending [7] with strong algorithmic constraints.

3.3.1. Reformulating the Bias Mitigation Procedure for Multi-Class Classification

The strategy proposed by [7] to mitigate undesired biases is to train optimal decision rules f_θ by optimising Equation (3), where the 2-Wasserstein distance between the prediction distributions $\mu_{\theta,0}$ and $\mu_{\theta,1}$ (i.e., the distribution of the predictions f_θ for observations in groups $S = 0$ and $S = 1$) is given by Equation (2). As described in Section 3.1, the predictions $f_\theta(x_i)$ are now a vector of dimension $K > 2$ in a multi-class classification context (specifically, $f_\theta(x) \in [0, 1]^K$). Their distributions $\mu_{\theta,0}$ and $\mu_{\theta,1}$ are then multivariate. In this context, Equation (2) does not hold, and another optimal transport metric such as the multivariate 2-Wasserstein distance or the Sinkhorn Divergence should be used [35]. Note that different implementations of these metrics exist and are compatible with our problem, e.g. those of [36,37]. This, however, opens a critical issue related to the number of observations needed to reasonably penalise the differences between two multivariate point clouds, representing the observations in groups $S = 0$ and $S = 1$. If the dimension K of the compared data becomes large, the number of observations required to reasonably compare

the point clouds at each gradient descent step explodes. This problem is very similar to the well-known *curse of dimensionality* phenomenon in machine learning, where the amount of data needed to accurately generalise the predictions grows exponentially as the number of dimensions grows.

This issue, therefore, lead us to think about which problem we truly need to solve when tackling undesired algorithmic bias in multi-class classification. From our application perspective, discrimination appears when there the prediction model f_θ is significantly more accurate at predicting a specific output in one of the two groups represented by $S \in \{0, 1\}$. For instance, suppose that someone looks for *Software Engineer* jobs and that an automatic prediction model f_θ is used to recommend job candidates to an employer. For a given job candidate x_i , the prediction model will return a set of K probabilities, each of them indicating whether x_i is recommended for the job class k . Now, k will denote the class of jobs x_i is looking for, i.e., *Software Engineer*. The prediction model will be considered as unfair if male profiles are on average clearly more often recommended by f_θ than female profiles, when an unbiased oracle would lead to equal opportunities, i.e.,

$$|P(\hat{Y}^k = 1 | Y^k = 1, S = 1) - P(\hat{Y}^k = 1 | Y^k = 1, S = 0)| > \tau, \quad (6)$$

where the left-hand term denotes the *True Positive Rate gap* (TPRg) and τ is a threshold above which the TPRg is considered as unfairly discriminating. As shown in Section 5, such situations can occur in automatic job profile recommendation systems using modern neural networks. Now that we have clarified the problem we need to tackle, we can reformulate the regularised multi-class model training procedure as follows:

- We first trained and tested a non-regularised multi-class classifier $f_{\theta^{bl}}$. We denote it the *baseline classifier*.
- We defined a threshold τ under which all occupations with predictions $k \in \{1, \dots, K\}$ should have a TPRg (see Equation (6)). We denote $\{c_1, \dots, c_C\}$ the classes for which this condition is broken, where each of these classes takes its values in $\{1, \dots, K\}$.
- We then retrained the multi-class classifier f_θ with regularisation constraints on the classes $\{c_1, \dots, c_C\}$ only. The regularisation strategy will be developed below in Section 3.3.3.

By using this procedure, the number of observations required at each mini-batch step will be first limited to observations in the groups $\{c_1, \dots, c_C\}$ only, which is a first step towards an algorithmically reasonable regularised training procedure. We also believe that this also avoids over-constraining the training procedure, which often penalises its convergence.

3.3.2. Regularisation Strategy

We now push further the algorithmic simplification of the regularisation procedure by focusing on the properties of the mini-batch observations. In this subsection, we suppose that x_i is an input mini-batch observation, and recall that we want to penalise large TPRg for specific classes $\{c_1, \dots, c_C\}$ only. In this mini-batch step, the observations x_i related to true output predictions $y_i^k = 1$ for which $k \notin \{c_1, \dots, c_C\}$ are not concerned by the regularisation, when computing the multivariate cumulative distribution function H_0 or H_1 . At each mini-batch step, it, therefore, appears as appealing to only consider the dimensions out of $\{c_1, \dots, c_C\}$, for which at least one true output observation respects $y_i^k = 1$, with $k \in \{c_1, \dots, c_C\}$. This would indeed allow further reducing the amount of additional predictions made in the mini-batch. The dimension of H_0 or H_1 would, however, vary at each mini-batch step, potentially making the distance estimation unstable if fully considering C -dimensional distributions.

To take into account the fact that not all output dimensions $\{c_1, \dots, c_C\}$ should be considered at each gradient descent step, we then made a simplification hypothesis: we neglected the relations between the different dimensions when comparing the output predictions in groups $S = 0$ and $S = 1$. This hypothesis is the same as the one made when using Naive Bayes classifiers [38,39]. We believe that this hypothesis is particularly suited for one-hot-encoded outputs, as they are constructed to ideally have a single value close to 1 and all other values close to 0. We then split the multivariate regularisation strategy into a multiple one-dimensional strategy and optimised

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \left\{ \mathcal{R}(\theta) + \sum_{l=1}^C \lambda_{c_l} \mathcal{W}_2^2(\mu_{\theta,0}^{c_l}, \mu_{\theta,1}^{c_l}) \right\}, \quad (7)$$

where $\mathcal{W}_2^2$ is the metric of Equation (2), λ_{c_l} is the weight given to regularise the TPR gaps in class c_l , and $\mu_{\theta,s}^{c_l}$ are the distributions of the output predictions on dimension c_l , i.e., the distribution of $f_{\theta}^{c_l}(x)$, when the true prediction is c_l , i.e., when $y^{c_l} = 1$. For a mini-batch observation x_i related to an output prediction in a regularised class $k \in \{c_1, \dots, c_C\}$, the impact of a mini-batch output $f_{\theta}^k(x_i)$ on the empirical approximation of $\mathcal{W}_2^2(\mu_{\theta,0}^k, \mu_{\theta,1}^k)$ can then be estimated by following the same principles as in [7]. We can then extend Equation (4) with

$$\Delta_{\tau} \left[1_{s_i=0} \frac{f_{\theta}^k(x_i) - \text{cor}_1(f_{\theta}^k(x_i))}{n_{k,0}(H_{k,0}^{j_i+1} - H_{k,0}^{j_i})} - 1_{s_i=1} \frac{\text{cor}_0(f_{\theta}^k(x_i)) - f_{\theta}^k(x_i)}{n_{k,1}(H_{k,1}^{j_i+1} - H_{k,1}^{j_i})} \right], \quad (8)$$

where $H_{k,s}$ are discrete and empirical versions of the cumulative distribution functions of the prediction outputs on dimension k , i.e., the $f_{\theta}^k(x)$, when class k should be predicted and the observations are in the group s . Note also that Equation (4) contains n_s , which is the number of observations in class $S = s$. In order to manage unbalanced output classes in the multi-class classification context, we also use a normalising term $n_{k,s}$ in Equation (8). It quantifies the number of training observations in group $s \in \{0, 1\}$ and class $k \in \{1, \dots, K\}$. Other notations are the same as in Equation (4).

In a mini-batch step, suppose, finally, that we only need to take into account the classes $\{\hat{c}_1, \dots, \hat{c}_D\}$ among $\{c_1, \dots, c_C\}$. These selected classes are those for which at least a $y_i^j = 1$, with $j \in \{1, \dots, C\}$, and i is an observation of the mini-batch $B \subset \{1, \dots, n\}$. We then have to only sample two-times m predictions, for each of the selected D classes, to compute the $H_{\hat{c}_d,0}$ and $H_{\hat{c}_d,1}$ required in Equation (8). This makes the computational burden to regularise the neural network training procedure reasonable, as the number of additional predictions to make only increases linearly with the number of treated classes at each mini-batch iteration. Note that no additional prediction will also be needed when a mini-batch contains no observation related to a regularised output class. This will naturally be often the case, when the number of classes K becomes large and/or the mini-batch size $\#B$ is small.

3.3.3. Proposed Training Procedure

The proposed strategy to train multi-class classifiers with mitigated algorithmic biases on specific classes' prediction was motivated by the future need of certifying that automatic decision models are not discriminatory. In order to make absolutely clear our strategy, we detail it in Algorithm 1.

Algorithm 1 Procedure to train bias mitigation multi-class neural network classifiers.

Require: Training observations $(x_i, s_i, y_i)_{i=1, \dots, n}$, where $x_i \in \mathbb{R}^p$, $s_i \in \{0, 1\}$ and $y_i \in \{0, 1\}^K$, plus a multi-class neural network model f_θ .

- 1: [*Detection of the output classes with discriminatory predictions*]
- 2: Train the baseline parameters θ^{bl} of f on $(x_i, s_i, y_i)_{i=1, \dots, n}$ with no specific regularisation.
- 3: Find the output classes $\{c_1, \dots, c_C\}$ on which the model $f_{\theta^{bl}}$ has unacceptable True Positive Rate gaps (TPRg) using Equation (6).
- 4: [*Multi-class W2reg training*]
- 5: Re-initialise the training parameters θ .
- 6: **for** e in epochs **do**
- 7: **for** b in batches **do**
- 8: Draw the batch observations $(x_i, s_i, y_i)_{i \in B}$, where B is a subset of $\{1, \dots, n\}$.
- 9: Compute the mini-batch predictions $f_\theta(x_i)$, $i \in B$.
- 10: Detect the output classes $\{\hat{c}_1, \dots, \hat{c}_D\}$ among $\{c_1, \dots, c_C\}$ for which at least a $y_i^{c_j} = 1$, with $j \in \{1, \dots, C\}$ and $i \in B$.
- 11: For each $j \in \{1, \dots, D\}$, pre-compute $H_{\hat{c}_j, 0}$ and $H_{\hat{c}_j, 1}$ using m output predictions $f_\theta(x_i)$, where $i \notin B$ and $y_i^{\hat{c}_j} = 1$.
- 12: Compute the empirical risk and its derivatives with respect to $f_\theta(x_i)$, $i \in B$.
- 13: Compute the pseudo-derivatives of the discretised $W_2^2(\mu_{\theta, 0}^{\hat{c}_j}, \mu_{\theta, 1}^{\hat{c}_j})$ with respect to the pertinent mini-batch outputs $f_\theta(x_i)^{\hat{c}_j}$ using Equation (8).
- 14: Backpropagate the risk derivatives and the pseudo-derivatives of the W_2^2 terms.
- 15: Update the parameters θ .
- 16: **end for**
- 17: **end for**
- 18: **return** Trained neural network f_θ with mitigated biases.

4. Experimental Protocol to Assess W2reg on Multi-Class Classification with NLP Data**4.1. Data**

We assessed our methodology using the *Bios* [9] dataset, which contains about 400 K biographies (textual data). For each biography, *Bios* specifies the gender (M or F) of its author, as well as his/her occupation (among 28 occupations, categorical data). As shown in Figure 1, this dataset contains heterogeneously represented occupation. Although the representation of some occupations is relatively well balanced between males and females (e.g., professor, journalist, etc.), other occupations are particularly unbalanced between males and females (e.g., nurse, software engineer, etc.). Note that, to build this dataset, its authors used Common Crawl and identified online biographies written in English. Then, they filtered the biographies starting with a name-like pattern followed by the string “is a(n) (xxx) title”, where title is an occupation out of the BLS Standard Occupation Classification system. Having identified the twenty-eight most-frequent occupations, they processed WET files from sixteen distinct crawls from 2014 to 2018, extracting online biographies corresponding to those occupations only. This resulted in about 400 K biographies with labelled corresponding occupations.

This dataset is particularly interesting for our study, as it first makes it possible to evaluate how our regularisation strategy can tackle undesirable gender biases when trying to predict the true occupations of potential job candidates as accurately as possible. It is, in addition, well known in the fair learning community, as it is the largest NLP dataset for multi-class classification with known genders for each observation. As shown in Figure 1, the amount of biographies is particularly unbalanced across the occupations to predict, and the portion of females and males is highly variable for each of these occupations. Statistically studying the gender biases obtained in this dataset, therefore, allows us to compare various cases where multi-class neural network classifiers can have an undesirable behaviour.

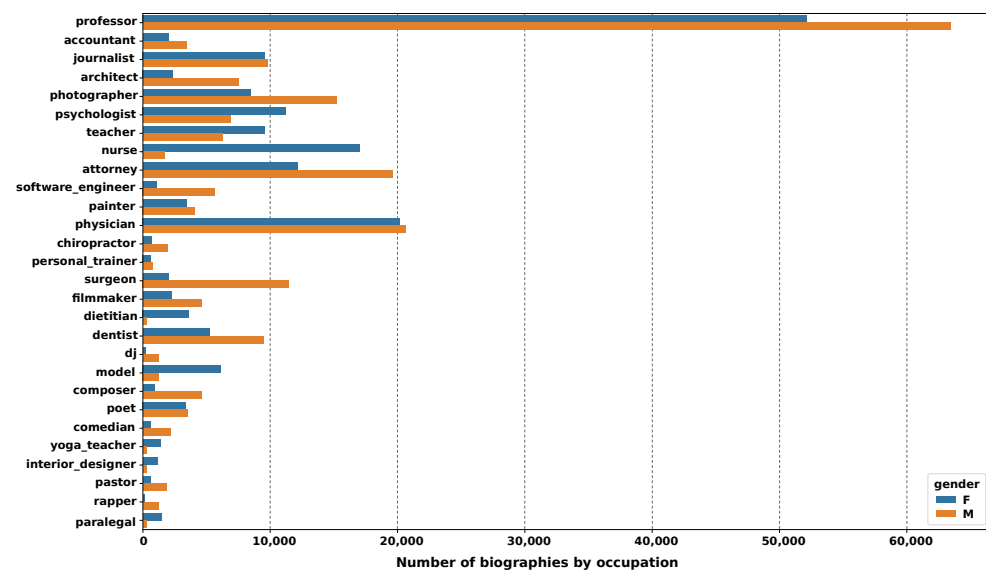


Figure 1. Number of biographies for each occupation by gender on the total *Bios* dataset [9].

4.2. Neural Network Model and Baseline Training Strategy

Our task was to predict the occupation using only the textual data of the biography. We did this by using a RoBERTa model [8], which is based on the transformer architecture and is pretrained with the Masked Language Modelling (MLM) objective. We specifically used a RoBERTa base model pretrained by Hugging Face. All information related to how it was trained can be found in [8]. It can be remarked that a very large training dataset was used to pretrain the model, as it was composed of five datasets: *BookCorpus* [40], a dataset containing 11,038 unpublished books; *English Wikipedia* (excluding lists, tables, and headers); *CC-News* [41], which contains 63 millions English news articles crawled between September 2016 and February 2019; *OpenWebText* [42], an open-source recreation of the WebText dataset used to train GPT-2; *Stories* [43], a dataset containing a subset of CommonCrawl data filtered to match the story-like style of Winograd schemas. Pre-training was performed on these data, by randomly masking 15% of the words in each of the input sentences and then trying to predict the masked words. After pre-training RoBERTa parameters on this huge dataset, we then trained it on the 400,000 biographies of the *Bios* dataset. The training was performed with PyTorch on 2 GPUs (Nvidia Quadro RTX6000 24 GB RAM) for 10 epochs with a batch size of 8 observations and a sequence length of 512 words. The optimiser was Adam with a learning rate of 10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^6$. The computational time was about 36 h for each run. We want to emphasise that 5 runs of the training procedure were performed to evaluate the stability of the accuracy and the algorithmic biases. For each of these runs, we split the dataset into 70% for training, 10% for validation, and 20% for testing. We denote, as the baseline models, the neural networks trained using this procedure.

4.3. Evaluating the Impact of a Gender-Neutral Dataset

In order to evaluate the impact of a classic gender unbiasing strategy, we reproduced the baseline training protocol of Section 4.2 on two *apparently* unbiased versions of the *Bios* dataset. This classic method for debiasing consists of removing explicit gender indicators (i.e., “he”, “she”, “her”, “his”, “him”, “hers”, “himself”, “herself”, “mr”, “mrs”, “ms”, “miss”, and first names). For a BERT model type, however, we could not just remove words because the model is sensitive to sentence structure, not just lexical information. We, therefore, adjusted the method by replacing all the first names by a neutral first name (*Camille*) and by choosing only one gender for all datasets (e.g., for all individuals of gender *g*, we did nothing; for the others, we replaced explicit gender indicators with those of *g*). We

then created two datasets with only female or male gender indicators and the only first name *Camille*.

Note that, by using a fully trained model on our dataset, setting all gender indicators to either feminine or masculine should naively not change anything, since the model would only “know” one gender (which would, therefore, be neutral). We, however, used a pre-trained model on gendered datasets. It is, therefore, important to verify that fine-tuning this model with a male-gendered dataset is equivalent to training it on a female-gendered dataset. To assess this, we carried out several student tests: one between the accuracy of the trained model on the female-gendered dataset and the accuracy of the male-gendered one and one on the TPR gender gap for each of the professions between the two models. None of these tests had a statistically significant difference. We will then only present in Section 5 the results obtained on the model trained on the female-gendered dataset.

4.4. Training Procedure for the Regularised Model

We now follow the procedure summarised in Algorithm 1 to train bias mitigation multi-class neural network classifiers on the textual data of the *Bios* dataset. We first considered the 5 baseline models of Section 4.2, which were trained on the original *Bios* dataset (and not one of the unbiased datasets of Section 4.3). As shown in Figure 2 (left), where the diagonal of the confusion matrices’ differences between males and females represents the TPR gap of all output classes, two classes have TPR gaps above 0.1 or under -0.1 : *Surgeon* (in favour of males) and *Model* (in favour of females). We then chose to regularise the predictions for these two occupations. Other occupations could have been considered (see Figure A1), but they did not contain enough statistical information to be properly treated. For instance, although the whole training set contains about 400.000 observations, it contains less than 100 female *DJs* and less than 100 male *Paralegals*.

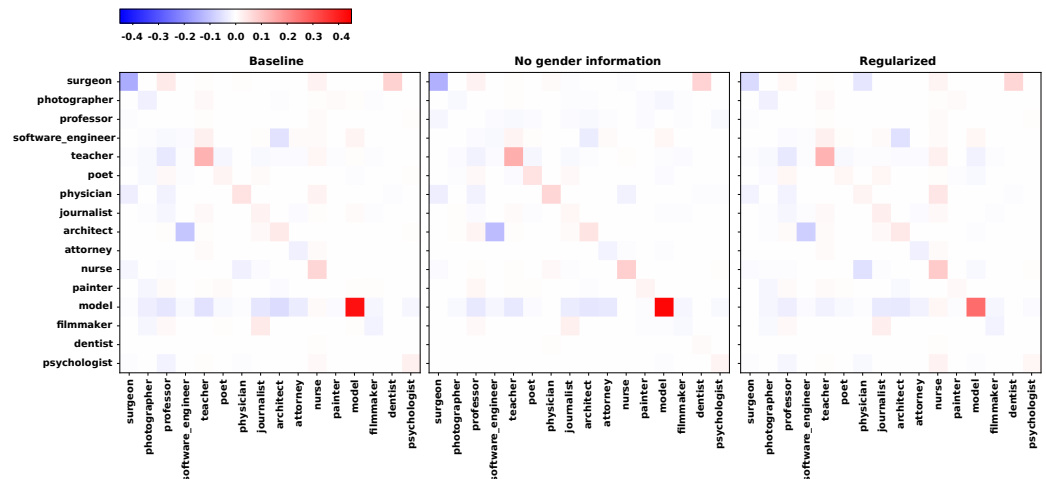


Figure 2. Average difference between the confusion matrices of the predicted outputs $\hat{y}$ versus the true outputs y obtained for females and males. Note that the diagonal values of these matrices correspond to the average TPR gaps. The confusion matrices were also normalised over the true (rows) conditions. The redder a value, the stronger the bias in favour of females is, and the bluer a value, the stronger the bias in favour of males is.

After having selected these two occupations, we trained 5 regularised models by minimising Equation (7). We chose a single λ parameter for the regularisation (the same for both classes, but we could have taken one per class), by using cross-validation, with the goal of effectively reducing the TPR gaps on regularised classes without harming the accuracy too much. The best performance/debiasing compromise we found was $\lambda = 0.0001$. An amount of $m = 16$ additional observations was used at each mini-batch step to compute each of the discrete cumulative histograms $H_{k,s}$ of the regularisation terms’ pseudo-derivatives Equa-

tion (8). The rest of the training procedure was the same as in Section 4.2. Computational times required about 70 h for each run.

4.5. Overview of the Classifiers Compared in Section 5

In order to assess the impact of the in-processing bias mitigation technique proposed in this paper, we will compare it in Section 5 to a non-regularised strategy (denoted as the *baseline*) and a pre-processing strategy where the biographies were made neutral (denoted as *no gender information*). We recall that 5 models were trained in each case to evaluate the stability of the training procedures and their biases. An overview of the pipeline used in each case is shown in Figure 3.

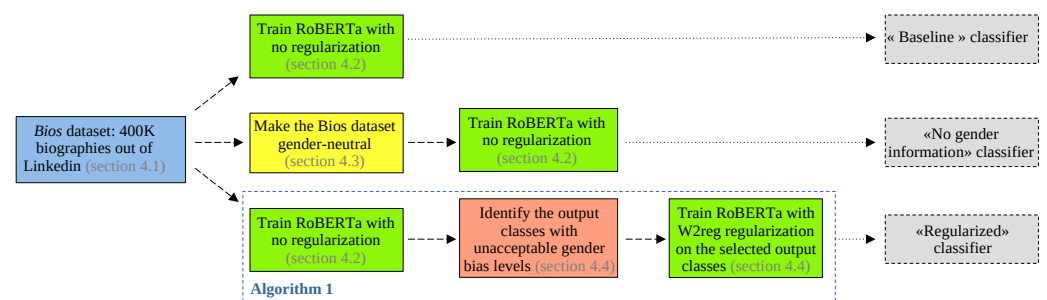


Figure 3. Pipelines followed to define the three classifiers compared in Section 5.

5. Results and Discussion

In commercial applications, fair prediction algorithms will be obviously more popular and useful if they remain accurate. Thus, we made sure that our regularisation technique did not have a strongly negative impact on the prediction accuracy. We then quantified different accuracy metrics: first, the average accuracy and, then, two variants of the F1-score, as it is very appropriate for a multiclass classification problem like ours. These two variants are the so-called “macro” F1-score, where we calculate the metric for each class, then we average it without taking into account the number of individuals per class; and the “weighted” F1-score, where the means are weighted using the classes’ representativeness. We can draw similar conclusions for these three metrics, as shown in Figure 4: our regularisation method was certainly a little below the baseline in terms of accuracy, but it was more stable. In addition, it was clearly more accurate than the gender-neutralising technique of Section 4.3.

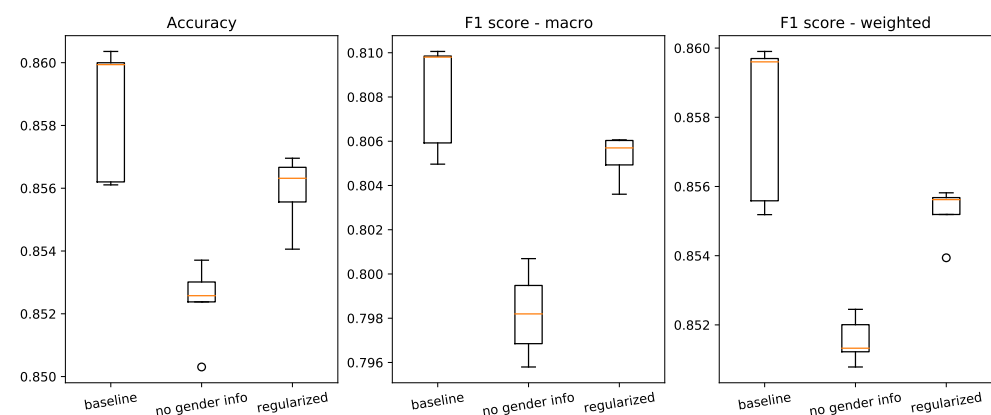


Figure 4. (Left to right) Box plots of the accuracy, unweighted F1-scores, and weighted F1-scores for the baseline models with raw biographies, the baseline models with unbiased biographies, and the regularised models with raw biographies.

We then specifically observed the impact of our regularisation strategy in terms of the TPR gap on the two regularised classes: *Surgeon* and *Model*. Boxplots of the TPR gap for these output classes are shown in Figure 5. They confirmed that the algorithmic bias was reduced for these two classes. For the class *Surgeon*, removing gender indicators had a strong effect, but the regularisation strategy further reduced the biases. For the class *Model*, removing gender indicators had little effect, and the regularisation strategy reduced the biases by almost a factor of two.

We finally wanted to make sure that reducing the unacceptable biases on these two classes would not be at the expense of newly generated biases. We then measured the difference between the average (on the five models) confusion matrix for females and males only. In Figure 2, we see the evolution of our biases according to the selected method. Note first that the diagonal of these matrix differences corresponds to the TPR gaps. We also remark that we only represent the results obtained on the 16 most-frequent occupations for visibility concerns, but the complete matrices are shown in the Appendix A. On our two regularised classes, we came closer to white (i.e., non-bias), and for the other classes, we also observed a decrease in bias in general and no outlier point. For a finer analysis and more clarity, we represent in Figure 6 the difference between the absolute values of the baseline matrix of Figure 2 and each of the compared matrices (i.e., with neutralised genders and regularisation). This clearly represents to us the “gains” of these two bias-reduction methods to compare them. Figure 6 confirms our intuition given in Figure 2: in the case where the gender indicators were removed, the gain was rather slight and depended on the class. In the case of our regularisation, the two regularised classes obtained a very clear positive gain, and there was no marked negative gain on the rest of the matrix.

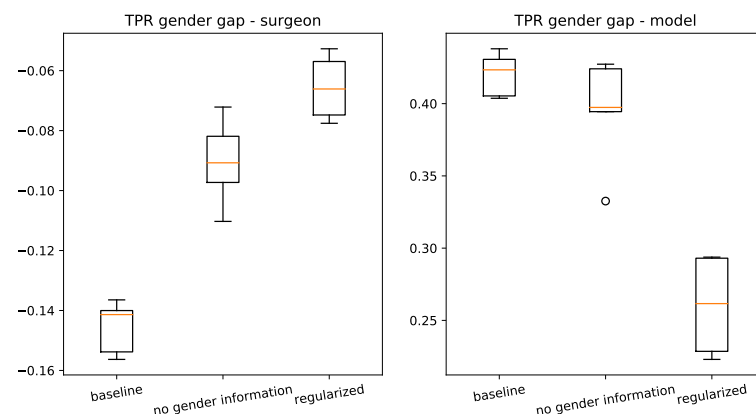


Figure 5. Box plots of the True Positive Rate (TPR) gender gaps for the output classes *Surgeon* and *Model* obtained using the baseline models with raw biographies, the baseline models with unbiased biographies, and the regularised models with raw biographies. Note that there is a bias in favour of females or males if a TPR gender gap is positive or negative, respectively.

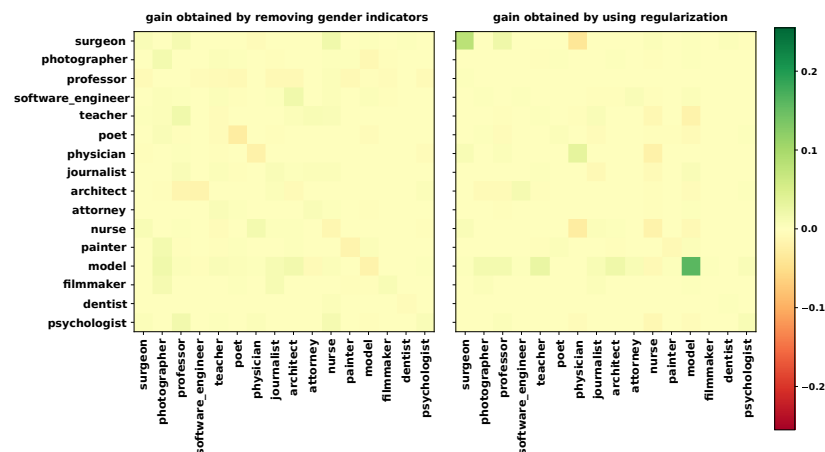


Figure 6. (Left) Difference between the baseline matrix of Figure 2 and the one obtained with an unbiased training set. (Right) Difference between the baseline matrix of Figure 2 and the one obtained using regularised optimisation. The greener the values, the more the technique has reduced the bias between men and women. The redder the values, the more the bias has been amplified.

6. Conclusions

In this paper, we defined a strategy to address the critical need for certifying that commercial prediction models present moderate discrimination biases. We specifically defined a new algorithm to mitigate undesirable algorithmic biases in multi-class neural network classifiers and applied it to an NLP application that is ranked as *High Risk* by EU regulations. Our method was shown to successfully temper algorithmic biases in this application and outperformed a classic strategy both in terms of prediction accuracy and mitigated bias. In addition, the computational times were only reasonably increased compared with a baseline training method. The state-of-the-art of *in-processing* unbiasing methods mainly focuses on binary models, and our approach addresses the multiclass problem. The possibility of choosing which classes to regularise and of applying a different λ for each class gives a wide range of application of the method.

We want to make clear two potential difficulties that we anticipate for future users of our method: (1) The W2reg method allows the user to choose a specific λ value for each regularised class. As mentioned in the previous paragraph, this makes the method very flexible and gives control on the level of regularisation required to obtain reasonable biases for each output class. Finding optimal regularisation weights can, however, be time consuming for the user. Note that this compromise between accuracy and regularisation is, however, extremely common in engineering science. (2) As thoroughly discussed in Appendix A, the regularisation strategy is also effective when the training set has a reasonable amount of observations in a treated class. Using our method in order to later certify that a multi-class classification neural network is not biased for a specific output class will then require having enough training data in this class. This phenomenon is related to the generalisation properties of any decision model trained on reference data. More observations will have to be acquired in these classes otherwise.

Now that [7] has been extended to multi-class classification, a natural perspective would be to also use it for the regression case. We expect this extension to be methodologically straightforward, as the binary and multi-class W2reg strategies for classification already regularise continuous outputs (specifically, the probabilities of belonging to a class). A more challenging perspective would be to adapt and eventually reformulate W2reg to other popular decision models, as all applications of AI are not based on neural networks. Adapting W2reg to other fairness criteria would finally make it more versatile.

We finally want to emphasise that, although our method was applied to NLP data, it can be easily applied to any multi-class neural network classifier. We also believe that it could be simply adapted to other fairness metrics. Our regularisation method was

implemented to work as a loss in PyTorch and is compatible with PyTorch-GPU. It is freely available on GitHub (<https://github.com/lrisser/W2reg> accessed on 14 March 2023).

Author Contributions: Conceptualisation, F.J. and L.R.; methodology, F.J., L.R. and J.-M.L.; software, F.J. and L.R.; validation, F.J.; formal analysis, F.J. and L.R.; investigation, F.J., L.R. and J.-M.L.; resources, F.J., T.T.K. and L.R.; data curation, F.J., T.T.K. and L.R.; writing—original draft preparation, F.J. and L.R.; writing—review and editing, F.J., L.R., N.A. and J.-M.L.; visualisation, F.J.; supervision, L.R.; project administration, N.A.; funding acquisition, N.A., J.-M.L. and L.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Artificial Intelligence (AI) Interdisciplinary Institute Artificial and Natural Intelligence Institute (ANITI), which is funded by the French “Investing for the Future—PIA3” program under Grant Agreement ANR-19-PI3A-0004. Titon Tshiongo Kaninku participated in the data curation and was funded by plan France Relance under Grant Agreement ANR-21-PRRD-0018.

Data Availability Statement: No new dataset was created in this study. The code to reproduce our experiments is freely available on GitHub (<https://github.com/lrisser/W2reg>).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

Appendix A

The results shown in Figures 2 and 6 selected the most-largely represented output classes for readability purposes. We show in this Appendix their extensions, Figures A1 and A2, to all output classes of the *Bios* dataset [9]. It can be observed in these figures that other output classes than *Model* and *Surgeon* presented high gender biases, when using the baseline strategy: *Paralegal*, *DJ*, and *Dietician*. Although we used these output classes when training the prediction model to make the classification task complex, we voluntarily decided not to regularise them for statistical concerns: These occupations are indeed first poorly represented in the *Bios* dataset and are additionally strongly unbalanced between males and females. Although the whole training set contains more than 400,000 biographies, there are less than 100 biographies for female *DJs*, male *Dieticians*, and male *Paralegals*. This makes their treatment with a statistically sound strategy unreliable. When applied to statistically poorly represented observations, a constrained neural network will not indeed learn to use generalizable features in the input biographies, but will instead overfit the specificities of each observation, which is strongly highlighted by the constraint. We can, however, see that the tested bias mitigation strategies on the classes *Model* and *Surgeon* did not amplify the biases on the *Paralegal*, *DJ*, and *Dietician* classes.

From a certification perspective in the EU, the *AI act* will ask to clearly mention to end-users the cases for which the predictions may be unreliable or potentially biased. In this context, our strategy makes it possible to certify that multi-class neural network classifiers make unbiased decisions on output classes that would be biased using standard training, if the training data offer a sufficient representativity and variability of the characteristics in these classes. In the case where a company would desire to certify that poorly represented classes in the training set are free of biases, the certification procedure will naturally require acquiring more observations.

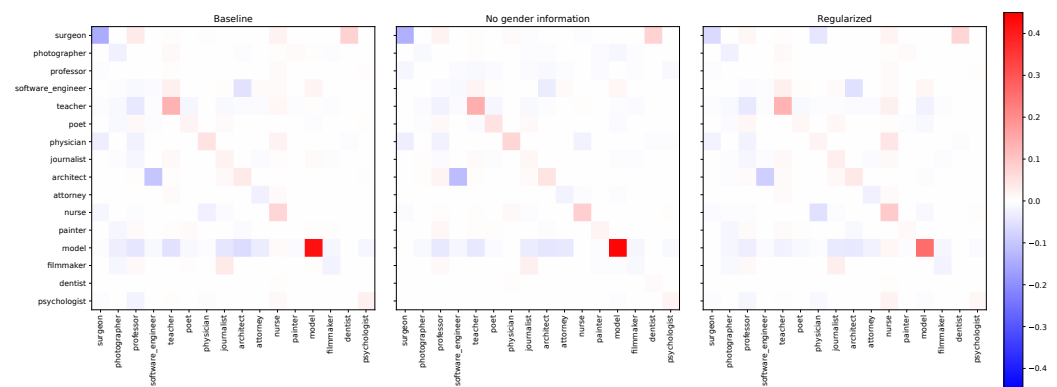


Figure A1. Extension of Figure 2 to all output classes of the Bios dataset and not only the most-frequent ones, which were selected in Figure 2 for readability purposes.

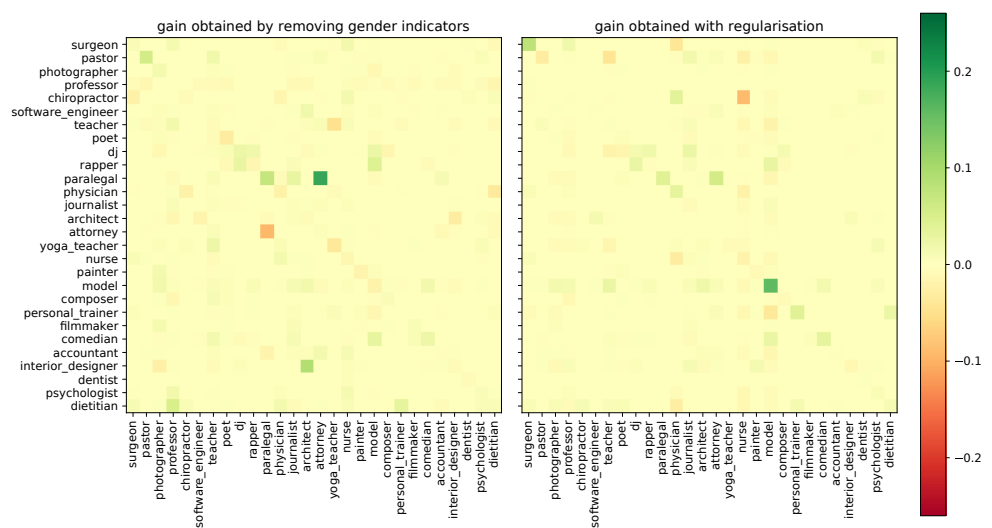


Figure A2. Extension of Figure 6 to all output classes of the Bios dataset and not only the most-frequent ones, which were selected in Figure 6 for readability purposes.

Appendix B

We present in Figure A3 the accuracy convergences for the training and the evaluation sets obtained when training a baseline and a regularised model. We kept the same axes to efficiently compare the two models. The accuracy of the regularised model was lower on the trained and evaluation set than that of the baseline on the first epochs. This difference, however, faded quickly on the following epochs. It, therefore, took longer for the regularised model to have the same accuracy as the baseline model, which is standard for regularised models. Note that we previously selected the typical amount of epochs, after which the computations stopped when using early stopping (i.e., when the accuracy continues increasing on the training set, but starts decreasing on the test set). We used this amount of epochs here and did not detect any significant overfitting on the convergence curves.

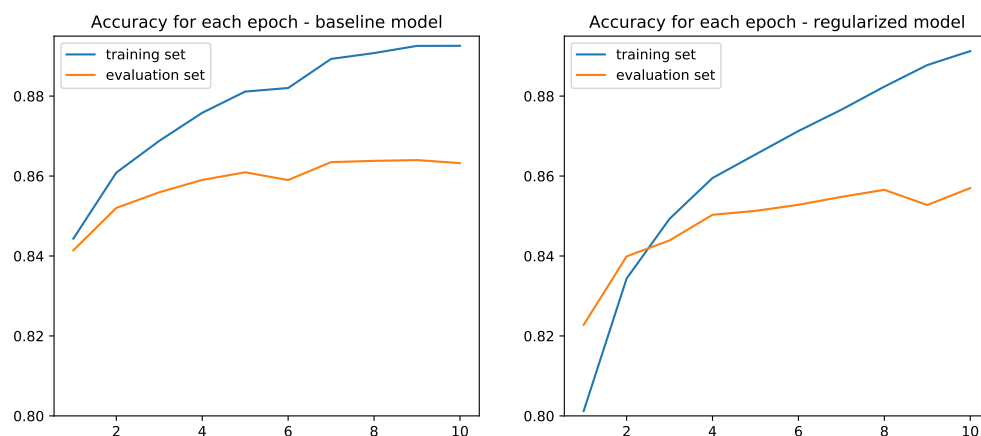


Figure A3. Learning curves with accuracy for each epoch for both models, on the training and evaluation sets.

References

- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
- Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. 2018, *Preprint*. Available online: <https://paperswithcode.com/paper/improving-language-understanding-by> (accessed on 14 March 2023).
- Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1–9.
- Besse, P.; Del Barrio, E.; Gordaliza, P.; Loubes, J.M.; Risser, L. A survey of bias in Machine Learning through the prism of Statistical Parity for the Adult Data Set. *Am. Stat.* **2022**, *76*, 188–198. [\[CrossRef\]](#)
- De Terwangne, C. Définitions clés et champ d’application du RGPD. In *Le Règlement Général sur la Protection des Données (RGPD/GDPR): Analyse Approfondie*; Larcier: Bruxelles, Belgium, 2018; pp. 59–84.
- Risser, L.; Sanz, A.G.; Vincenot, Q.; Loubes, J.M. Tackling Algorithmic Bias in Neural-Network Classifiers using 2-Wasserstein Regularization. *J. Math. Imaging Vis.* **2022**, *64*, 672–689. [\[CrossRef\]](#)
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
- De-Arteaga, M.; Romanov, A.; Wallach, H.; Chayes, J.; Borgs, C.; Chouldechova, A.; Geyik, S.; Kenthapadi, K.; Kalai, A.T. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; pp. 120–128.
- Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness through awareness. In Proceedings of the 3rd innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 214–226.
- Hardt, M.; Price, E.; Srebro, N. Equality of opportunity in supervised learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1–9.
- Verma, S.; Rubin, J. Fairness definitions explained. In Proceedings of the 2018 IEEE/ACM International Workshop on Software Fairness (Fairware), Gothenburg, Sweden, 29 May 2018; pp. 1–7.
- Kleinberg, J.; Mullainathan, S.; Raghavan, M. Inherent trade-offs in the fair determination of risk scores. *arXiv* **2016**, arXiv:1609.05807.
- Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data* **2017**, *5*, 153–163. [\[CrossRef\]](#)
- Pleiss, G.; Raghavan, M.; Wu, F.; Kleinberg, J.; Weinberger, K.Q. On fairness and calibration. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
- Skeem, J.; Lowenkamp, C. *Risk, Race, & Recidivism: Predictive Bias and Disparate Impact* (SSRN Scholarly Paper No. ID 2687339); Social Science Research Network: Rochester, NY, USA, 2015.
- Cirillo, D.; Catuara-Solarz, S.; Morey, C.; Guney, E.; Subirats, L.; Mellino, S.; Gigante, A.; Valencia, A.; Rementeria, M.J.; Chadha, A.S.; et al. Sex and gender differences and biases in artificial intelligence for biomedicine and healthcare. *NPJ Digit. Med.* **2020**, *3*, 1–11. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ali, M.; Sapiezynski, P.; Bogen, M.; Korolova, A.; Mislove, A.; Rieke, A. Discrimination through optimization: How Facebook’s Ad delivery can lead to biased outcomes. *Proc. ACM Hum.-Comput. Interact.* **2019**, *3*, 1–30. [\[CrossRef\]](#)
- Sapiezynski, P.; Ghosh, A.; Kaplan, L.; Rieke, A.; Mislove, A. Algorithms that “Don’t See Color”: Comparing Biases in Lookalike and Special Ad Audiences. *arXiv* **2019**, arXiv:1912.07579.

20. Garg, N.; Schiebinger, L.; Jurafsky, D.; Zou, J. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, E3635–E3644. [\[CrossRef\]](#)
21. Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; Chang, K.W. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv* **2018**, arXiv:1804.06876.
22. Caliskan, A.; Bryson, J.J.; Narayanan, A. Semantics derived automatically from language corpora contain human-like biases. *Science* **2017**, *356*, 183–186. [\[CrossRef\]](#)
23. Gonen, H.; Goldberg, Y. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv* **2019**, arXiv:1903.03862.
24. Sikdar, S.; Lemmerich, F.; Strohmaier, M. GetFair: Generalized Fairness Tuning of Classification Models. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 289–299.
25. Denis, C.; Elie, R.; Hebiri, M.; Hu, F. Fairness guarantee in multi-class classification. *arXiv* **2021**, arXiv:2109.13642.
26. Zhang, B.H.; Lemoine, B.; Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 2–3 February 2018; pp. 335–340.
27. Madras, D.; Creager, E.; Pitassi, T.; Zemel, R. Learning adversarially fair and transferable representations. In Proceedings of the International Conference on Machine Learning, PMLR, New Orleans, LA, USA, 1–3 February 2018; pp. 3384–3393.
28. Kamishima, T.; Akaho, S.; Asoh, H.; Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, 24–28 September 2012; pp. 35–50.
29. Manisha, P.; Gujar, S. Fnnn: Achieving fairness through neural networks. *arXiv* **2018**, arXiv:1811.00247.
30. Zafar, M.B.; Valera, I.; Gomez Rodriguez, M.; Gummadi, K.P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 May 2017; pp. 1171–1180.
31. Zafar, M.B.; Valera, I.; Rodriguez, M.G.; Gummadi, K.P. Fairness constraints: Mechanisms for fair classification. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 962–970.
32. Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; Chang, K.W. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv* **2017**, arXiv:1707.09457.
33. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311. [\[CrossRef\]](#)
34. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [\[CrossRef\]](#)
35. Chizat, L.; Roussillon, P.; Léger, F.; Vialard, F.X.; Peyré, G. Faster Wasserstein Distance Estimation with the Sinkhorn Divergence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 2257–2269.
36. Flamary, R.; Courty, N.; Gramfort, A.; Alaya, M.Z.; Boisbunon, A.; Chambon, S.; Chapel, L.; Corenflos, A.; Fatras, K.; Fournier, N.; et al. POT: Python Optimal Transport. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
37. Feydy, J.; Séjourné, T.; Vialard, F.X.; Amari, S.I.; Trounev, A.; Peyré, G. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, Naha, Japan, 16–18 April 2019; pp. 2681–2690.
38. Hand, D.J.; Yu, K. Idiot’s Bayes—Not So Stupid After All? *Int. Stat. Rev.* **2001**, *69*, 385–398.
39. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Washington, DC, USA, 4–10 August 2001; Volume 3, pp. 41–46.
40. Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; Fidler, S. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv* **2015**, arXiv:1506.06724.
41. Mackenzie, J.; Benham, R.; Petri, M.; Trippas, J.R.; Culpepper, J.S.; Moffat, A. CC-News-En: A Large English News Corpus. In Proceedings of the CIKM ’20, 29th ACM International Conference on Information & Knowledge Management, online, 19–23 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 3077–3084. [\[CrossRef\]](#)
42. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
43. Trinh, T.H.; Le, Q.V. A simple method for commonsense reasoning. *arXiv* **2018**, arXiv:1806.02847.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.