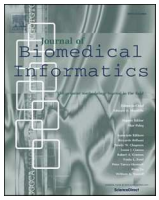
Journal of Biomedical Informatics 156 (2024) 104671

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



Special Communication

Identify and mitigate bias in electronic phenotyping: A comprehensive study from computational perspective

Sirui Ding [a,1], Shenghan Zhang [b,1], Xia Hu [c], Na Zou [d,*]

[a] Department of Computer Science & Engineering, Texas A&M University, College Station, TX, United States
[b] Department of Biomedical Informatics, Harvard University, Boston, MA, United States
[c] Department of Computer Science, Rice University, Houston, TX, United States
[d] Department of Industrial Engineering, University of Houston, Houston, TX, United States

ARTICLE INFO

Keywords:
Fairness in healthcare
Electronic phenotyping
Algorithm fairness
Bias mitigation

ABSTRACT

Electronic phenotyping is a fundamental task that identifies the special group of patients, which plays an important role in precision medicine in the era of digital health. Phenotyping provides real-world evidence for other related biomedical research and clinical tasks, e.g., disease diagnosis, drug development, and clinical trials, etc. With the development of electronic health records, the performance of electronic phenotyping has been significantly boosted by advanced machine learning techniques. In the healthcare domain, precision and fairness are both essential aspects that should be taken into consideration. However, most related efforts are put into designing phenotyping models with higher accuracy. Few attention is put on the fairness perspective of phenotyping. The neglection of bias in phenotyping leads to subgroups of patients being underrepresented which will further affect the following healthcare activities such as patient recruitment in clinical trials. In this work, we are motivated to bridge this gap through a comprehensive experimental study to identify the bias existing in electronic phenotyping models and evaluate the widely-used debiasing methods' performance on these models. We choose pneumonia and sepsis as our phenotyping target diseases. We benchmark 9 kinds of electronic phenotyping methods spanning from rule-based to data-driven methods. Meanwhile, we evaluate the performance of the 5 bias mitigation strategies covering pre-processing, in-processing, and post-processing. Through the extensive experiments, we summarize several insightful findings from the bias identified in the phenotyping and key points of the bias mitigation strategies in phenotyping.

1. Introduction

Phenotyping stands as a cornerstone in the realm of biomedical research, serving as the linchpin that enables medical practitioners to accurately pinpoint diseases [1,2], facilitates the acceleration of drug development [3], and plays a pivotal role in the meticulous design of clinical trials [4]. Its foundational significance reverberates throughout the entire healthcare ecosystem, fundamentally shaping the trajectory of patient care, research advancements, and medical innovation as illustrated in Fig. 1(b).

Riding the wave of progress in electronic health records within the biomedical domain [5], the landscape of phenotyping has undergone a remarkable transformation, driven by the integration of cutting-edge computational methodologies, including advanced statistical analyses and artificial intelligence techniques [6]. As a result, electronic phenotyping methods have consistently demonstrated their prowess, exhibiting exceptional precision and efficiency across a multitude of scenarios [7]. This evolution heralds a new era in healthcare, one where data-driven insights are poised to revolutionize medical diagnosis and treatment.

However, bias is an inevitable factor in computational-based phenotyping methods, and its implications extend to various biomedical activities, including clinical trial design [4]. Bias in phenotyping indicates the phenotyping results were affected by the sensitive attributes like gender, race, ethnicity, etc. The most common of phenotyping bias is some subgroups were underrepresented by the phenotyping method. For example, less patients may be diagnosed with a specific disease than the ground truth. In this work, we will focus on the group bias which

* Corresponding author.
E-mail address: nzou2@central.uh.edu (N. Zou).
[1] indicates equal contribution.

https://doi.org/10.1016/j.jbi.2024.104671
Received 1 October 2023; Received in revised form 26 May 2024; Accepted 5 June 2024
Available online 12 June 2024

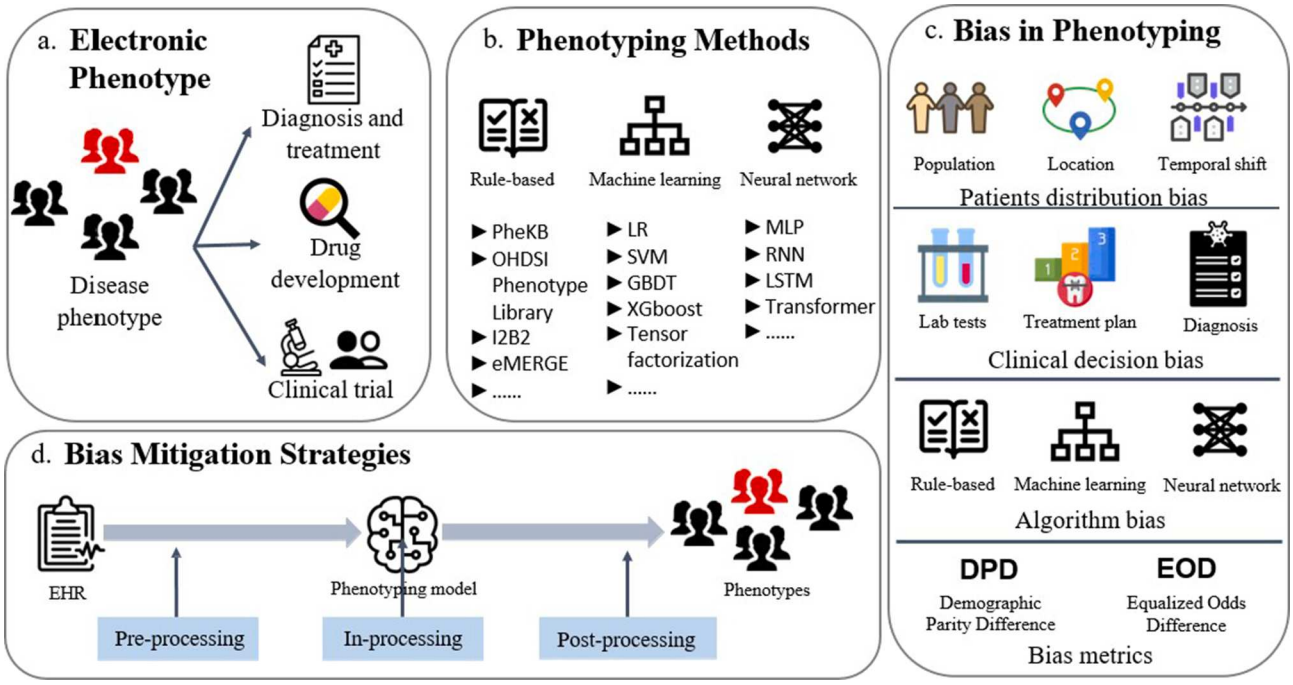


Fig. 1. Overview of the identification and mitigation of the bias in electronic phenotyping.

indicates the bias between different patient subgroups to make our contribution focused. For instance, when minority groups are underrepresented in the phenotyping process, this bias carries over to clinical trials during patient recruitment [8]. Addressing bias in electronic phenotyping poses a dual challenge for several reasons. Firstly, identifying bias from a computational standpoint is complex, as it often originates from two primary sources: data bias and model bias. Secondly, mitigating bias in electronic phenotyping and selecting appropriate debiasing techniques for different phenotype applications require careful consideration.

We are, therefore, highly motivated to embark on an extensive investigation aimed at identifying and mitigating biases within the realm of electronic phenotyping. This comprehensive study will encompass a meticulous review of prevalent electronic phenotyping methodologies, diligently scrutinizing the inherent biases within these approaches. Subsequently, we will delve into the computational aspects of bias identification and mitigation. Our research will encompass practical experiments designed to assess both the prevailing biases within existing electronic phenotyping algorithms and the efficacy of widely employed debiasing techniques. In addition, we present pivotal insights derived from our extensive experimentation. These findings encapsulate valuable knowledge and discoveries that shed light on the intricacies of electronic phenotyping and bias mitigation. By undertaking this multifaceted examination, we aim to pave the way for more equitable and unbiased electronic phenotyping practices. The contributions of this work can be succinctly summarized in three key aspects as follows:

- We benchmark and analyze the bias of 9 commonly used phenotyping models from the computational perspective.
- We evaluate 5 machine learning-based debiasing strategies for the phenotyping models. We analyze the advantages and disadvantages of each category debiasing strategy.
- We conduct extensive experiments to identify and mitigate the bias on pneumonia and sepsis phenotyping tasks and summarize insightful key findings from the experimental results.

2. Backgroundd

Electronic phenotyping can identify the patients with a specific disease, which has wide applications in disease diagnosis, treatment recommendation, clinical trial, etc as shown in Fig. 1(a). In this section, we embark on a comprehensive exploration of prevalent electronic phenotyping methods, classifying them for clarity and context. Subsequently, we delve into an insightful discussion on the latent biases that can emerge within these methods, dissecting them from both the data and model perspectives.

2.1. Phenotyping methods

We categorize the electronic phenotyping methods into 4 categories, which are rule-based, traditional machine learning, neural network, and tensor factorization as shown in Fig. 1(b).

Rule-based Method: Rule-based method is one of the most fundamental and widely applied phenotyping techniques [9,10]. The core idea is to heuristically identify the phenotypes from electronic health records by the expert-defined rules. The widely adopted rule-based methods benefit from the characteristics of interpretability, simplicity, and ease of implementation. PheKB [9] is a public rule-based phenotyping algorithm that is widely used. However, due to the human-defined rules are usually limited to some specific scenarios, they are hard to adapt to different disease or patient distributions. For example, Kho at al. [11] designed a phenotyping method specialized for type II diabetes.

Traditional Machine Learning: The main kinds of traditional machine learning include logistic regression (LR) [12], tree-based methods [13,14], and SVM [15]. These methods don't require large amounts of data in the training stage. Feature engineering [16] is an essential step for these methods to achieve competitive performance which will also require domain expertise. While traditional machine learning has found applications in various disease phenotyping tasks [17,18], Li et al. introduced Xrare [19], which leverages Gradient Boosting Decision Trees (GBDT) for diagnosing rare diseases from genetics and phenotypic data. Tensor factorization stands out as another prominent computational phenotyping method [31,32]. Ho et al. [31] propose Limestone to

generate patients' phenotypes without supervision. Afshar et al. designed a framework TASTE [32] for the temporal EHR data. Tensor factorization has the ability to break down high-dimensional patient data into more manageable low-dimensional vectors, which can then serve as phenotypes for various downstream tasks. Nonetheless, these methods still have limitations that impact their performance and adaptability. For instance, SVM is tailored for binary classification, rendering it impractical for multiclass phenotyping. LR is sensitive to outlier data [20], which is very common in EHR [21], and the tree-based model needs laborious hyper-parameter tuning for a stable performance [22].

Neural Network: With the increasing availability of electronic health records [23], neural networks have garnered significant attention in the healthcare domain due to their outstanding performance [24]. Their robust performance is primarily attributed to large-scale training data. Furthermore, the diverse architectures of neural networks facilitate seamless adaptation to various tasks; for instance, RNN-based networks effectively process temporal EHR data [25], while Transformer-based models excel in clinical text analysis [26]. However, their inherent black-box nature [27] poses a challenge in real-world applications [28,29]. Additionally, the scarcity of data in certain rare disease phenotyping tasks may render direct application of neural networks unfeasible [30].

Given the strengths and weaknesses of these electronic phenotyping methods, the selection of the most suitable approach should be tailored to the specific task and application context.

2.2. Bias in phenotyping

We conduct a comprehensive analysis of bias in electronic phenotyping, examining it from both data and model perspectives as presented in Fig. 1(c). Bias in phenotyping becomes evident when we observe variations in the method's performance across different subgroups defined by sensitive attributes like gender, race, and other factors. The origins of potential bias and their impact on phenotype outcomes will be discussed in greater detail below.

Data-level bias: The EHR data encompasses a diverse range of sources, including lab tests, diagnosis codes, treatment codes, and more. Given that electronic phenotyping methods heavily rely on data, any bias within the data significantly impacts the phenotyping outcomes. We categorize data bias into two main types: human decision bias and patient distribution bias as shown in Fig. 1 (c). Human decision bias arises from clinical judgments, where certain records, such as diagnoses [33] and treatments [34], may exhibit biases due to human clinical decisions. For instance, phenotype rules crafted by humans may inadvertently underrepresent certain subgroups [33]. On the other hand, patient distribution bias indicates an imbalance in patient representation due to disparities in cohort selection procedures. This can occur when minority patient groups are underrepresented, possibly stemming from limited access to the healthcare system [35]. It's crucial to recognize that biases at the data level inevitably permeate into the models trained on such data [36].

Model-level bias: The bias in the phenotyping model will also affect the phenotype fairness. As described in Section 2.1, the bias can be summarized into two categories. The first one is the bias in human-defined rules, which usually exists in some heuristic phenotyping methods like the rule-based method [33]. The second one is algorithm bias which commonly exists in artificial intelligence methods as shown in Fig. 1(c). The artificial intelligence algorithm will be trained toward optimal prediction accuracy while sacrificing fairness [37,38]. There will be prediction disparities between different subgroups, e.g., some subgroups will have more positive predictions, the accuracy will also be higher on some subgroups, etc [39].

The presence of bias in electronic phenotyping can lead to unfair treatment of certain patient subgroups. Moreover, the patient cohorts derived from phenotyping may introduce bias into subsequent processes, such as the recruitment of patients for clinical trials. Addressing bias in phenotyping, both at the data and model levels, represents an ongoing challenge and an area for further research.

Table 1
Statistical summary of MIMIC-III database.

	# Patients	# Healthy	# Diagnoses	# ICD codes	# Medications
MIMIC-III	38699	7821	15692	5435	1339918
Pneumonia	1419	1419	\	1606	954
Sepsis	1096	1096	\	1513	892

2.3. Bias mitigation strategies

The methods for mitigating bias can be classified into three categories: pre-processing, in-processing, and post-processing [39]. These approaches are implemented at various stages within the electronic phenotyping pipeline as demonstrated in Fig. 1(d).

Pre-processing strategy: Pre-processing method [40,41] aims to remove the bias-related information in the input data. There are two kinds of input bias-related information. The first one is sensitive features (explicit bias information) such as gender and race for which we can directly remove them. This category of method is a naïve method to reduce the bias in the field of fair machine learning. Due to the implicit bias existing in other features, directly removing the sensitive attributes cannot usually effectively reduce the bias [59]. In this paper, this method will be used as a baseline method for comparison and evaluation. The second one is implicit bias [42,43]. For example, the zip code is not a sensitive attribute but may be related to the race population. Moreover, we can resample the subgroups in the training data or re-weight each sample to mitigate the bias in training [40].

In-processing strategy: In-processing method focuses on the model training part. The in-processing method will guide the model to be trained for unbiased predictions by adding fairness-related constraints or regularization. This kind of method is the most commonly used one in the machine learning community because of its flexibility and generalizability for different scenarios and settings. One kind of in-processing method is adding regularization, e.g., neural network local interpretation during the training stage [44,45]. Another main category of the method is adversarial learning, which will train a model for prediction and another model for adversarial classification [43,46].

Post-processing strategy: The post-processing method directly processes the model outputs to force the outputs to be less biased. This method can be widely applied to various kinds of methods but it needs the patients' sensitive attributes which may be unavailable due to the private issue [47,48].

3. Datasets and methods

3.1. Datasets and tasks

Pneumonia and sepsis phenotype. We use the widely applied MIMICIII [49,50] as the dataset for the following experiments in this work. Based on the MIMIC-III, we choose pneumonia and sepsis as our target phenotyping diseases because of their significant importance [51,52].

Cohort selection. We select the target patient cohort based on their"DIAGNOSIS" feature in the"ADMISSIONS" file. We filter out 1419 patients diagnosed with pneumonia and 1096 patients diagnosed with sepsis as shown in Table 1. For the negative patients, we randomly sampled the same number as the positive patients from the neonatal patients in MIMIC-III.

Data processing. We extract the diagnostic codes and drug names from the patients' histories. We follow the data preprocessing procedures in TASTE framework [32] (TASTE framework is a recent state-of-

the-art electronic phenotyping algorithm, which can be applied to electronic health records.) to group these ICD-9 codes and medical names into higher-level categories to avoid the potential data leakage issue in the following model training. We will first convert the ICD-9 codes to the ICD-10 codes and respectively use the Clinical Classification Software (CCS) system and the Anatomical Therapeutic Chemical (ATC) classification system to transfer ICD codes and drug names into more general classifications. The large number of features from ICD codes and medications has been reduced discernibly to CCS and ATC codes. For pneumonia, we get 232 CCS codes and 285 ATC codes. For sepsis, we get 231 CCS codes and 270 ATC codes. As each patient may have multiple visits, we formulate the input containing both temporal features and static features. We formulate the input as 3D tensors [32] consisting of patients, hospital visits, and temporal attributes. For the sensitive attributes, we chose gender and race as the research targets.

3.2. Study design

In this section, we will introduce the proposed study design to comprehensively investigate and mitigate the bias in electronic phenotyping. First, we will discuss how to quantitatively identify and measure the bias in phenotyping with two bias metrics. Then, we will investigate how to mitigate the bias from the computational perspective.

Identify bias in electronic phenotyping. To investigate the bias in electronic phenotyping comprehensively, we first benchmark 4 main categories of widely used phenotyping methods as described in Section 2.1. We include **9** electronic phenotyping methods in this work, which are the rule-based method, logistic regression (LR) [12], random forest (RF) [14], SVM [15], gradient boosting decision tree (GBDT) [13], MLP [53], RNN [25], and LSTM [54]. We use the ROCAUC as metrics to measure the phenotyping accuracy and demographic parity difference (DPD), and equality odds difference (EOD) as the bias metrics. We will introduce the details of the phenotyping methods and metrics in Section 3.3.1. We will analyze different methods' performance on the phenotyping tasks and the bias respectively.

Mitigate bias from the computational perspective. We evaluate three main categories of debiasing algorithms as introduced in Section 2.3 to mitigate the phenotyping bias. We choose **2** pre-processing debias method, **2** in-processing debias method and **1** post-processing debias method. All these representative debiasing methods will be tested on the phenotyping methods described above if applicable. We will use the bias and performance metrics to investigate the mitigation effectiveness of different debiasing methods on various phenotyping algorithms.

3.3. Methods

3.3.1. Bias measure metrics

We will introduce the details of two bias metrics and their clinical meaning in the electronic phenotyping as follows. We use $\widehat{Y}$ to denote the prediction of the phenotyping model, Y to denote the true label, and S to denote the sensitive attribute of each patient, e.g., gender, race, etc.

Demographic Parity Difference (DPD): The DPD measures the disparities of positive model outputs between different subgroups as shown in the Eq. (1). In the context of phenotype, the positive outputs indicate the diagnosis of specific diseases. DPD implies the bias in the probability of diagnosis between different patient groups.

$$\boldsymbol{DPD} = |P(\widehat{Y}=1|S=0) - P(\widehat{Y}=1|S=1)| \tag{1}$$

Equalized Odds Difference (EOD): The EOD measures the disparities of true positive outcomes between different subgroups as presented in the Eq. (2). EOD measures the bias of correctly identifying patients with specific diseases or phenotypes.

$$\boldsymbol{EOD} = |P(\widehat{Y}=1|Y=1,S=0) - P(\widehat{Y}=1|Y=1,S=1)| \tag{2}$$

3.3.2. Eletronic phenotyping methods

We formulate four categories of phenotyping methods as follows.

Rule-based methods: Rule-based methods are usually human-defined *if…else…* rules, whose inputs are selected features $X_{selected}$, e.g., ICD-codes, etc. Rule-based methods can be represented as follows in general.

$$\widehat{Y} = Rules(Xselected) \tag{3}$$

The rule-based method adopted in this work is based on the PheKB.

Traditional machine learning: Traditional machine learning methods consist of training and testing stages and require feature engineering on the raw patient data Raw_{train}, Raw_{test}. We formulate the traditional machine learning phenotype pipeline as follows:

$$Xtrain, Xtest = FE(Rawtrain, Rawtest) \tag{4}$$

$$\widehat{Y}train = ML(Xtrain) \tag{5}$$

$$\widehat{Y}test = ML(Xtest) \tag{6}$$

where *ML* models can be LR, RF, GBDT, and SVM in this work.

Neural networks: Neural network needs to design the network architecture and train on large-scale data. We formulate the phenotyping method with the neural network as follows.

$$\widehat{Y}train = NN(Xtrain, \theta), Loss = l(\widehat{Y}train, Ytrain) \tag{7}$$

$$\widehat{Y}test = NN(Xtest, \theta) \tag{8}$$

where θ is the trainable parameters of the neural network and the loss function l can be binary cross entropy. We choose MLP, RNN, and LSTM these three representative models to instantiate *NN* in this work.

Tensor factorization: Tensor factorization algorithm decomposes the input data into latent factor matrices for all three dimensions. We use the latent factor matrix of patient dimension for our phenotyping task and one machine learning algorithm as the classifier. This phenotyping method can be formulated as follows.

$$Mp, Mv, Mt = TF(X), Loss = l(Mp\hat{A}\cdot Mv\hat{A}\cdot Mt, X) \tag{9}$$

$$Mtrainp, Mtestp = split(Mp) \tag{10}$$

$$\widehat{Y}test = ML(Mtestp) \tag{11}$$

where X represents the raw input and *split*() is to separate the whole dataset into train and test sets. *TF* represents the tensor factorization process, and the loss function is based on the cross entropy between the product of the resulting three latent factor matrices and the raw input data. We choose PARAFAC [55] as the tensor factorization algorithm and LR as the classifier in this work.

3.3.3. Debiasing methods

We formulate three kinds of debiasing methods as follows.

Pre-processing debias: One kind of pre-processing debias method will be operated on the input data to remove the explicit and implicit bias features. Specifically, we utilize the Pearson Correlation Coefficient (PCC) to determine the level of correlation between the two variables and set a threshold to remove strongly correlated features that exceed this threshold. This process can be presented as follows.

$$Xdebias = Remover(X, threshold) \tag{12}$$

$$\widehat{Y} = Model(Xdebias) \tag{13}$$

where *Remover*() is the algorithm that removes the sensitive related features, for which we choose correlation remover in this work. The threshold is manually set for determining if the feature should be

Table 2
Debiasing results of pneumonia phenotyping (Demographic Parity Difference. Correlation remover and resample are the pre-processing methods. Reduction and adversarial mitigation are the in-processing methods. Threshold optimizer is a post-processing method. w/o debias is the baseline method without any debiasing strategy.).

Disease phenotyping		Rule Based	Machine Learning				Tensor Factorization	Deep Learning		
Sensitive Attribute	Debias Method	PheKB-ICD	LR	RF	SVM	GBC	PARAFAC + LR	MLP	RNN	LSTM
Gender (Input include)	Correlation Remover	0.000	0.031 ± 0.001	0.036 ± 0.001	0.047 ± 0.001	0.037 ± 0.001	0.037 ± 0.001	0.041 ± 0.001	0.041 ± 0.001	0.040 ± 0.001
	Resample	0.010	0.036 ± 0.001	0.040 ± 0.001	0.043 ± 0.001	0.038 ± 0.001	0.037 ± 0.001	0.038 ± 0.001	0.035 ± 0.002	0.037 ± 0.001
	Reduction	/	0.039 ± 0.001	0.036 ± 0.001	0.039 ± 0.001	0.037 ± 0.001	0.047 ± 0.001	0.036 ± 0.001	0.036 ± 0.001	0.035 ± 0.001
	Threshold Optimizer	0.000	0.049 ± 0.001	0.045 ± 0.000	0.052 ± 0.001	0.045 ± 0.001	0.048 ± 0.001	0.053 ± 0.001	0.043 ± 0.001	0.040 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.043 ± 0.001	0.038 ± 0.001	0.087 ± 0.014
	w/o debias	0.000	0.031 ± 0.001	0.036 ± 0.001	0.047 ± 0.001	0.037 ± 0.001	0.037 ± 0.001	0.041 ± 0.001	0.064 ± 0.001	0.040 ± 0.001
Race (Input include)	Correlation Remover	0.006	0.127 ± 0.002	0.141 ± 0.001	0.039 ± 0.000	0.142 ± 0.001	0.131 ± 0.001	0.146 ± 0.001	0.150 ± 0.001	0.148 ± 0.001
	Resample	0.036	0.024 ± 0.001	0.028 ± 0.000	0.024 ± 0.000	0.027 ± 0.000	0.045 ± 0.003	0.026 ± 0.000	0.021 ± 0.000	0.026 ± 0.000
	Reduction	/	0.082 ± 0.002	0.088 ± 0.001	0.052 ± 0.002	0.098 ± 0.001	0.049 ± 0.001	0.074 ± 0.001	0.064 ± 0.002	0.060 ± 0.002
	Threshold Optimizer	0.001	0.034 ± 0.001	0.034 ± 0.001	0.026 ± 0.000	0.036 ± 0.002	0.029 ± 0.001	0.030 ± 0.000	0.023 ± 0.000	0.028 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.141 ± 0.001	0.169 ± 0.002	0.146 ± 0.001
	w/o debias	0.006	0.127 ± 0.002	0.141 ± 0.001	0.039 ± 0.000	0.142 ± 0.001	0.131 ± 0.001	0.148 ± 0.001	0.145 ± 0.001	0.072 ± 0.022
Gender (Input exclude)	Correlation Remover	0.000	0.037 ± 0.001	0.037 ± 0.001	0.037 ± 0.000	0.037 ± 0.001	0.050 ± 0.001	0.036 ± 0.001	0.043 ± 0.001	0.043 ± 0.001
	Resample	0.010	0.038 ± 0.001	0.038 ± 0.001	0.038 ± 0.001	0.038 ± 0.001	0.039 ± 0.001	0.039 ± 0.001	0.040 ± 0.001	0.038 ± 0.001
	Reduction	/	0.037 ± 0.001	0.037 ± 0.001	0.037 ± 0.000	0.037 ± 0.001	0.050 ± 0.001	0.037 ± 0.001	0.040 ± 0.001	0.042 ± 0.001
	Threshold Optimizer	0.000	0.047 ± 0.001	0.037 ± 0.001	0.047 ± 0.001	0.043 ± 0.001	0.060 ± 0.001	0.041 ± 0.001	0.044 ± 0.001	0.040 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.041 ± 0.001	0.035 ± 0.001	0.035 ± 0.000
	w/o debias	0.000	0.037 ± 0.001	0.037 ± 0.001	0.037 ± 0.001	0.037 ± 0.001	0.050 ± 0.001	0.036 ± 0.001	0.043 ± 0.001	0.041 ± 0.001
Race (Input exclude)	Correlation Remover	0.006	0.142 ± 0.001	0.142 ± 0.001	0.142 ± 0.0009	0.142 ± 0.001	0.103 ± 0.000	0.141 ± 0.001	0.138 ± 0.001	0.141 ± 0.000
	Resample	0.036	0.027 ± 0.000	0.028 ± 0.001	0.028 ± 0.001	0.027 ± 0.000	0.043 ± 0.001	0.028 ± 0.001	0.033 ± 0.001	0.030 ± 0.001
	Reduction	/	0.126 ± 0.002	0.144 ± 0.002	0.125 ± 0.001	0.123 ± 0.001	0.101 ± 0.001	0.135 ± 0.001	0.136 ± 0.000	0.138 ± 0.000
	Threshold Optimizer	0.001	0.025 ± 0.000	0.027 ± 0.001	0.043 ± 0.000	0.041 ± 0.000	0.019 ± 0.000	0.033 ± 0.000	0.044 ± 0.000	0.035 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.125 ± 0.001	0.135 ± 0.001	0.110 ± 0.004
	w/o debias	0.006	0.142 ± 0.001	0.142 ± 0.001	0.142 ± 0.001	0.142 ± 0.001	0.103 ± 0.000	0.141 ± 0.001	0.141 ± 0.001	0.141 ± 0.001

eliminated.

Another pre-processing debias method is resampling, which resample the ratio of different subgroups to make the balance of them. The process can be represented as follows.

$$Xresample = Resampler(X, S) \tag{14}$$

$$\widehat{Y} = Model(Xresample) \tag{15}$$

In-processing debias: In-processing debias method performs during the model training stage. One method to guide the model to be trained toward fair predictions is by adding some fairness constraints, which is one kind of widely applied method. In our experiment, the classification reduction algorithm [56] is adopted for this guiding. Typically, the objective of this algorithm is to minimize the disparity in prediction between different groups during the training process. This process can be presented as follows.

$$\widehat{Y} = Model(X) \tag{16}$$

$$Loss = l(\widehat{Y}train, Ytrain) + fairnessconstraint \tag{17}$$

where *fairness constraint* is the regularization that ensures the predic-tion fairness, for which we use demographic parity constraint in this work.

Another mainstream of the in-processing debiasing method is adversarial learning, which will train a predictor model and an adversary model. The predictor model will be trained with conventional strategy as shown below.

$$\widehat{Y}train = Predictor(Xtrain, \theta P), LP = l(\widehat{Y}train, Ytrain) \tag{18}$$

Meanwhile, an adversary model will be trained to predict the sensitive attributes based on the predictions from the *Predictor* model. This process can be formulated as follows:

Table 3
Debiasing results of pneumonia phenotyping (Equalized Odds Difference. w/o debias is the baseline method without any debiasing strategy.).

Disease phenotyping		Rule Based	Machine Learning				Tensor Factorization	Deep Learning		
Sensitive Attribute	Debias Method	PheKB-ICD	LR	RF	SVM	GBC	PARAFAC + LR	MLP	RNN	LSTM
Gender (Input included)	Correlation Remover	0.007	0.030 ± 0.000	0.008 ± 0.000	0.055 ± 0.002	0.000 ± 0.000	0.024 ± 0.000	0.007 ± 0.000	0.010 ± 0.000	0.006 ± 0.000
	Resample	0.019	0.009 ± 0.000	0.007 ± 0.000	0.025 ± 0.000	0.000 ± 0.000	0.016 ± 0.000	0.007 ± 0.000	0.010 ± 0.000	0.007 ± 0.000
	Reduction	/	0.025 ± 0.000	0.008 ± 0.000	0.039 ± 0.001	0.000 ± 0.000	0.029 ± 0.001	0.010 ± 0.000	0.017 ± 0.000	0.013 ± 0.000
	Threshold Optimizer	0.005	0.039 ± 0.001	0.061 ± 0.001	0.052 ± 0.002	0.055 ± 0.001	0.037 ± 0.000	0.050 ± 0.001	0.045 ± 0.001	0.043 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.015 ± 0.000	0.010 ± 0.000	0.146 ± 0.084
	w/o debias	0.007	0.030 ± 0.000	0.008 ± 0.000	0.055 ± 0.002	0.000 ± 0.000	0.024 ± 0.000	0.007 ± 0.000	0.011 ± 0.000	0.006 ± 0.000
Race (Input included)	Correlation Remover	0.048	0.064 ± 0.000	0.010 ± 0.000	0.121 ± 0.003	0.000 ± 0.000	0.049 ± 0.000	0.020 ± 0.000	0.024 ± 0.000	0.021 ± 0.000
	Resample	0.072	0.053 ± 0.003	0.009 ± 0.000	0.074 ± 0.004	0.000 ± 0.000	0.107 ± 0.003	0.021 ± 0.000	0.028 ± 0.000	0.043 ± 0.001
	Reduction	/	0.102 ± 0.001	0.093 ± 0.001	0.116 ± 0.002	0.074 ± 0.001	0.140 ± 0.002	0.110 ± 0.001	0.125 ± 0.003	0.135 ± 0.003
	Threshold Optimizer	0.048	0.224 ± 0.004	0.246 ± 0.001	0.210 ± 0.005	0.252 ± 0.004	0.234 ± 0.002	0.215 ± 0.002	0.243 ± 0.001	0.238 ± 0.002
	Adversarial Mitigation	/	/	/	/	/	/	0.016 ± 0.000	0.047 ± 0.005	0.146 ± 0.001
	w/o debias	0.048	0.064 ± 0.000	0.010 ± 0.000	0.121 ± 0.003	0.000 ± 0.000	0.049 ± 0.000	0.018 ± 0.000	0.021 ± 0.000	0.024 ± 0.000
Gender (Input excluded)	Correlation Remover	0.007	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.040 ± 0.002	0.001 ± 0.000	0.017 ± 0.000	0.020 ± 0.000
	Resample	0.019	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.050 ± 0.000	0.002 ± 0.000	0.015 ± 0.000	0.006 ± 0.000
	Reduction	/	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.040 ± 0.002	0.000 ± 0.000	0.019 ± 0.000	0.020 ± 0.001
	Threshold Optimizer	0.005	0.046 ± 0.002	0.049 ± 0.000	0.040 ± 0.001	0.038 ± 0.001	0.053 ± 0.001	0.031 ± 0.001	0.039 ± 0.000	0.027 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.020 ± 0.000	0.015 ± 0.000	0.016 ± 0.000
	w/o debias	0.007	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.040 ± 0.002	0.001 ± 0.000	0.020 ± 0.000	0.017 ± 0.000
Race (Input excluded)	Correlation Remover	0.048	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.028 ± 0.001	0.001 ± 0.000	0.010 ± 0.000	0.013 ± 0.000
	Resample	0.072	0.000 ± 0.000	0.003 ± 0.000	0.003 ± 0.000	0.000 ± 0.000	0.058 ± 0.000	0.003 ± 0.000	0.023 ± 0.000	0.018 ± 0.000
	Reduction	/	0.026 ± 0.000	0.018 ± 0.000	0.022 ± 0.001	0.026 ± 0.000	0.052 ± 0.000	0.019 ± 0.000	0.017 ± 0.000	0.022 ± 0.000
	Threshold Optimizer	0.048	0.247 ± 0.001	0.257 ± 0.001	0.241 ± 0.002	0.236 ± 0.003	0.125 ± 0.000	0.216 ± 0.001	0.230 ± 0.004	0.195 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.036 ± 0.001	0.017 ± 0.000	0.010 ± 0.000
	w/o debias	0.048	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.028 ± 0.001	0.001 ± 0.000	0.012 ± 0.000	0.012 ± 0.000

$$\widehat{S}train = AdverseNN(\widehat{Y}train, \theta A), LA = l(\widehat{S}train, Strain) \tag{19}$$

The overall optimization goal is combining two losses of predictor network and adversary model as follows.

$$L = \alpha LP + \beta LA \tag{20}$$

where α, β are hyper-parameters that control the ratio of two losses. In this work, we use the adversarial debiasing method proposed by Zhang et al. [57].

Post-processing debias: Post-processing debiasing method directly calibrates the model outputs, which can be formulated as:

$$\widehat{Y} = Model(X) \tag{21}$$

$$\widehat{Y}cal = Calibrator(\widehat{Y}) \tag{22}$$

In this work, a threshold-based post-processing technique is employed as a method of calibration, based on the principle of equality of opportunity in model predictions, as articulated by Hardt et al. [48]. The threshold-based post-processing method adjusts a predictive model's decision boundary for different groups to meet fairness constraints like EOD or DPD. It tests various thresholds on the model's output and selects the one that best balances fairness and performance. Technically, we use the ThresholdOptimizer class from Fairlearn library to achieve this.

3.4. Implementation details

This section introduces the implementation details of different types of rule-based and machine learning models.

Algorithm implementation: In our experiment, all algorithmic implementations have been actualized within the Python 3.8 environment. We leverage the rule-based algorithms available from the Phenotype Knowledgebase (PheKB) [9] community as part of our analytical framework on the MIMIC-III. Furthermore, we leverage the scikit-learn library to implement traditional machine learning methodologies, employing the tensorly library for tensor factorization

Table 4
Debiasing results of sepsis phenotyping (Demographic Parity Difference. w/o debias is the baseline method without any debiasing strategy.).

Disease phenotyping		Rule Based	Machine Learning				Tensor Factorization	Deel Learning		
Sensitive Attribute	Debias Method	PheKB-ICD	LR	RF	SVM	GBC	PARAFAC + LR	MLP	RNN	LSTM
Gender (Input include)	Correlation Remover	0.007	0.017 ± 0.000	0.012 ± 0.000	0.019 ± 0.000	0.014 ± 0.000	0.018 ± 0.000	0.018 ± 0.000	0.019 ± 0.000	0.016 ± 0.000
	Resample	0.001	0.041 ± 0.001	0.032 ± 0.001	0.042 ± 0.001	0.033 ± 0.000	0.034 ± 0.000	0.033 ± 0.000	0.037 ± 0.000	0.035 ± 0.001
	Reduction	/	0.021 ± 0.000	0.012 ± 0.000	0.025 ± 0.000	0.014 ± 0.000	0.042 ± 0.001	0.019 ± 0.000	0.016 ± 0.000	0.019 ± 0.000
	Threshold Optimizer	0.000	0.023 ± 0.000	0.015 ± 0.000	0.060 ± 0.003	0.017 ± 0.000	0.024 ± 0.000	0.022 ± 0.000	0.023 ± 0.000	0.027 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.027 ± 0.000	0.012 ± 0.000	0.015 ± 0.000
	w/o debias	0.007	0.017 ± 0.000	0.012 ± 0.000	0.019 ± 0.000	0.014 ± 0.000	0.018 ± 0.000	0.019 ± 0.000	0.016 ± 0.000	0.016 ± 0.000
Race (Input include)	Correlation Remover	0.140	0.122 ± 0.001	0.149 ± 0.001	0.051 ± 0.002	0.148 ± 0.001	0.135 ± 0.002	0.163 ± 0.001	0.163 ± 0.001	0.160 ± 0.001
	Resample	0.029	0.039 ± 0.001	0.028 ± 0.000	0.043 ± 0.002	0.037 ± 0.000	0.021 ± 0.000	0.024 ± 0.000	0.020 ± 0.000	0.022 ± 0.000
	Reduction	/	0.091 ± 0.003	0.094 ± 0.003	0.061 ± 0.003	0.092 ± 0.003	0.060 ± 0.001	0.068 ± 0.002	0.072 ± 0.002	0.066 ± 0.002
	Threshold Optimizer	0.004	0.041 ± 0.001	0.033 ± 0.000	0.028 ± 0.001	0.053 ± 0.001	0.029 ± 0.000	0.035 ± 0.001	0.047 ± 0.001	0.047 ± 0.002
	Adversarial Mitigation	/	/	/	/	/	/	0.158 ± 0.001	0.156 ± 0.002	0.166 ± 0.001
	w/o debias	0.140	0.122 ± 0.001	0.148 ± 0.001	0.051 ± 0.002	0.148 ± 0.001	0.135 ± 0.002	0.163 ± 0.001	0.163 ± 0.001	0.162 ± 0.001
Gender (Input exclude)	Correlation Remover	0.007	0.015 ± 0.000	0.014 ± 0.000	0.015 ± 0.000	0.015 ± 0.000	0.063 ± 0.002	0.014 ± 0.000	0.017 ± 0.000	0.019 ± 0.000
	Resample	0.001	0.034 ± 0.000	0.034 ± 0.000	0.034 ± 0.000	0.034 ± 0.000	0.050 ± 0.001	0.033 ± 0.000	0.038 ± 0.001	0.038 ± 0.002
	Reduction	/	0.015 ± 0.000	0.014 ± 0.000	0.015 ± 0.000	0.015 ± 0.000	0.063 ± 0.002	0.014 ± 0.000	0.017 ± 0.000	0.019 ± 0.000
	Threshold Optimizer	0.000	0.026 ± 0.000	0.025 ± 0.000	0.020 ± 0.000	0.019 ± 0.000	0.050 ± 0.001	0.016 ± 0.000	0.026 ± 0.000	0.020 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.013 ± 0.000	0.010 ± 0.000	0.023 ± 0.000
	w/o debias	0.007	0.015 ± 0.000	0.014 ± 0.000	0.015 ± 0.000	0.015 ± 0.000	0.063 ± 0.002	0.014 ± 0.000	0.019 ± 0.000	0.018 ± 0.000
Race (Input exclude)	Correlation Remover	0.140	0.149 ± 0.001	0.148 ± 0.001	0.149 ± 0.001	0.149 ± 0.001	0.138 ± 0.003	0.148 ± 0.001	0.155 ± 0.001	0.156 ± 0.001
	Resample	0.029	0.035 ± 0.000	0.034 ± 0.000	0.034 ± 0.000	0.035 ± 0.000	0.058 ± 0.001	0.035 ± 0.000	0.026 ± 0.000	0.026 ± 0.000
	Reduction	/	0.127 ± 0.000	0.147 ± 0.001	0.138 ± 0.000	0.125 ± 0.002	0.127 ± 0.001	0.149 ± 0.001	0.156 ± 0.002	0.154 ± 0.001
	Threshold Optimizer	0.004	0.021 ± 0.000	0.044 ± 0.002	0.018 ± 0.000	0.055 ± 0.000	0.032 ± 0.000	0.036 ± 0.001	0.046 ± 0.001	0.032 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.141 ± 0.001	0.140 ± 0.001	0.149 ± 0.002
	w/o debias	0.140	0.149 ± 0.001	0.148 ± 0.001	0.149 ± 0.001	0.149 ± 0.001	0.138 ± 0.003	0.148 ± 0.001	0.155 ± 0.001	0.154 ± 0.001

algorithms and the PyTorch library for the development and deployment of our neural network models. For the critical task of debiasing methods, we call upon the 0.9.0 version of the fairlearn library for the traditional machine learning implementation. However, in instances where the fairlearn library does not provide any support, we undertake the development of our own debias procedures for our models.

Model and training detail: In pursuit of robust and reliable results during the training phase, we rigorously employ a 5-fold cross-validation methodology, thereby facilitating the robust estimation of our measuring metrics. In configuring the training hyperparameters, we set the maximum iterations for logistic regression (LR) and support vector machines (SVM) to 120, while opting for a total of 30 estimators for tree-based models(Random Forest, Gradient Boosting Classifier). The maximum iterations in tensor factorization are set 100 for PARAFAC and 400 for LR's prediction. Meanwhile, we select the top 20 features from the latent factor matrix. The hidden size of both LSTM and RNN models is set to 128. For neural network models, we deliberately define key parameters, specifying a learning rate of $1e-04$, a minibatch size of 256, and an epoch number of 40 to ensure convergence and effective training. Additionally, in the context of tensor factorization, we establish the rank of the latent factor matrix at 20. The hyperparameter of debiasing strategies are as follows. The threshold of the correlation remover strategy is set to 0.5. The resample strategy used down-sampling mode. For all the fairness constraint, we use the demographic parity as the implementation.

4. Results

We analyze the experiment results from two perspectives. The first one is bias measurement in the phenotyping. The other is how the debiasing algorithms perform.

4.1. Bias measurement in phenotyping

We summarize several key findings from the bias measurement results in two diseases phenotyping as follows.

Table 5
Debiasing results of sepsis phenotyping (Equalized Odds Difference. w/o debias is the baseline method without any debiasing strategy.).

Disease phenotyping		Rule Based	Machine Learning				Tensor Factorization	Deep Learning		
Sensitive Attribute	Debias Method	PheKB-ICD	LR	RF	SVM	GBC	PARAFAC + LR	MLP	RNN	LSTM
Gender (Input included)	Correlation Remover	0.005	0.037 ± 0.000	0.010 ± 0.000	0.085 ± 0.001	0.004 ± 0.000	0.027 ± 0.000	0.011 ± 0.000	0.011 ± 0.000	0.009 ± 0.000
	Resample	0.004	0.026 ± 0.000	0.014 ± 0.000	0.062 ± 0.002	0.004 ± 0.000	0.036 ± 0.000	0.013 ± 0.000	0.015 ± 0.000	0.013 ± 0.000
	Reduction	/	0.015 ± 0.000	0.010 ± 0.000	0.049 ± 0.001	0.004 ± 0.000	0.058 ± 0.001	0.009 ± 0.000	0.029 ± 0.000	0.025 ± 0.000
	Threshold Optimizer	0.010	0.029 ± 0.000	0.014 ± 0.000	0.121 ± 0.002	0.012 ± 0.000	0.027 ± 0.000	0.019 ± 0.000	0.020 ± 0.000	0.020 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.046 ± 0.002	0.016 ± 0.000	0.007 ± 0.000
	w/o debias	0.005	0.037 ± 0.000	0.010 ± 0.000	0.085 ± 0.001	0.004 ± 0.000	0.027 ± 0.000	0.011 ± 0.000	0.013 ± 0.000	0.009 ± 0.000
Race (Input included)	Correlation Remover	0.093	0.093 ± 0.001	0.010 ± 0.000	0.129 ± 0.007	0.006 ± 0.000	0.063 ± 0.003	0.038 ± 0.001	0.049 ± 0.003	0.038 ± 0.002
	Resample	0.062	0.056 ± 0.004	0.010 ± 0.000	0.087 ± 0.004	0.007 ± 0.000	0.111 ± 0.003	0.048 ± 0.000	0.063 ± 0.001	0.063 ± 0.001
	Reduction	/	0.100 ± 0.002	0.091 ± 0.001	0.090 ± 0.002	0.096 ± 0.002	0.150 ± 0.001	0.141 ± 0.001	0.129 ± 0.001	0.142 ± 0.001
	Threshold Optimizer	0.187	0.251 ± 0.003	0.266 ± 0.000	0.130 ± 0.005	0.241 ± 0.003	0.220 ± 0.000	0.265 ± 0.002	0.255 ± 0.004	0.264 ± 0.006
	Adversarial Mitigation	/	/	/	/	/	/	0.039 ± 0.003	0.040 ± 0.003	0.043 ± 0.001
	w/o debias	0.093	0.093 ± 0.001	0.010 ± 0.000	0.129 ± 0.007	0.006 ± 0.000	0.063 ± 0.003	0.038 ± 0.001	0.042 ± 0.002	0.042 ± 0.002
Gender (Input excluded)	Correlation Remover	0.005	0.002 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.091 ± 0.006	0.000 ± 0.000	0.029 ± 0.000	0.033 ± 0.000
	Resample	0.004	0.002 ± 0.000	0.006 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.093 ± 0.000	0.004 ± 0.000	0.027 ± 0.000	0.031 ± 0.000
	Reduction	/	0.002 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.091 ± 0.006	0.000 ± 0.000	0.028 ± 0.000	0.032 ± 0.000
	Threshold Optimizer	0.010	0.024 ± 0.000	0.032 ± 0.001	0.024 ± 0.000	0.013 ± 0.000	0.068 ± 0.002	0.004 ± 0.000	0.041 ± 0.000	0.035 ± 0.000
	Adversarial Mitigation	/	/	/	/	/	/	0.027 ± 0.000	0.012 ± 0.000	0.028 ± 0.000
	w/o debias	0.005	0.002 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.091 ± 0.006	0.000 ± 0.000	0.034 ± 0.000	0.034 ± 0.000
Race (Input excluded)	Correlation Remover	0.093	0.004 ± 0.000	0.001 ± 0.000	0.004 ± 0.000	0.004 ± 0.000	0.090 ± 0.004	0.002 ± 0.000	0.039 ± 0.001	0.045 ± 0.001
	Resample	0.062	0.004 ± 0.000	0.007 ± 0.000	0.007 ± 0.000	0.004 ± 0.000	0.081 ± 0.002	0.004 ± 0.000	0.040 ± 0.001	0.029 ± 0.000
	Reduction	/	0.047 ± 0.001	0.014 ± 0.000	0.044 ± 0.001	0.027 ± 0.000	0.095 ± 0.003	0.017 ± 0.000	0.044 ± 0.002	0.038 ± 0.001
	Threshold Optimizer	0.187	0.229 ± 0.003	0.253 ± 0.005	0.225 ± 0.001	0.280 ± 0.005	0.134 ± 0.001	0.264 ± 0.001	0.232 ± 0.002	0.255 ± 0.001
	Adversarial Mitigation	/	/	/	/	/	/	0.015 ± 0.000	0.022 ± 0.000	0.032 ± 0.001
	w/o debias	0.093	0.004 ± 0.000	0.001 ± 0.000	0.004 ± 0.000	0.004 ± 0.000	0.090 ± 0.004	0.002 ± 0.000	0.043 ± 0.001	0.041 ± 0.001

- **The electronic phenotyping bias across races is more significant than genders.** From Table 2, we can observe the race DPD bias is about 7 % higher than the gender bias on average across different phenotyping methods without debiasing strategies. From Table 4, the race DPD bias is over 12 % larger than gender bias. Similarly, when the bias metric is EOD, the race bias is 2 %, 3 % higher than gender bias on pneumonia and sepsis phenotyping respectively according to the Table 3 and Table 5. The potential reason behind is the patient distributions across different races is very diverse than the distributions across genders. There are also more salient differences on some social determinants like economic condition, workload, etc among races than genders. The inner high diverse distributions of different races than genders cause the larger bias measured in the experiments.
- **Phenotyping bias varies across different phenotyping algorithms. Rule-based Phenotyping method shows significantly less bias.**

From Table 2 and Table 4, we can find different phenotyping method presents various levels of bias under different settings. When sensitive attribute is included, in pneumonia phenotyping, RNN shows the highest gender bias, and MLP has the highest racial bias. For the sepsis phenotyping, SVM and MLP present the highest gender bias when gender is not included in the input. Meanwhile, MLP and RNN have the highest racial bias in sepsis phenotyping. When we exclude the sensitive attribute from input features, the RNN presents the largest gender bias while 4 ML models show the highest racial bias in both pneumonia and sepsis phenotyping. Moreover, we find the rule-based method presents significantly less bias compared to other methods. In the pneumonia phenotyping, rule-based method shows 4 % lower gender bias and 11 % lower race bias in terms of DPD. Noticeably, observed from Table 3 and Table 5, while the EOD gap in gender is 1 % lower, the average race bias of rule-based algorithms is 1 % higher. The bias varies on different phenotyping algorithm is probably caused by the computational mechanisms behind each algorithm is different. For example, the rule-based method will only rely on several expert selected features, which

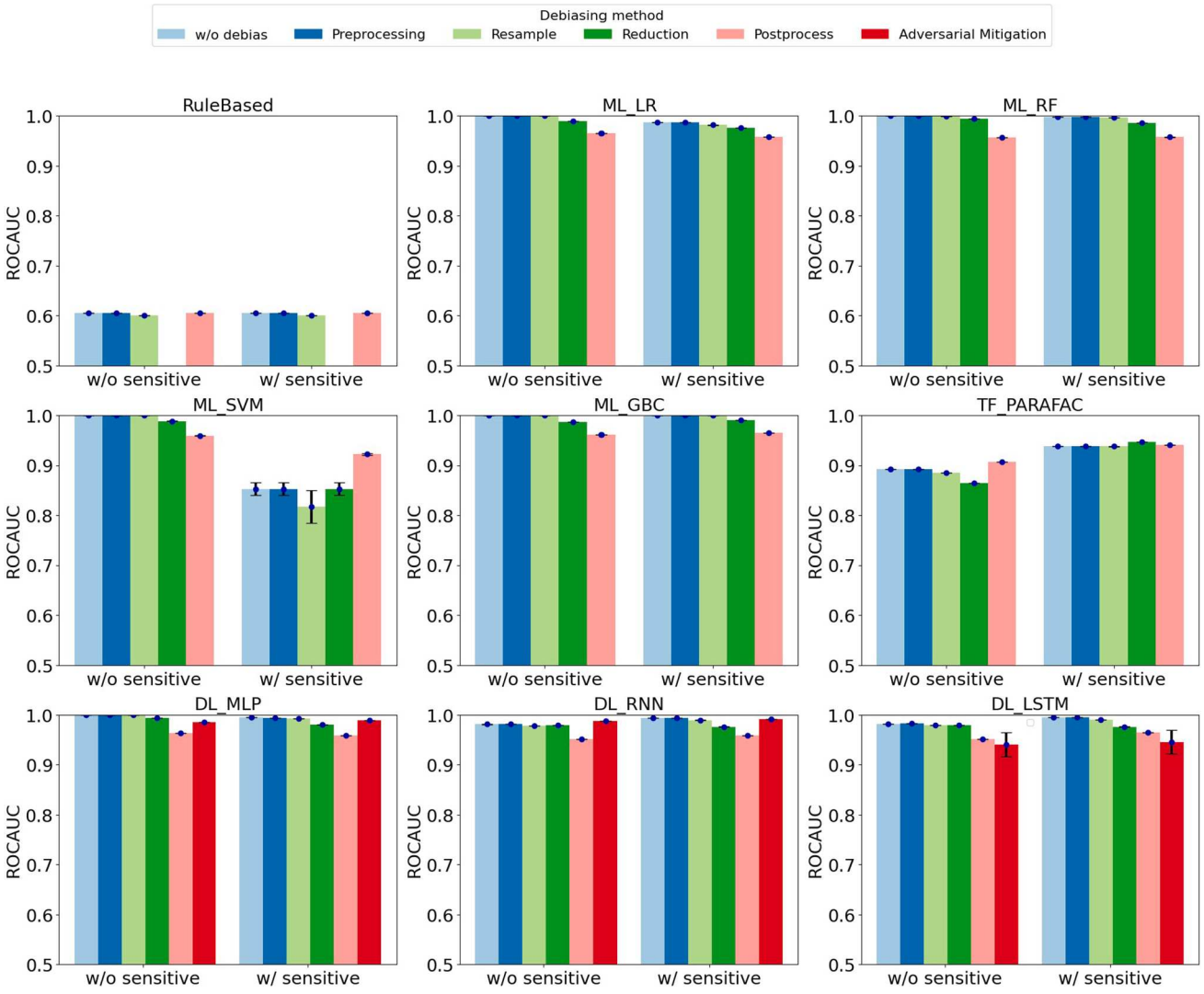


Fig. 2. Pneumonia phenotype performance. (w/o debias is the baseline method without any debiasing strategy. w/o sensitive represents omit the sensitive attributes from input features. w/ sensitive represents the input features includes the sensitive attributes.).

include less feature as well as less bias among the input data.

- **Exclude the sensitive attributes from input data has a trivial effect on the bias.** Excluding the sensitive attributes is the most intuitive method to mitigate the bias from the observation in Table 2, 3, 4, and 5. However, we find that after excluding the sensitive attributes, the bias is still significant. In pneumonia phenotyping, the gender bias of LR, RF, GBC, and LSTM increases after the gender feature is excluded, and the racial bias of LR, RF, SVM, LSTM increases after the race is excluded. In sepsis phenotyping, RF, GBC, RNN, and LSTM's gender bias increases. SVM, GBC's racial bias increases after the race attribute is excluded. The reason behind is other features contains the implicit information related to the sensitive attributes. So directly removing the sensitive attributes may not work effectively.

4.1.1. *Performance of debiasing algorithms*

- **The debiasing strategies are more effective on racial bias than gender bias.** From the gender part in both Table 2 and Table 4, we can find that the gender bias decreases with a non-trivial level. The reason may be the bias across genders is relatively small and trivial. So the debiasing method couldn't effectively mitigate the gender bias. However, most debiasing methods can reduce race bias significantly. For example, the race bias of MLP in pneumonia phenotyping is reduced by 12.2 % with the resample debiasing method. The race bias of SVM can be further reduced by 1.5 % with resample strategy in the sepsis phenotyping task. This is because the racial bias is more salient than the gender bias, so the debiasing strategies show better performance on the racial bias.

- **Correlation removing method is not capable of mitigating the bias in phenotyping.** We can observe from Table 2 and Table 4, that removing the sensitive correlation from input features doesn't work for the sepsis and pneumonia phenotyping. For pneumonia phenotyping, the gender bias even increases a bit after the correlation removal in DL methods of **LSTM**. The race bias of **MLP** and **RNN** increases after the correlation removal. The situation is similar in the sepsis phenotype. This may be caused by the input feature containing little information related to the sensitive attributes. This phenomenon further shows the bias in phenotyping is more likely to be caused by the phenotyping algorithm and data distribution. Neither the removal of sensitive attributes and correlation among other features has the satisfying debiasing performance.

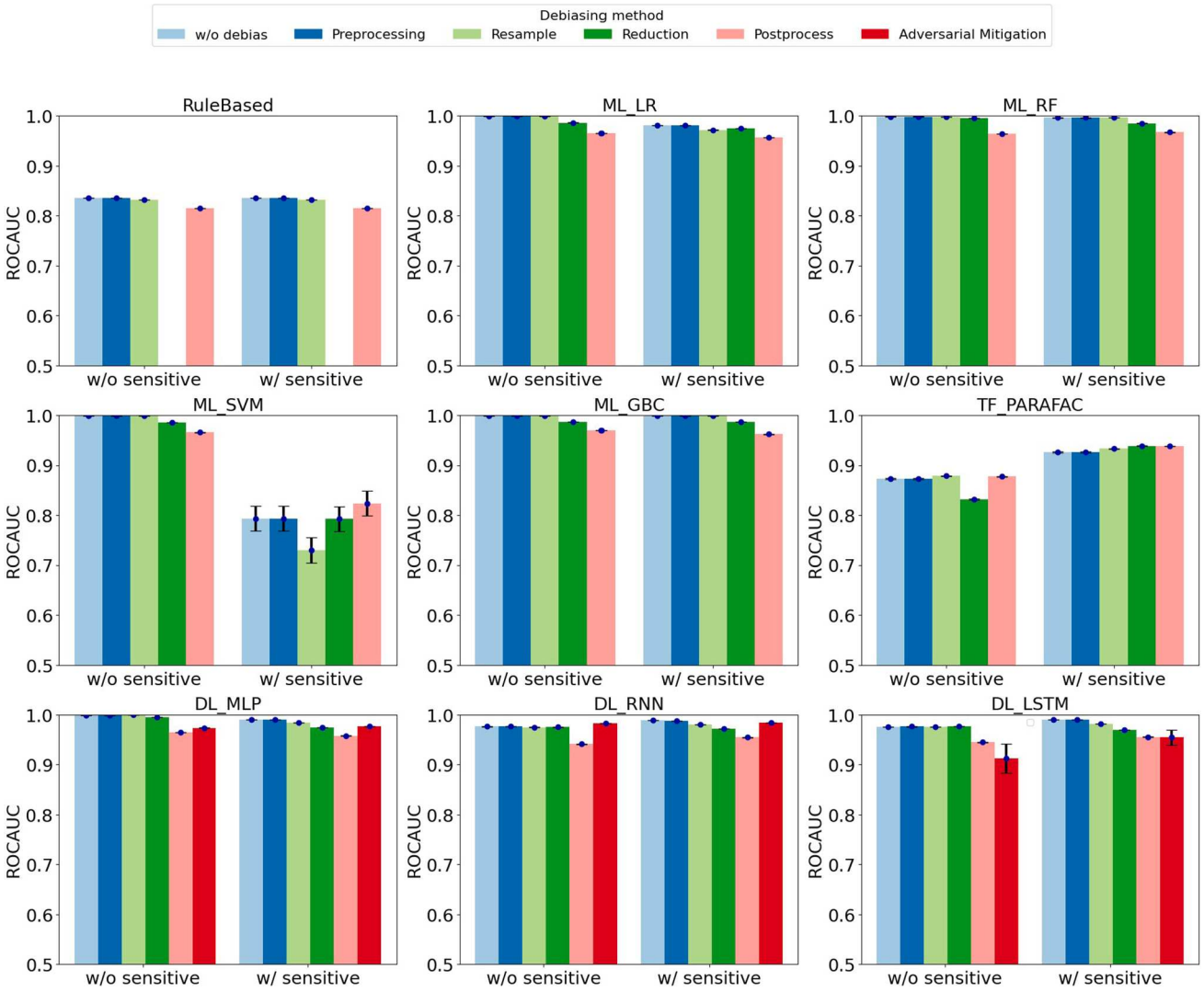


Fig. 3. Sepsis phenotype performance. (w/o debias is the baseline method without any debiasing strategy. w/o sensitive represents omit the sensitive attributes from input features. w/ sensitive represents the input features includes the sensitive attributes.).

- **Resample the patients' data and postprocess the outputs are two very simple yet effective debiasing methods.** From Table 2 and Table 4, we can find resampling the patients' data to make each subgroup size more balanced can significantly reduce the phenotype bias. The race bias in pneumonia phenotype has been reduced by 9 % on average with either of resampling or postprocessing method.

The highest gender bias can be reduced by 2.9 % with resampling on RNN. For the sepsis phenotyping task, the highest race bias decrease by 14 %. Nevertheless, the gender biases of our models are almost all below 2 % and can hardly be further reduced even with resampling or postprocessing. These two methods are the simplest yet effective debiasing strategies. Resampling method will make the patients of different subgroups more balanced which can significantly mitigate the prediction performance gaps. Postprocess is a more straightforward strategy which directly mitigate the bias by tuning the outputs.

- **There is a trade-off between phenotyping accuracy and bias.** From Fig. 2, we can find that most phenotyping models' phenotype accuracy will decrease when the debiasing method is applied. This phenomenon also appears in sepsis phenotype as shown in Fig. 3. So when we develop and deploy the phenotyping method, we need to make a trade-off between accuracy and bias based on the real-world phenotyping requirement. The trade-off between accuracy and bias also exists in other prediction tasks. How to maintain the accuracy as well as bias in phenotyping remains as a challenge in designing phenotyping algorithms.

5. Limitations

There are still limitations of this work, for which we conclude them into three points. The first is we only consider two representative diseases, which cannot cover all the disease types. The findings from this paper may not be scalable to some specific type of diseases. For example, some disease has unique patient distributions, e.g., breast cancer, prostate cancer, etc. In this paper, we haven't considered the bias and fairness issue in these specific diseases. Secondly, in the implementation of the data processing and methods, there may be bias in this process. One of them is the conversion from ICD to 9 to ICD-10. There is potential bias because the one-to-one conversion from ICD to 9 to ICD-10 is not 100 % straightforward. The other is the random sampling of negative patients. The random process to select patients may involve the bias. More granularity method to sample the negative patients can be employed in the future work.

6. Discussion and conclusion

From the experiment analysis on the main categories of phenotyping models and debiasing methods. We will discuss some limitations and future directions of this topic. We will also conclude this work with several takeaways and conclusions.

In this work, we choose two common diseases which are pneumonia and sepsis. However, there are some diseases that have specific characteristics. These specialties may make phenotyping bias on these diseases different from the findings we summarize in this work. For example, breast cancer is more commonly diagnosed among females compared to males [58]. The patients' data distribution across genders will be obviously different between females and males, which may cause significant gender bias in phenotyping. So for some specific diseases, we need to analyze their potential bias case by case.

We investigate the bias issue in phenotyping from a computational perspective. However, there is still a gap between the computational perspective and the clinical perspective. Addressing this gap represents one of the most promising and crucial directions for future research. In our future work, we will consider developing some methods that can clearly deliver computational fairness to the clinical practitioners and involve them to collaborate in the study. In this work, we mainly focus on the bias mitigation strategy in the data processing, model training, and output calibration steps. However, the data collection in healthcare is also very important. How to collect the data containing less bias remains a promising future direction.

To summarize, we comprehensively investigate the bias and the bias mitigation methods with pneumonia and sepsis phenotyping. From the perspective of phenotyping bias, we find that race bias is more obvious than gender bias and the rule-based phenotyping method demonstrates significantly less bias than machine learning phenotyping methods. Simply excluding the sensitive attributes doesn't work well in bias mitigation. Moreover, from the perspective of bias mitigation, we find that resample and post-process these two methods are simple yet effective in bias mitigation. Moreover, if the fairness of the phenotyping model improves through mitigation, the phenotyping accuracy will be negatively affected to some extent. So the tradeoff between fairness and accuracy needs to be considered when implementing and deploying the phenotyping model. The future work in this line of research can be derived in several directions. The first one is to develop more advanced debiasing methods for the phenotyping models according to the task specialties. The second is to bridge the gap of fairness between computation and clinical, which will help translate the computational debiasing methods into real-world clinical practice. The third direction is inspired by the findings from our experimental results that we can attach more importance to the healthcare data collection stage and improve the access of healthcare resources to the underrepresented groups.

Statement of significance

Problem or Issue: The bias issue in phenotyping in the era of EHR is not sufficiently researched especially from the computational perspective.

What is Already Known: There is bias in healthcare and especially the phenotyping task. And the bias in phenotyping stage will do no good to some underrepresented groups. Further, the bias in phenotyping will also affect the other related biomedical activities, like clinical trial matching, etc.

What this Paper Adds: This paper aims to provide comprehensive study of the bias issue in electronic phenotyping from a computational perspective. To the best of our knowledge, we are the first work to intensively investigate this fairness problem. We expect our work can inspire more efforts in this topic in the future.

CRediT authorship contribution statement

Sirui Ding: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Shenghan Zhang:** Writing – original draft, Methodology, Data curation. **Xia Hu:** Investigation. **Na Zou:** Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jbi.2024.104671.

References

[1] C.M. Delude, Deep phenotyping: the details of disease, Nature 527 (7576) (2015) S14–S15.

[2] S. Zhang, H. Li, R. Tang, S. Ding, L. Rasmy, D. Zhi, N. Zou, X. Hu, Pheme: a deep ensemble framework for improving phenotype prediction from multi-modal data, arXiv preprint arXiv:2303.10794 (2023).

[3] J.A. Williams, S.I. Hurst, J. Bauman, B.C. Jones, R. Hyland, J.P. Gibbs, R.S. Obach, S.E. Ball, Reaction phenotyping in drug discovery: moving forward with confidence? Curr. Drug Metab. 4 (6) (2003) 527–534.

[4] R.R. Edwards, R.H. Dworkin, D.C. Turk, M.S. Angst, R. Dionne, R. Freeman, P. Hansson, S. Haroutounian, L. Arendt-Nielsen, N. Attal, et al., Patient phenotyping in clinical trials of chronic pain treatments: immpact recommendations, Pain Reports 6 (1) (2021).

[5] L. Poissant, J. Pereira, R. Tamblyn, Y. Kawasumi, The impact of electronic health records on time efficiency of physicians and nurses: a systematic review, J. Am. Med. Inform. Assoc. 12 (5) (2005) 505–516.

[6] J.M. Banda, M. Seneviratne, T. Hernandez-Boussard, N.H. Shah, Advances in electronic phenotyping: from rule-based definitions to machine learning models, Annual Rev. Biomedical Data Sci. 1 (2018) 53–68.

[7] H. Alzoubi, R. Alzubi, N. Ramzan, D. West, T. Al-Hadhrami, M. Alazab, A review of automatic phenotyping approaches using electronic health records, Electronics 8 (11) (2019) 1235.

[8] I. Chien, N. Deliu, R. Turner, A. Weller, S. Villar, N. Kilbertus, Multidisciplinary fairness considerations in machine learning for clinical trials, Proce. 2022 ACM Conference on Fairness, Accountability, and Transparency (2022) 906–924.

[9] J.C. Kirby, P. Speltz, L.V. Rasmussen, M. Basford, O. Gottesman, P.L. Peissig, J. A. Pacheco, G. Tromp, J. Pathak, D.S. Carrell, et al., Phekb: a catalog and workflow for creating electronic phenotype algorithms for transportability, J. Am. Med. Inform. Assoc. 23 (6) (2016) 1046–1052.

[10] C. Shivade, P. Raghavan, E. Fosler-Lussier, P.J. Embi, N. Elhadad, S.B. Johnson, A. M. Lai, A review of approaches to identifying patient phenotype cohorts using electronic health records, J. Am. Med. Inform. Assoc. 21 (2) (2014) 221–230.

[11] A.N. Kho, M.G. Hayes, L. Rasmussen-Torvik, J.A. Pacheco, W.K. Thompson, L. L. Armstrong, J.C. Denny, P.L. Peissig, A.W. Miller, W.-Q. Wei, et al., Use of diverse electronic medical record systems to identify genetic risk for type 2 diabetes within a genome-wide association study, J. Am. Med. Inform. Assoc. 19 (2) (2012) 212–218.

[12] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, J. Machine Learning Res. 9 (2008) 1871–1874.

[13] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[14] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[15] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[16] S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, Science and Information Conference, IEEE 2014 (2014) 372–378.

[17] R.J. Carroll, A.E. Eyler, J.C. Denny, Naïve electronic health record phenotype identification for rheumatoid arthritis, in: AMIA Annual Symposium Proceedings, Vol. 2011, American Medical Informatics Association, 2011, p. 189.

[18] S. Yang, P. Varghese, E. Stephenson, K. Tu, J. Gronsbell, Machine learning approaches for electronic health records phenotyping: a methodical review, J. Am. Med. Inform. Assoc. 30 (2) (2023) 367–381.

[19] Q. Li, K. Zhao, C.D. Bustamante, X. Ma, W.H. Wong, Xrare: a machine learning method jointly modeling phenotypes and genetic evidence for rare disease diagnosis, Genet. Med. 21 (9) (2019) 2126–2134.

[20] T. Norman, N. Weinberger, K. Y. Levy. (2023) Robust linear regression for general feature distribution, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 2405–2435.

[21] Y. Park, J.C. Ho, Tackling overfitting in boosting for noisy healthcare data, IEEE Trans. Knowl. Data Eng. 33 (7) (2019) 2995–3006.

[22] R.G. Mantovani, T. Horváth, R. Cerri, S.B. Junior, J. Vanschoren, A.C.P.d.L.F. de Carvalho, An empirical study on hyperparameter tuning of decision trees, arXiv preprint arXiv:1812.02207 (2018).

[23] M. Ross, W. Wei, L. Ohno-Machado, "big data" and the electronic health record, Yearb. Med. Inform. 23 (01) (2014) 97–104.
[24] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinform. 19 (6) (2018) 1236–1246.
[25] E. Choi, A. Schuetz, W.F. Stewart, J. Sun, Using recurrent neural network models for early detection of heart failure onset, J. Am. Med. Inform. Assoc. 24 (2) (2017) 361–370.
[26] S. Gao, M. Alawad, M.T. Young, J. Gounley, N. Schaefferkoetter, H.J. Yoon, X.-C. Wu, E.B. Durbin, J. Doherty, A. Stroup, et al., Limitations of transformers on clinical text classification, IEEE J. Biomed. Health Inform. 25 (9) (2021) 3596–3607.
[27] R. Shwartz-Ziv, N. Tishby, Opening the black box of deep neural networks via information, arXiv preprint arXiv:1703.00810 (2017).
[28] E. Zihni, V.I. Madai, M. Livne, I. Galinovic, A.A. Khalil, J.B. Fiebach, D. Frey, Opening the black box of artificial intelligence for clinical decision support: a study predicting stroke outcome, PLoS One 15 (4) (2020) e0231166.
[29] G. Yang, Q. Ye, J. Xia, Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: a mini-review, two showcases and beyond, Information Fusion 77 (2022) 29–52.
[30] J. Lee, C. Liu, J. Kim, Z. Chen, Y. Sun, J.R. Rogers, W.K. Chung, C. Weng, Deep learning for rare disease: a scoping review, J. Biomed. Inform. (2022) 104227.
[31] J.C. Ho, J. Ghosh, S.R. Steinhubl, W.F. Stewart, J.C. Denny, B.A. Malin, J. Sun, Limestone: High-throughput candidate phenotype generation via tensor factorization, J. Biomed. Inform. 52 (2014) 199–211.
[32] A. Afshar, I. Perros, H. Park, C. Defilippi, X. Yan, W. Stewart, J. Ho, J. Sun, Taste: temporal and static tensor factorization for phenotyping electronic health records, Proce. ACM Conference on Health, Inference, and Learning (2020) 193–203.
[33] L.L. Coventry, J. Finn, A.P. Bremner, Sex differences in symptom presentation in acute myocardial infarction: a systematic review and meta-analysis, Heart Lung 40 (6) (2011) 477–491.
[34] J.L. Mehta, Z. Bursac, P. Mehta, D. Bansal, L. Fink, J. Marsh, R. Sukhija, R. Sachdeva, Racial disparities in prescriptions for cardioprotective drugs and cardiac outcomes in veterans affairs hospitals, Am. J. Cardiol. 105 (7) (2010) 1019–1023.
[35] T. Y. Sun, S. A. Bhave, J. Altosaar, N. Elhadad. (2022) Assessing phenotype definitions for algorithmic fairness, in: AMIA Annual Symposium Proceedings, Vol. 2022, American Medical Informatics Association. p. 1032.
[36] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Computing Surveys (CSUR) 54 (6) (2021) 1–35.
[37] S. Ding, R. Tang, D. Zha, N. Zou, K. Zhang, X. Jiang, X. Hu. (2022) Fairly predicting graft failure in liver transplant for organ assigning, in: AMIA Annual Symposium Proceedings, Vol. 2022, American Medical Informatics Association. p. 415.
[38] C. Li, S. Ding, N. Zou, X. Hu, X. Jiang, K. Zhang, Multi-task learning with dynamic re-weighting to achieve fairness in healthcare predictive modeling, J. Biomed. Inform. 143 (2023) 104399.
[39] M. Du, F. Yang, N. Zou, X. Hu, Fairness in deep learning: a computational perspective, IEEE Intell. Syst. 36 (4) (2020) 25–34.
[40] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowl. Inf. Syst. 33 (1) (2012) 1–33.
[41] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, K.R. Varshney, Optimized pre-processing for discrimination prevention, Adv. Neural Inf. Proces. Syst. 30 (2017).
[42] T. Wang, J. Zhao, M. Yatskar, K.-W. Chang, V. Ordonez, Balanced datasets are not enough: estimating and mitigating gender bias in deep image representations, Proce. IEEE/CVF Int. Conference on Comp. Vision (2019) 5310–5319.
[43] Y. Elazar, Y. Goldberg, Adversarial removal of demographic attributes from text data, arXiv preprint arXiv:1808.06640 (2018).
[44] A.S. Ross, M.C. Hughes, F. Doshi-Velez, Right for the right reasons: training differentiable models by constraining their explanations, arXiv preprint arXiv: 1703.03717 (2017).
[45] F. Liu, B. Avci, Incorporating priors with feature attribution on text classification, arXiv preprint arXiv:1906.08286 (2019).
[46] D. Madras, E. Creager, T. Pitassi, R. Zemel. (2018) Learning adversarially fair and transferable representations, in: International Conference on Machine Learning, PMLR. pp. 3384–3393.
[47] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Men also like shopping: reducing gender bias amplification using corpus-level constraints, arXiv preprint arXiv:1707.09457 (2017).
[48] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Adv. Neural Inf. Proces. Syst. 29 (2016).
[49] A.E. Johnson, T.J. Pollard, L. Shen, L.-W.-H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, Mimic-iii, a freely accessible critical care database, Sci. Data 3 (1) (2016) 1–9.
[50] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff,P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000 (June 13)) e215–e220, circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID: 1085218; doi: 10.1161/01.CIR.101.23.e215.
[51] C.W. Seymour, J.N. Kennedy, S. Wang, C.-C.-H. Chang, C.F. Elliott, Z. Xu, S. Berry, G. Clermont, G. Cooper, H. Gomez, et al., Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis, JAMA 321 (20) (2019) 2003–2017.
[52] L. Gattinoni, D. Chiumello, P. Caironi, M. Busana, F. Romitti, L. Brazzi, L. Camporota. (2020) Covid-19 pneumonia: different respiratory treatments for different phenotypes?.
[53] M.-C. Popescu, V.E. Balas, L. Perescu-Popescu, N. Mastorakis, Multilayer perceptron and neural networks, WSEAS Trans. Circuits and Systems 8 (7) (2009) 579–588.
[54] H. Harutyunyan, H. Khachatrian, D.C. Kale, G. Ver Steeg, A. Galstyan, Multitask learning and benchmarking with clinical time series data, Sci. Data 6 (1) (2019) 96.
[55] R.A. Harshman, et al., Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis, UCLA working papers in phonetics (1970).
[56] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, H. Wallach. (2018) A reductions approach to fair classification, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Vol. 80 of Proceedings of Machine Learning Research, PMLR. pp. 60–69.
[57] B.H. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, Proce. 2018 AAAI/ACM Conference on AI, Ethics, and Society (2018) 335–340.
[58] E. Midding, S.M. Halbach, C. Kowalski, R. Weber, R. Wurstlein, N. Ernstmann, Men with a "woman's disease": stigmatization of male breast cancer patients—a mixed methods analysis, American J. Men's Health 12 (6) (2018) 2194–2207.
[59] Han, Xiaotian, Zhimeng Jiang, Ninghao Liu, Na Zou, Qifan Wang, Xia Hu. (2022) "Do We Really Achieve Fairness with Explicit Sensitive Attributes?.".