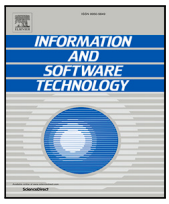
Information and Software Technology 189 (2026) 107898

Contents lists available at ScienceDirect

Information and Software Technology

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



FAIR-CARE: A comparative evaluation of unfairness mitigation approaches☆

Chiara Criscuolo [a,*], Mattia Salnitri [b], Davide Martinenghi [a]

[a] Politecnico di Milano DEIB, Piazza Leonardo da Vinci, 32, Milan, 20133, Italy
[b] University of Bergamo, Via Salvecchio, 19, Bergamo, 24129, Italy

ARTICLE INFO

Keywords:
Data bias
Fairness
Machine learning
Binary classification
Mitigation

ABSTRACT

Bias and unfairness in Machine Learning (ML) are challenging to detect and mitigate, particularly in critical fields such as finance, hiring, and healthcare. While numerous unfairness mitigation techniques exist, most evaluation frameworks assess only a limited set of fairness metrics, primarily focusing on the trade-off between fairness and accuracy. We introduce FAIR-CARE, a new open-source and robust approach that consists of an evaluation pipeline designed for the systematic assessment of unfairness mitigation techniques. Our approach simultaneously evaluates multiple fairness and performance metrics across various ML models. We conduct a comparative analysis on healthcare datasets with diverse distributions—including target class, protected attribute, and their joint distributions—to identify the most effective mitigation technique for each processing type (pre-, in-, and post-processing). Furthermore, we determine the best-performing techniques across different datasets, fairness metrics, performance metrics, and ML models. Finally, we provide practical insights into the application of these techniques, offering actionable guidance for both researchers and practitioners.

1. Introduction

It is now generally understood that applications based on Artificial Intelligence (AI) pose a number of societal and ethical concerns. In response, governments and international organizations are regulating the creation process of AI-based applications, from design to use. Among the first regulations to address the risks of AI and Machine Learning (ML) systems, the *AI Act* [1] introduces a classification of software products according to their level of societal risk. High-risk applications include systems with a potential adverse impact on people's lives, such as ML models for automated law enforcement or healthcare prediction. The strict requirements imposed by these regulations are designed to evaluate and guarantee the transparency and fairness of the adopted models.

Fairness in ML has been widely studied in the past few years, leading to the definition of many unfairness mitigation techniques. Yet, their effectiveness, measured with specific fairness metrics, varies based on the characteristics of the datasets and the applied ML model. New mitigation techniques appear every year, and the challenging task of selecting the most appropriate and effective ones requires a proper comparative analysis. This is further complicated by the large number of possible combinations of mitigation techniques, fairness and performance metrics, ML algorithms, and the variety of application domains (e.g., healthcare, hiring, justice, etc.). While prior comparative studies [2–4] have examined mitigation strategies across domains, our study focuses exclusively on healthcare, to provide context-specific insights. This domain-specific approach allows for a more cohesive assessment using a broader set of fairness metrics and datasets than typical surveys, enabling a deeper evaluation within this critical high-risk domain. In fact, recent studies [5,6] highlight that the effectiveness of fairness mitigations can vary significantly across contexts, reinforcing the value of our focused analysis. In particular, healthcare datasets usually contain sensitive information such as race, sex, and age, suitable for fairness evaluation, which indeed requires the presence of protected attributes. In other domains, the collection of such attributes is usually not allowed for privacy reasons. Moreover, healthcare is a high-risk sector according to the AI Act. It is therefore particularly important to guarantee fairness in ML models used in this context to prevent wrong decisions.

ML-driven decision-making in healthcare has demonstrated great potential to make informed healthcare decisions tailored to individual patients, obtaining new and highly reliable therapies [7], such as diabetic retinopathy detection [8,9]. However, its effectiveness is grounded on the reliability of the results of ML algorithms, which are based on the quality of the training datasets. Biases in training datasets can result in unfair predictions, particularly affecting underrepresented groups. Decision-making systems for healthcare that are based on ML algorithms have been shown to cause wrong or ineffective therapies. This phenomenon occurs, for instance, in ML systems

☆ This article is part of a Special issue entitled: 'Fairness in software systems' published in Information and Software Technology.

* Corresponding author.
E-mail address: chiara.criscuolo@polimi.it (C. Criscuolo).

https://doi.org/10.1016/j.infsof.2025.107898
Received 30 April 2025; Received in revised form 21 August 2025; Accepted 17 September 2025
Available online 25 September 2025

that privilege certain sub-populations at the expense of other ones. A recent example [10] has shown that ML models trained on chest X-rays exhibit underdiagnosis for intersectional subgroups, such as Black female patients. This issue is exacerbated when minorities are subject to discrimination, resulting in lower-precision analyses and diminished healthcare quality due to their under-representation in the training data.

These examples highlight the growing need for unfairness mitigation techniques to become a routine task in the data science community, as also required by legislation. Researchers should not have to spend excessive time determining the most effective mitigation technique for a given problem. Instead, given a dataset, an ML model, and a few fairness and performance metrics, they should be able to confidently identify the most suitable mitigation technique.

To address these challenges, we introduce our approach, called FAIR-CARE (FAIRness in maChine leArning algoRithms Evaluation), to systematically assess the effectiveness of unfairness mitigation techniques, helping users and researchers in identifying the best one for their specific needs.

Our main contribution is the development of an evaluation pipeline that, in the specific domain of healthcare, evaluates 12 mitigation techniques, across 6 ML models, using 11 fairness metrics and 4 performance metrics, over 7 datasets from the healthcare sector, thereby covering a number of configurations well beyond the extent of previous studies. The pipeline is fully automated, extensible, available open source [11], freely accessible and easily usable through the Colab environment [12].

Another contribution is the identification of the most effective mitigation technique, for each processing type (pre-, in-, and post-processing), across the following four different aspects:

1. *dataset category*, i.e., skewed or balanced according to 3 different distributions (target class, protected attribute, and their joint distributions);
2. *statistical group fairness metric*, thereby evaluating overall fairness and possible improvements wrt the baseline (using 2 different comparison criteria), focusing on one protected attribute at a time;
3. *performance metric*, assessing the variations wrt the baseline and the trade-offs with fairness;
4. *ML model*, examining the impact of the model choice on the results.

By providing ranked lists of techniques for each dataset configuration and for each of the mentioned aspects, we identify the techniques that consistently achieve the best fairness and performance values across the ML models.

As a last contribution, we summarize our findings providing practical insights into the application of these techniques and introducing a guideline composed by 5 points, with a comprehensive discussion of the general limitations and key considerations encountered during our experiments. We acknowledge that focusing exclusively on healthcare datasets limits the generalizability of our findings across domains. However, this domain-specific scope is intentional, as it allows us to provide more actionable and context-aware guidance for practitioners in healthcare settings. Importantly, the FAIR-CARE framework and accompanying codebase are designed to be adaptable and remain applicable to datasets from other domains with minimal adjustments. Other domains beyond healthcare, such as finance and hiring, can benefit from the proposed pipeline as well, provided that the datasets respect the input requirements and that the parameters of mitigation techniques are tuned accordingly.

The rest of the paper is organized as follows: Section 2 presents the related work, Section 3 outlines the foundational concepts of our approach, Section 4 presents the adopted fairness metrics, and Section 5 classifies the unfairness mitigation techniques used in our evaluation. Section 6 analyzes the characteristics of the selected datasets, and Section 7 describes the execution pipeline and the results of our experiments. Section 8 summarizes our findings, while Section 9 concludes the paper.

2. Related work

Fairness in Machine Learning (ML) has emerged as a critical research focus, driven by the rapid expansion of ML applications across numerous sectors, including healthcare. Fairness is broadly understood as *the absence of any prejudice or favoritism toward an individual or group based on their inherent or acquired characteristics* [13], and has been extensively studied under this perspective in the field of ML [14,15]. When applied to the healthcare sector, fairness can be more specifically framed as *the equitable treatment of individuals or groups by ML models, irrespective of protected characteristics*. Ensuring fairness in healthcare is particularly critical, since biased outcomes can exacerbate health disparities, undermine trust, and lead to unequal access to life-saving treatments [16].

Several theoretical surveys have addressed fairness in ML [13–15,17]. However, these works primarily focus on conceptual and theoretical aspects without performing extensive empirical studies or experiments. This limitation also extends to fairness research within the healthcare sector, where studies such as [16,18–20] analyze fairness issues, but fail to deliver holistic evaluations that encompass both ethical and technical dimensions.

To address this gap, we surveyed existing experimental works that apply unfairness mitigation techniques to ML models. Table 1 summarizes the key aspects of these studies, and reports on the number of ML models, fairness metrics, mitigation techniques, datasets (along with their protected attributes), and performance metrics used. Specifically, the third row refers to those mitigation techniques that address group fairness and have a public library code implementation.[1]

The last row is the product of all other rows, with the additional observation that the number of fairness metrics investigated within FAIR-CARE is doubled, since we used 2 criteria to compare metrics. With 69,696 experimental configurations, FAIR-CARE explores a significantly broader evaluation space than prior studies, providing a robust statistical foundation for analyzing the interplay between fairness mitigations, model performance, and dataset characteristics.

Our work aims to address several limitations arising in previous efforts:

Scope of ML Models Evaluated: In the healthcare domain, practitioners often compare multiple models to identify the best-performing one. Moreover, they evaluate both interpretable models (e.g., logistic regression, decision trees) and more sophisticated ensemble methods (e.g., random forest, bagging, and boosting) to find the best predictive performance while maintaining some interpretability [27]. Existing studies, such as [3,25,26], frequently restrict their analysis to basic models. In contrast, our approach incorporates a wider range of ML models, reflecting real-world clinical model selection and enabling a more practical evaluation of fairness across diverse ML models.

Restricted Fairness Metrics Coverage: Existing works, such as [2,3,22,23], often use only a few (typically 2–3) fairness metrics, which limits the scope of their analysis. Since fairness metrics can sometimes conflict and are context-dependent, relying on a small subset can overlook crucial insights. Importantly, no single metric or subgroup of metrics can be universally deemed “best” or most representative. Our study overcomes this limitation by employing a broader range of fairness metrics (11 in total, with two different comparison approaches), allowing for a more thorough evaluation.

Contextual Limitations in Dataset Selection: A significant shortcoming of the current literature, as in [2,4,21,24], is the use of generic datasets (e.g., COMPAS [28], Adult [29], German [30]), which are often treated as benchmark despite their limitations (e.g. outdated

[1] The works [24–26] additionally consider 2, 9, and, respectively, 7 techniques that do not meet these requirements and, thus, are outside the scope of the present study.

Table 1
Comparison of experimental surveys. The focus is on mitigation techniques addressing group fairness and with a public library code implementation.

	[21]	[22]	[23] Fairea	[24]	[25] FairCR	[2] MAAT	[3]	[4]	[26]	FAIR-CARE
# ML models	4	7	3	5	1	5	1	12	3	**6**
# Fairness metrics	8	2	2	5	7	3	3	7	5	**11 · 2**
# Mitigation techniques	4	8	8	11	12	5	6	7	10	**12**
# Datasets × Prot. Attrs.	9	8	5	3	6	8	8	7	8	**11**
# Performance metrics	4	3	1	4	1	4	2	2	5	**4**
# Exp. configurations	4608	2688	240	3300	504	2400	288	8232	6000	**69,696**

data, presence of errors, etc.) [5]. Moreover, prior studies, to the best of our knowledge, rarely evaluate datasets within a specific context. In our work, we focus exclusively on healthcare-related datasets, ensuring better contextualization and more actionable insights. By adopting datasets previously unused for fairness evaluation, we align our experiments with real-world healthcare applications, providing clearer guidelines for fairness assessments in this domain.

Implementation Inconsistencies: Comparisons across existing surveys are often hindered by inconsistent or ad-hoc implementations of unfairness mitigation techniques. Since publicly available implementations are scarce, researchers frequently develop their own versions of these techniques, introducing variations that complicate comparisons (such as [24–26]). To ensure reproducibility and transparency, our approach relies exclusively on official libraries and publicly available implementations of mitigation techniques, avoiding any modification of the source code.

To conclude, several recent studies have empirically explored the fairness–performance trade-off across diverse settings and classifiers, including [4,26], which offer a broad benchmark of mitigation techniques. Our work builds upon this foundation by offering a more domain-specific and structured evaluation within healthcare settings. Through systematic experimentation, we observe that while fairness mitigation often entails some performance degradation, this trade-off is neither universal nor linear. Some mitigations, show favorable fairness improvements with minimal loss in predictive accuracy. These findings align with broader trends in the literature but offer more actionable insights for practitioners operating in high-risk domains, where balancing fairness and reliability is critical.

3. Design of experiments

This section outlines the foundational concepts characterizing the experiments run in this paper, focusing on how to measure fairness and improve it in practical applications.

3.1. Structure of an experiment

The experiments to be discussed in this paper aim to measure the impact of various techniques for mitigating unfairness on a dataset when adopting a specific ML model. The *target users* are data scientists and analysts who need to evaluate the effectiveness of mitigation techniques.

Input. Every configuration of an experiment consists of the following input elements:

- a ML *model*;
- a *performance metric* used to evaluate the ML model;
- a *fairness metric*;
- an *unfairness mitigation technique*;
- a *dataset*;
- a *criterion* for comparing fairness.

For the fairness metrics to be applicable, the dataset must include at least one binary *protected attribute* g and a binary *(target) class* y (whose *label* ℓ can be positive (1) or negative (0)). Although we only cover binary protected attributes, it is always possible to transform non-binary attributes into binary ones. Variable $\hat{y}$ identifies the corresponding *prediction* computed by the ML model, typically associated with a probability that is then compared to a classification threshold.

Table 2
ML models taken from [31].

Acr.	Algorithm	Parameters
LORE	*Logistic Regression*	max_iter = 500
DETR	*Decision Tree*	–
EXRT	*Extremely Rand. Trees*	n_est = 30
RAFO	*Random Forest*	n_est = 30
BAGG	*Bagging*	Dec. Tree, depth = 3, n_est = 30
ADBO	*Ada Boost Boosting*	Dec. Tree, depth = 3, n_est = 30

The protected attribute is a characteristic such as *sex*, *ethnicity*, or *age* that should not be the determinant of discrimination with respect to the target class to be predicted. Consider, for instance, a dataset of hospitalized people, where a ML model predicts diabetes in patients, i.e., the target class y is the presence of diabetes. If the protected attribute g is the sex attribute, patients with either value of g (e.g., female) should not be discriminated when predicting the target class (e.g., with a lower precision of the diabetes forecast).

To improve confidence in the measures resulting from our experiments and reduce the dependence on the intrinsic characteristics of a specific model or technique, each element of the configuration of an experiment will be instantiated in multiple ways. In particular, by using various datasets, we will be able to compare the outcomes and generalize the results obtained with the different mitigation techniques and ML models, which, in evidence-based medicine, are used for heterogeneous purposes. Similarly, using multiple ML models will allow for a comprehensive understanding of how the different algorithms may contribute to unfairness. Our focus is on classification techniques, since fairness is evaluated on classification.

Overall, we will use 7 different datasets, 6 ML models, 11 fairness metrics, 12 unfairness mitigation techniques, 4 performance metrics, and 2 criteria for comparing fairness. We start by describing the ML models in Section 3.2 and the performance metrics in Section 3.3. Then, fairness metrics used are described, through a catalog, in Section 4, along with an illustration of the criteria we adopt for comparing fairness before and after mitigation; mitigation techniques are presented in Section 5; finally, the datasets used in our experiments are presented in Section 6.

Output. The output of an experiment consists of the values of the measured (fairness and performance) metrics. In particular, we will measure fairness according to a fairness metric and compare it (through either division or subtraction) before and after applying a mitigation technique. Mitigation is considered *effective* on a dataset when it improves the fairness of the results of a ML model. Additionally, depending on the type of mitigation technique (pre-processing, in-processing, or post-processing, as will be discussed in Section 5), the execution of the experiment may also include, as a byproduct, a mitigated dataset (pre-processing), a mitigated ML model (in-processing), or a mitigated prediction (post-processing).

Table 3
Fairness metrics catalog. Here, $\hat{y}$ is the prediction, y is the actual value of the class in the dataset, and g is the protected attribute, which can only take one of two values: p (privileged group) or d (discriminated group).

Name	Definition	Constraint
Group Fairness, a.k.a. Statistical Parity (GFA)	*Both groups have an equal probability of a positive prediction* [33]	$P(\hat{y}=1\|g=d)=P(\hat{y}=1\|g=p)$
Predictive Parity (PPA)	*Given a positive prediction, both groups have the same probability of having a negative class* [34]	$P(y=0\|\hat{y}=1,g=d)=P(y=0\|\hat{y}=1,g=p)$
Predictive Equality (PEQ)	*Given a negative class, both groups have the same probability of having a positive prediction* [34]	$P(\hat{y}=1\|y=0,g=d)=P(\hat{y}=1\|y=0,g=p)$
Equal Opportunity (EOP)	*Given a positive class, both groups have the same probability of having a negative prediction* [34]	$P(\hat{y}=0\|y=1,g=d)=P(\hat{y}=0\|y=1,g=p)$
Equalized Odds (EOD)	*Both metrics PEQ and EOP are satisfied* [35]	$P(\hat{y}=0\|y=1,g=d)=P(\hat{y}=0\|y=1,g=p)\wedge P(\hat{y}=1\|y=0,g=d)=P(\hat{y}=1\|y=0,g=p)$
Conditional Use Accuracy Equality (CUA)	*Given a positive (negative) prediction, both groups have an equal probability of a positive (negative) class* [36]	$P(y=1\|\hat{y}=1,g=d)=P(y=1\|\hat{y}=1,g=p)\wedge P(y=0\|\hat{y}=0,g=d)=P(y=0\|\hat{y}=0,g=p)$
Overall Accuracy Equality (OAE)	*Both groups have the same probability of true positives and true negatives* [36]	$P(\hat{y}=y\|g=d)=P(\hat{y}=y\|g=p)$
Treatment Equality (TEQ)	*Both groups have an equal ratio of false negatives and false positives* [34]	$\frac{P(\hat{y}=0\wedge y=1\|g=p)}{P(\hat{y}=1\wedge y=0\|g=p)}=\frac{P(\hat{y}=0\wedge y=1\|g=d)}{P(\hat{y}=1\wedge y=0\|g=d)}$
FOR Parity (FOR)	*Given a negative prediction, both groups have an equal probability of a positive class* [37]	$P(y=1\|\hat{y}=0,g=d)=P(y=1\|\hat{y}=0,g=p)$
FN/GS Parity (FNP)	*Both groups have the same probability of a positive class and a negative prediction* [37]	$P(\hat{y}=0,y=1\|g=d)=P(\hat{y}=0,y=1\|g=p)$
FP/GS Parity (FPP)	*Both groups have the same probability of a negative class and a positive prediction* [37]	$P(\hat{y}=1,y=0\|g=d)=P(\hat{y}=1,y=0\|g=p)$

3.2. ML models

Each ML model we consider addresses a binary classification problem, in which the goal is to predict the binary label ℓ (e.g., 0 or 1) of the target class y. Thus, given the dataset, the protected attribute g and the target class, the ML model is trained on a training set and evaluated over a test set. Table 2 shows the ML models adopted in this research, along with their parameters. The selected models reflect a mix of interpretable algorithms (e.g., LORE , DETR) and more complex ensemble methods (e.g., EXRT, ADBO), which are commonly used in healthcare predictive modeling due to their high performance and varying levels of interpretability [27]. This selection aligns with clinical practice, where model choice often balances accuracy with interpretability to support trust in automated decisions. While our framework supports the inclusion of additional models, we focused on this subset to provide a meaningful yet manageable comparison across fairness mitigations.

3.3. Performance metrics

To evaluate our ML models from a performance perspective, we adopt the most common metrics: *accuracy* (Acc), *precision* (Pre), *recall*, (Rec) and *F1 score* (F_1). They are based on the values of the confusion matrix generated by the prediction algorithm [32], where TP (*true positives*) and TN (*true negatives*) are the cardinalities of the sets of correct predictions ($y=\hat{y}$) in the dataset with a positive or negative target class, respectively. In contrast, FP (*false positives*) and FN (*false negatives*) are the cardinalities for which the prediction is not equal to the target class ($y\neq\hat{y}$), with a negative or positive label, respectively.

$$\texttt{Acc}=\frac{TP+TN}{N};\texttt{Pre}=\frac{TP}{TP+FP};\texttt{Rec}=\frac{TP}{TP+FN};F_1=\frac{2\cdot\texttt{Pre}\cdot\texttt{Rec}}{\texttt{Pre}+\texttt{Rec}}$$

4. Fairness metrics

We concentrate on statistical group fairness metrics, rather than causal or counterfactual ones, because metrics and mitigations developed in the statistical field are prevalent in the state of the art and more easily interpreted and understood. Furthermore, the causal approach requires causal graphs, which, in turn, require domain knowledge and manual input, which cannot be properly built fully automatically. We did not focus on intersectional fairness, e.g., multiple attribute at the same time, since the majority of mitigation techniques available handle only one protected attribute at a time. Given the complexity of fairness in healthcare, we selected a broad set of well-established statistical group metrics to capture different types of disparities across protected attributes like sex and age. Rather than aiming to identify a single best metric, our goal is to offer practitioners a comprehensive view of fairness trade-offs, acknowledging that no one-size-fits-all metric exists in this domain. Table 3 shows the representative set of statistical fairness metrics, with their respective definitions as proposed in [38].[2] As customary, these definitions refer to two possible values for the protected attribute g: a privileged group p and a discriminated one d. They also exploit the values of the confusion matrix TP, TN, FP, FN; if these values are computed for the discriminated group we use a d subscript: TP_d, TN_d, FP_d, FN_d; similarly for the privileged group p: TP_p, TN_p, FP_p, FN_p. Table 3 reports constraints in the form of equalities between probabilities, which can equivalently be expressed through explicit reference to true/false positives/negatives. In our work, we do not redefine fairness for healthcare but instead apply general fairness definitions in a healthcare context.

We say that a metric (or, more precisely, its associated constraint) is *satisfied*, i.e., one is fair with respect to that metric if, through comparison of the two probabilities, they do not deviate by more than a given amount τ. We adopt two different *comparison approaches*: division and subtraction. For both, we aim to obtain values in the $[-1,+1]$ interval, where 0 represents fairness, while values near -1 (resp., $+1$) indicate discrimination towards the discriminated (resp., privileged) group. The

[2] We omitted *Conditional Statistical Parity*, which extends Group Fairness by allowing the inclusion of a legitimate attribute for assessment. Indeed, identifying such an attribute requires extensive knowledge of the healthcare domain, which led us to exclude this metric from our evaluation catalog.

value is then compared with an application-specific threshold τ, which, in our experiments, we set to $\tau = 0.15$, consistently with similar values adopted in the literature [39,40] (i.e., a 15% deviation is sufficient to determine unfairness between discriminated and privileged groups).

The division approach requires computing the ratio between the left- and the right-hand side of the constraint of a metric as defined in Table 3, i.e., between the privileged and discriminated expression. For example, with GFA, this ratio is $\frac{P(\hat{y}=1|g=d)}{P(\hat{y}=1|g=p)}$. Equivalently, the numerator can be computed as $\frac{TP_d+FP_d}{|d|}$, while the denominator as $\frac{TP_p+FP_p}{|p|}$, obtaining the final value $\upsilon = \frac{TP_d+FP_d}{TP_p+FP_p} \cdot \frac{|p|}{|d|}$. This value is then normalized in the $[-1, 1]$ interval as $\tilde{\upsilon} = \frac{\upsilon-1}{\upsilon+1}$, so that, when $\upsilon = 1$ (corresponding to non-discrimination), $\tilde{\upsilon} = 0$. Finally, $\tilde{\upsilon}$ is compared to the threshold value τ: the outcome is fair if and only if $|\tilde{\upsilon}| < \tau$. When the constraint consists of two equalities, as in the case of TEQ and CUA, we consider each equality separately, thus obtaining two normalized values $\tilde{\upsilon}_1$ and $\tilde{\upsilon}_2$: the final value we consider is the largest one in absolute value, i.e., $\tilde{\upsilon} = \text{argmax}_{x \in \tilde{\upsilon}_1, \tilde{\upsilon}_2} |x|$.

Comparison by subtraction works as before, but with a subtraction between the left- and right-hand side of the constraint. In this case, GFA amounts to computing $\upsilon = \frac{TP_d+FP_d}{|d|} - \frac{TP_p+FP_p}{|p|}$, which requires no normalization and can be directly compared with τ.

For improved robustness, the value of each metric is the result of applying k-fold cross-validation over the dataset computing the mean and standard deviation (std dev) over the k samples. Cross-validation is a more solid approach wrt train-test split in evaluating the model's performance on unseen data for each fold. For smaller datasets ($\leq$1500 tuples), using $k = 5$ ensures that each fold still contains a sufficient number of samples, reducing the risk of instability in the evaluation due to insufficient test data. For larger datasets (>1500 tuples), $k = 10$ leads to more stable estimates of metrics because each fold has enough data, and more folds provide a better representation of the dataset's variability. Using the mean to summarize the performance across folds can inadvertently average out positive and negative deviations from 0, potentially leading to a final value that appears closer to the ideal fairness metric value 0 than any individual fold. To address the risk associated with using the mean alone, we also report the std dev of the fairness metrics.

5. Unfairness mitigation techniques

Unfairness mitigation techniques can be classified into three different types, according to their intervention procedure:

- pre-processing, acting on the training data;
- in-processing, acting on the model;
- post-processing, acting on the model's predictions.

Table 4 shows the most relevant state-of-the-art techniques across each type of mitigation. These techniques are acknowledged as the most representative and commonly adopted in the relevant literature — see, e.g., [13,15]. We evaluate all three kinds of mitigation techniques to provide a holistic evaluation and comparison of how unfairness can be mitigated at various points in the ML pipeline, since focusing on just one subset would provide an incomplete picture and potentially overlook important opportunities for unfairness reduction. All of them aim to find a good trade-off between fairness and performance metrics (usually accuracy) when applied to binary classification models. Additionally, they can all be used off-the-shelf, with parametric customization only, and are therefore ideal for data science applications in a production environment [41].

The unfairness mitigation techniques used in our experiments are implemented by two tools: *AIF360* [39], the acronym for the *AI Fairness 360* tool by IBM, and *FairLearn* [46], by Microsoft. Correlation Remover (CR) and Threshold Optimizer (TO) are implemented by *FairLearn*, while all the other techniques are originally implemented by *AIF360*.

Table 4
Catalog of unfairness mitigation techniques.

Processing	Acr.	Technique
Pre-	RW	*Reweighing* [42]
	DI	*Disparate Impact Remover* [43]
	LF	*Learning Fair Representation* [44]
	OP	*Optimized Pre-processing* [45]
	CR	*Correlation Remover* [46]
In-	AD	*Adversarial Debiasing* [47]
	PR	*Prejudice Remover* [48]
	ER	*Exponential Gradient Reduction* [49]
Post-	EO	*Equalized Odds Post-Processing* [35]
	CE	*Calibrated Equalized Post-Processing* [50]
	RO	*Reject Option Classification* [51]
	TO	*Threshold Optimizer* [35]

5.1. Pre-processing mitigation techniques

In this type of processing, all techniques target the dataset, producing a repair typically by modifying the original dataset. The ML model will then be trained on the repaired dataset.

Reweighing (RW). RW [42] generates a specific set of weights for the tuples in the training set in each (group, label) combination to ensure fairness before classification. The goal of RW is to make the model treat groups fairly by adjusting the weights of the training instances, rather than changing their labels or features. The weights are calculated to ensure that, after applying RW, there is no disparity across the groups in terms of their likelihood of receiving positive or negative predictions. This technique counteracts any initial imbalance in the data by making the algorithm pay more or less attention to certain groups during training. A key feature of RW is that it preserves the overall probability of the positive target class in the dataset. In other words, the proportion of positive prediction remains the same across the dataset, but the distribution is adjusted to ensure fairness.

Disparate Impact Remover (DI). DI [43] edits feature values to increase group fairness while preserving rank ordering within groups. DI builds a repaired dataset where any attribute in the original dataset that could be used to predict the protected attribute is modified to guarantee fairness, while the protected attribute remains unchanged. The repair process is such that the ability to predict the target class is preserved. To address the trade-off between accuracy and fairness, the algorithm allows fixing a repair level between 0 and 1, where 0 yields the unmodified data, while 1 is a fully repaired dataset.

Learning Fair Representation (LF). LF [44] transforms the dataset into a new one that closely preserves the original data while effectively obfuscating information related to the protected attributes. LF aims to achieve group fairness by mapping each individual into a probability distribution in a new representation space (repaired dataset), while fulfilling the following three criteria: (i) the transformation from the original data to the new representation satisfies a specific fairness metric, (ii) the new representation space preserves the essential information in the original data, excluding details related to protected attributes, (iii) the produced predictive function, obtained by first mapping probabilistically to the new representation space and then to the target, remains similar to the original one.

Optimized pre-processing (OP). OP learns a probabilistic transformation that edits the attributes and labels in the data wrt group fairness and individual distortion. The goal of OP [45] is to determine a randomized mapping that transforms the original dataset into a new dataset satisfying the following properties: (i) discrimination control (limiting the dependence of the target class on the protected attributes); (ii) distortion control (using constraints to limit the extent of changes in the data mapping, ensuring that individuals with similar original

characteristics remain represented as close in the transformed space); and (iii) utility preservation (ensuring statistical similarity between the distributions of the transformed data and the original data).

Correlation Remover (CR). CR [46] applies a linear transformation to reduce the correlation between protected and non-protected attributes, while preserving the non-protected attribute values as much as possible. The goal is to preserve information by minimizing the difference between the transformed, non-protected attributes and their original values. At the same time, a constraint parameter enforces the reduction of correlation between the non-protected attributes and the protected one. The solution involves centering the protected attribute, fitting a linear regression model to predict non-protected attributes based on the protected one, and using the residuals (the part of the non-protected attributes uncorrelated with the protected one) as the transformed attributes.

5.2. In-processing mitigation techniques

The techniques in this type of processing create a new ML model that improves a given fairness metric without modifying the original dataset.

Adversarial Debiasing (AD). AD [47] learns a classifier to maximize prediction accuracy and reduce an adversary's ability to determine the protected attributes from the predictions. This approach leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit. More specifically, this algorithm simultaneously trains two neural network models: a predictor and an adversarial model. The predictor model seeks to minimize the prediction loss function while also ensuring that the adversary model cannot infer protected attributes from the predictor outputs.

Prejudice Remover (PR). PR [48] adds a discrimination-aware regularization term to the learning objective. PR reduces the mutual information between the target class and the protected attribute, according to a parameter that controls the degree of unfairness mitigation. Increasing the parameter generally damages accuracy because the technique is designed to remove prejudice by sacrificing accuracy in prediction. The result is a model that minimizes discrimination during training by penalizing patterns leading to unfair treatment of groups, and that is more likely to make unbiased predictions, reducing disparities between privileged and discriminated groups.

Exponential Gradient Reduction (ER). ER [49] formulates a fairness-constrained classification problem as a sequence of classification tasks, ensuring that the classifier prioritizes fairness constraints during training. By solving these tasks iteratively, the technique generates a randomized classifier that achieves minimal error while adhering to fairness constraints, even when the true data distribution is unknown. ER introduces sources of approximation, such as using empirical data instead of true values and setting limits on certain optimization parameters. Some errors, such as statistical errors due to empirical estimates, are unavoidable, while others deriving from optimization choices can be minimized by adjusting parameters or running more iterations. The ER technique ultimately aims to balance accuracy and fairness, influencing the quality of the final classifier's performance and fairness with these adjustments.

5.3. Post-processing mitigation techniques

The techniques in this type of processing, given fairness constraints, act on the ML predictions, not modifying the original dataset or the ML model.

Equalized Odds Post-Processing (EO). EO [35] solves the problem of modifying prediction labels while optimizing fairness metrics like EOD. EO creates fair predictions by adjusting an initial model to reduce discrimination based on a protected attribute. A pre-trained classifier is then post-processed and transformed to satisfy fairness constraints with no need to retrain the model.

Calibrated Equalized Post-Processing (CE). CE [50] changes prediction labels so as to satisfy different fairness metrics. The goal is to modify a classifier's predictions for a particular group so that the error rates match across groups, while also ensuring calibration. Calibration means that the classifier's probabilities associated with its predictions are aligned with actual target class frequencies, which helps maintain accuracy. To balance these errors, CE does not always use the classifier's original prediction. In particular, when calibration incurs too high a loss in terms of accuracy, based on an interpolation parameter that determines the trade-off between confidence and conservativeness, then CE replaces the classifier's prediction with a fixed mean prediction.

Reject Option Classification (RO). RO [51] offers a controlled way to improve fairness by strategically "rejecting" uncertain cases (i.e., instances whose probability is close to the threshold) and reassigning them, through tunable parameters, in a manner that compensates for possible discrimination. Instead of forcing a classification, uncertain cases are "rejected", i.e., they are left unlabeled in the initial stage. Given a classifier, RO examines its confidence level. If the classifier's probability for a positive or negative outcome is close to 0 or 1, the prediction is made with high certainty. However, when the probability is closer to 0.5, the prediction is more uncertain. The RO method defines a "critical region" using a threshold to identify these uncertain cases. Instances for which the classifier's confidence falls below the threshold are rejected. When an instance is rejected, the technique applies a fairness-aware rule: the instance is labeled positively if it belongs to a discriminated group, negatively otherwise. This approach is intended to counter potential biases and ensure fairer predictions for the discriminated group.

Threshold Optimizer (TO). TO [35] is designed to adjust classification thresholds for different protected groups to achieve a specified fairness criterion exactly, without leaving any residual disparity between groups. The primary objective of TO is to satisfy a chosen fairness constraint, such as EOP. For each protected attribute, TO applies a unique decision threshold to the predictions of a base classifier. This means that instead of using a single threshold for all individuals, the method customizes thresholds by group, allowing it to meet the fairness constraints more precisely. TO considers all possible thresholds for each group, selecting the best combination that satisfies the fairness constraint while maximizing the model's performance within that constraint. During this process, the distribution and sizes of the different groups are considered to balance the fairness constraints accurately across groups. To achieve an exact match in fairness metrics, TO may randomize predictions between two thresholds for a given group. This randomization is necessary when no single threshold can fully equalize the fairness metrics across groups.

6. Datasets

Guided by a healthcare domain expert, we selected 7 datasets from the health sector, containing data on prevalent diseases associated with mortality rates [52] and used in at least one healthcare publication for reliability. As required by our experimental framework, all of them have at least one binary protected attribute and a binary target class. We now describe each dataset and summarize the data-cleaning steps used to optimize the performance of the ML models and mitigation techniques. To ensure data quality and support fair model evaluation, we applied the following preprocessing steps to each dataset[3]:

Table 5
Overview of the preprocessed datasets chosen for the experiments, where P indicates probability.

Dataset	Year	Size	#Attrs.	Prot. Attr. g (discr. group d)	Positive label ($\ell = 1$) for target class y	$P(y = 1)$	$P(g = d)$	$P(y = 1, g = p)$	$P(y = 0, g = d)$
Diabetes Women	1988	677	9	Age (young)	Diabetic	0.33	0.35	0.43	0.85
Diabetes US	2008	73 450	37	Race (non-caucasian)	Diabetic	0.57	0.24	0.43	0.59
Stroke	2020	2882	9	Residence (rural)	Has stroke	0.05	0.49	0.05	0.95
Sepsis	2013	878	14	Sex (female), Age (young)	Not survived	0.04	0.40, 0.61	0.04, 0.07	0.96, 0.98
AIDS	1996	1818	24	Race (non-white), Homo (homosexual), Age (young)	Not survived	0.23	0.29, 0.66, 0.52	0.25, 0.21, 0.24	0.81, 0.76, 0.78
Myocardial	2020	1245	24	Sex (female)	Not survived	0.08	0.36	0.06	0.88
Alzheimer	2024	2149	33	Sex (female), Race (non-caucasian)	Has Alzheimer	0.35	0.51, 0.59	0.36, 0.36	0.66, 0.66

1. Protected attributes and target class were binarized to match fairness mitigation requirements, with group mappings clearly defined.
2. Tuples with missing values were removed to prevent bias and maintain data consistency.
3. Outliers were detected and eliminated using the IQR (Interquartile Range) method to reduce the influence of extreme values.
4. Feature selection was performed using a correlation matrix to retain the most relevant attributes for prediction.
5. Finally, the dataset was shuffled before model training to avoid ordering bias.

Table 5 presents key statistics for the cleaned datasets used in our experiments. For each dataset, we report the collection year or the date of the first publication that utilized it, the total number of records ("Size"), the number of attributes ("#Attrs"), the protected attribute(s), and the positive label used for the target class. In accordance with the domain expert, we intentionally selected real-world datasets with varying sizes, numbers of attributes, and collection years to ensure diversity. While two of the datasets are relatively small, this reflects the common reality of real-world healthcare studies, which often rely on limited sample sizes; indeed, a large size increases the cost of data collection and raises concerns about the quality of the collected data [53]. We also include relevant probabilities characterizing the dataset distribution:

- Target Class Distribution: the probability of a positive outcome, $P(y = 1)$;
- Protected Attribute Distribution: the probability of belonging to the protected group, $P(g = d)$;
- Joint Protected Attribute and Target Class Distribution: two joint probabilities of belonging to the protected group and a specific target class, $P(y = 1, g = p)$ and $P(y = 0, g = d)$.

Understanding the distribution is essential for studying fairness [54], since a dataset that does not adequately represent the population can lead to biased outcomes. To address this, we include both balanced ($P \in [0.4, 0.6]$, green-colored cells) and skewed ($P \notin [0.4, 0.6]$, orange colored cells) datasets wrt these probabilities to capture a comprehensive range of distributional characteristics relevant to fairness.

While Table 5 shows that no (*dataset*, *protected attribute*) pair, which we call a *configuration*, is balanced in all aspects, we can usefully categorize such configurations as follows:

- **S** (Skewed everywhere). 5 configurations: *Diabetes Women, Sepsis (age), Myocardial, AIDS (race), AIDS (homo)*.
- **1-B** (Balanced once). 5 configurations: *Stroke, Sepsis (sex), AIDS (age), Alzheimer (sex), Alzheimer (race)*, balanced wrt the Protected Attribute distribution.
- **2-B** (Balanced twice). 1 configuration: *Diabetes US*, balanced on Target Class and Joint distributions.

Regarding this categorization, our approach follows the rationale outlined in [54]. A dataset satisfies the representation rate requirement if the groups and the target class have similar counts, which in a binary setting translates to approximately 50% probability. We leveraged this criterion to differentiate our datasets into three distinct categories.

Diabetes Women is originally from the National Institute of Diabetes and Digestive and Kidney Diseases [55], first adopted in a publication in 1988 [56]. The objective of the dataset is to predict diabetes based on certain diagnostic measurements included in the dataset. The population for this study was the Pima Indian population near Phoenix, Arizona. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. This dataset is originally composed of 9 attributes, 768 tuples and one protected attribute *Age* (young vs. adult). After data cleaning, the dataset contains 677 tuples and 9 attributes.

Diabetes US [57] collects ten years of clinical care information from 130 US hospitals [58,59], with the objective to predict whether the patient was re-admitted to the hospital after being diagnosed with diabetes. The original dataset contains 101,766 tuples, 47 attributes, with *Race* (Caucasian vs. non-Caucasian) as a protected attribute; the target class is the re-admission of the patient. After data cleaning, the dataset contains 73,450 tuples and 37 attributes.

Stroke [60] collects patients' information with the objective to predict whether a patient is likely to have a stroke [61,62]. The dataset contains 5110 tuples, 10 attributes, with *Residence* of the patient (urban vs. rural) as a protected attribute; the target class is whether the patient had a stroke or not. After data cleaning the dataset is composed by 2882 tuples and 9 attributes.

Sepsis includes data from patients diagnosed with Systemic Inflammatory Response Syndrome (SIRS) and sepsis, two of the leading causes of in-hospital mortality. The dataset comprises information collected from 1257 eligible medical and surgical patients admitted to intensive care units (ICUs) of two hospitals between 2006 and 2013. The data was initially published in 2016 [63] and was subsequently used to investigate differences in prognostic clinical features between patients with SIRS and those with sepsis [64]. Each sample in the dataset represents a patient and includes 16 features. The target variable indicates whether the patient survived or died during their ICU stay, and there are two protected attributes: *Age* (young vs. adult) and *Sex* (female vs. male). After cleaning, the dataset contains 878 tuples and 14 attributes.

[3] To apply the framework to a new dataset, users are expected to perform similar preprocessing steps manually; however, we provide modular and customizable Colab notebooks to guide and support this process.

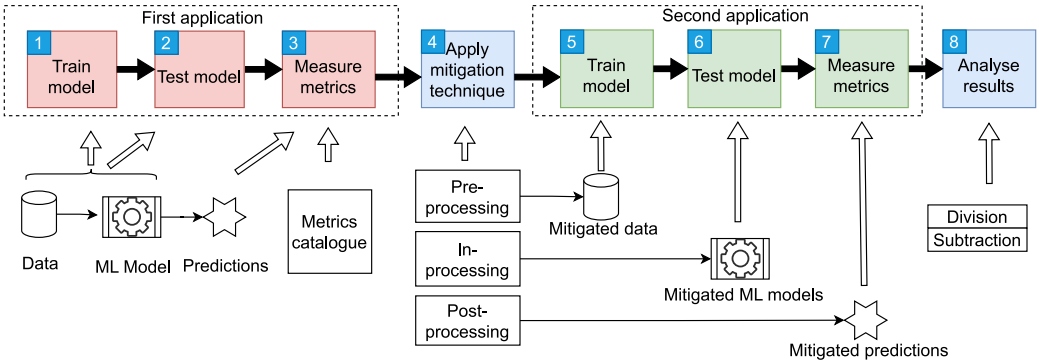


Fig. 1. FAIR-CARE pipeline used in the experiments.

AIDS [65] includes healthcare statistics and categorical information on patients diagnosed with AIDS. The dataset was designed in 1996 to evaluate the efficacy of two different treatment types and to predict whether a patient would survive within a specified time window [66]. The dataset includes various attributes describing the patient's general health, with 2139 tuples, 25 features, and four protected attributes: *Race* (non-white vs. white), *Sexual Orientation* (homosexual vs. non-homosexual), *Age* (young vs. adult), and *Sex* (female vs. male). Of these, we did not use *Sex* in the analysis, due to the insufficient representation of the female category (<20%). After cleaning, 1818 tuples and 24 attributes remained.

Myocardial [67] contains information on patients who have experienced myocardial infarction, focusing on complications arising after the event. The objective is to analyze the factors contributing to post-infarction complications and assess potential disparities based on demographic or clinical characteristics [68]. It has 1700 tuples, 111 attributes, one protected attribute (*Sex*), and a target class that indicates no complication (the patient survived) or a lethal outcome. After cleaning, 1245 tuples and 24 attributes remained.

Alzheimer [69] contains information about Alzheimer's disease, with features such as gender, ethnicity, and diagnostic status. The goal is to analyze the factors contributing to Alzheimer's diagnosis and evaluate disparities based on demographic characteristics. It contains 2149 tuples and 35 attributes, two protected attributes *Sex*, and *Race*; the target class indicates if the patient was diagnosed with Alzheimer. After cleaning, the number of tuples remained unchanged and we kept 33 attributes.

7. Execution and results

This section describes the execution of the pipeline used for our experiments (Section 7.1), and the corresponding results (Section 7.2).

7.1. Evaluation pipeline

This section describes the various steps of the pipeline used in the experiments, also summarized in Fig. 1. Essentially, the process includes a first application of classification according to the ML model (Steps 1–3) and the application of mitigation techniques (Step 4). After that, the ML models are applied again (Steps 5–7) and the results of the mitigations are reported (Step 8).

Step 1 receives as input the ML model and the dataset where the classification algorithm will be applied.

Step 2 consists of testing the ML model trained before. The test is key for the next step, where fairness metrics are used.

In **Step 3**, we measure the fairness of results computed in Step 2, which is essential to establish if the results of the ML models are biased according to any of the 11 fairness metrics in Table 3. This step also includes performance metrics for evaluating ML models.

Step 4 consists of the application of mitigation techniques.

Each mitigation technique is applied to a specific phase of the application of the ML model. The techniques applied in the *pre-processing* phase modify the original dataset, creating a modified version that can be used to train the ML model. This is represented in Fig. 1 with an association that connects the pre-processing to the dataset that feeds Step 5, which is the second train model.

Mitigation techniques that apply to the in-processing phase, rather than modifying the dataset, will modify the parameters of the model. In this case, we do not need to train again the ML model, which can be directly tested. This is represented in Fig. 1 with an association with the in-processing of the models that feeds the test model step of the second iteration.

The mitigation techniques that operate in the post-processing phase will modify the results of the first iteration. Therefore, we do not need to train or test the model again, but the results can be directly measured. This is represented in Fig. 1 with an association from the post-processing to the predictions that feeds Step 7.

Steps 5–7 consist of the second application of ML models. While the first three steps are in red to highlight their possibly unfair outcome, Steps 5 to 7 are colored in green since their fairness issues are mitigated. The second iteration, i.e., the repetition of the first three steps, is required to get a comparison to measure the effectiveness of the mitigation techniques. However, depending on the mitigation techniques, some steps of the second iteration might be skipped. Indeed the pre-processing mitigation performed in Step 4 will modify the dataset and, therefore, Steps 7–8 will have to be performed to train again the ML model, perform the tests and measure fairness on the results. The in-processing mitigation will modify the trained ML model. In this case, Step 5 will be skipped and the tests (Step 6) and the measures (Step 7) will be performed. Finally, post-processing mitigation directly modifies the results of the ML models, so only Step 7 will be executed.

Step 8 consists of aggregating and comparing the metrics of the first application (without the effect of the mitigation techniques) with the second application (after mitigation). As explained in Section 4, we used two approaches for the comparison of the results: by division and by subtraction. We finally aggregate the obtained results across multiple datasets, mitigation techniques, ML models, and both performance and fairness metrics. To assess whether the differences for different aspects (i.e., fairness and performance metrics, dataset category, and ML models) across mitigation strategies are statistically significant, we first apply a one-way ANOVA test for each metric. If one mitigation differs significantly, we proceed with Tukey's Honest Significant Difference (HSD) test to identify which specific mitigation pairs are significantly different. For interpretability, we visualize the results using box plots and compute rankings for each mitigation technique for each studied aspect. This combined statistical and visual analysis helps reveal overall trends, assess the consistency of mitigation techniques across models and datasets, and supports the identification of techniques that most effectively improve fairness and performance

(and for which ML models), also reporting some trade-offs between fairness and performance metrics.

Overall, we applied 6 ML techniques to 7 datasets, which, with the protected attributes, gave rise to 11 different configurations. We identified and used 12 mitigation techniques in the 3 phases of processing of ML models, thereby running $6 \times 11 \times 12 = 792$ experiments. During these, we measured fairness according to 11 metrics and performance according to 4 metrics. *This resulted in* $792 \times 4 = 3168$ *measures of performance and* $792 \times 11 = 8712$ *measures of fairness for the first iteration and as many for the second iteration (each taken twice, according to the two criteria for comparing metrics)*. To apply FAIR-CARE to a new dataset, users just need to follow the instructions provided in the repository's README and update the configuration file with the dataset-specific parameters.

7.2. Results

In the presentation of results, due to the large number of experiments, we refrain from showing here all the plots (more than 1000), the interested reader can find all the plots in a shared folder.[4] Before presenting the most important results, it is essential to emphasize the following key points:

Category-based Analysis: For each category of dataset configuration (see Section 6), we present a specific analysis of the impact of the distribution characteristics.

Comparison within Mitigation-Processing Types: We first start by evaluating techniques within each processing type (pre-, in-, or post-processing), since they differ in construction. We follow up with a cross-type comparison to analyze how techniques perform to provide a comprehensive view of the trade-offs and benefits across different types of mitigation techniques.

In-Processing Model Comparison: For in-processing techniques, we compared all six models (in which the "o" prefix refers to the original, unmitigated models) because each mitigation technique produces a new ML model that incorporates fairness adjustments. This allows a direct comparison of all models within this category. Pre-processing and post-processing techniques work with one ML model at a time, and, thus, we group our results by ML model. In the interest of space, we only do this for a representative selection of models.

Plot Representation: The results are visualized using bar plots. Although we also computed box plots with both mean and std dev, to avoid clutter, we only report the bar plots with just the mean. The X-axis represents the fairness metrics, while the Y-axis indicates their values, ranging from -1 to 1. Each bar corresponds to the value of a specific metric for a given mitigation technique, with an additional bar, labeled OR, for the original baseline (i.e., dataset before mitigation).

User-Oriented Insights: At the end of each section, we provide concise summaries that highlight actionable takeaways for practitioners. While based on our experimental datasets, these insights are designed to guide users applying FAIR-CARE to similar real-world healthcare scenarios.

7.2.1. 2-B datasets

For the *Diabetes US* dataset, which is considered "balanced" in terms of both the Target Class and Joint distributions, the initial results (i.e., without applying any mitigation techniques) are remarkably strong across all ML models. Regarding fairness, between 9 (for LORE and RAFO) and 11 out of 11 fairness metrics are satisfied using the "division" comparison. For the subtraction comparison, the initial results are similar: apart from RAFO, which satisfies 10 out of 11 fairness metrics, all the other ML models satisfy all fairness metrics. In terms of performance, BAGG achieves the highest Acc (0.616 ± 0.034), while DETR has the lowest Acc (0.540 ± 0.010) but delivers the best F_1 (0.465 ± 0.027). This experiment highlights that a balanced dataset can lead to optimal fairness outcomes. Notably, in this case, the models with the best performance (BAGG and DETR) also satisfy all fairness metrics.

Insight #1. *In our experiments on* 2-B *Diabetes US data, high-performing models like BAGG achieved competitive fairness outcomes—challenging the assumption that strong performance comes at the cost of fairness. This suggests that, in certain healthcare contexts, mitigation may not be necessary if both goals are met by default.*

Mitigation techniques were also applied to this dataset, and the results were consistent with those observed in the other categories. For this reason, we do not discuss them here, but instead provide a detailed analysis in the following paragraphs.

7.2.2. 1-B datasets

Table 6 reports the initial results (i.e., without any mitigation) for 1-B datasets. Each cell displays the number of satisfied fairness metrics using the "division" comparison: if we consider both mean and std dev, values range from 3 to 7 (from 4 to 11 if we only consider the mean) out of 11 fairness metrics. For the subtraction comparison, the initial results are not shown for space reasons, but are present in our project repository [11]. In particular, the results are very similar to those we describe below for the division approach, but the subtraction approach is generally less affected by the threshold, thereby sometimes causing more metrics to be satisfied.

The dataset that satisfies the smallest number of fairness metrics for all models is *Sepsis-sex* - highlighted in orange - and the best one is *Stroke* - highlighted in green. All models obtained similar fairness results, with ADBO (in orange) slightly worse than the others. Regarding performance, RAFO and BAGG achieve the highest Acc (ranging between 0.892 and 0.960) as well as strong F_1 values (between 0.685 and 0.900).

In this category, models with strong performance metrics also tend to exhibit good fairness outcomes. However, Table 6 shows that fairness results vary significantly across datasets, even when they are in the same category. Specifically, the number of satisfied fairness metrics ranges from a minimum of 3 to a maximum of 7.

Insight #2. *In our experiments, even balanced datasets like Sepsis-sex did not always lead to fair outcomes. Being in the* 1-B *category does not guarantee fairness, so practitioners should always assess fairness metrics explicitly.*

Although all dataset configurations have a different initial fairness profile, they all benefit in similar ways from mitigation effects from the various techniques. This allows us to select the *Alzheimer-sex* as the representative dataset for the *1-B* category and examine the effectiveness of pre-, in-, and post-processing mitigation techniques with the EXRT model. The chosen model has the worst performance on this configuration ($\texttt{Acc} = 0.837 \pm 0.073, \texttt{Pre} = 0.855 \pm 0.062, \texttt{Rec} = 0.657 \pm 0.109, F_1 = 0.740 \pm 0.094$), but achieves the best fairness (5 metrics satisfied considering both mean and std dev values, and 11 metrics considering only the mean).

Pre-processing. Fig. 2 shows the fairness metrics values after the application of pre-processing techniques RW, DI, LF, CR and OP. The OP technique improves most fairness metrics, considering both the mean and std dev (to avoid clutter, only mean values are shown in the plots) and proves to be the winning technique even though Acc degrades from 0.837 to 0.714 and F_1 0.740 from to 0.539.

In-processing. Fig. 3 shows fairness metrics results after the application of in-processing techniques AD, ER, and PR. PR demonstrates its strength in directly embedding fairness constraints into the training process, leading to substantial improvements in fairness metrics, specifically EOP, TEQ, FOR and FNP. Its Acc is still high (0.828) and also F_1 is significant (0.749).

[4] https://drive.google.com/drive/folders/1ioSw5K5CghWy4hrTNnlGFZQPoIiS67kh?usp=sharing.

Table 6
Satisfied fairness metrics with division comparison wrt mean and std dev (in brackets, only wrt the mean) for unmitigated 1-B datasets across all ML models.

	LORE	DETR	BAGG	RAFO	EXRT	ADBO
Stroke	6 (6)	7 (11)	6 (10)	6 (10)	6 (8)	6 (6)
AIDS-age	5 (7)	5 (10)	6 (10)	6 (11)	5 (8)	5 (11)
Sepsis-sex	3 (5)	3 (6)	3 (8)	3 (6)	3 (4)	3 (8)
Alzh.-sex	5 (11)	4 (9)	4 (7)	4 (7)	5 (8)	4 (7)
Alzh.-race	4 (9)	4 (8)	4 (10)	4 (11)	4 (10)	4 (5)

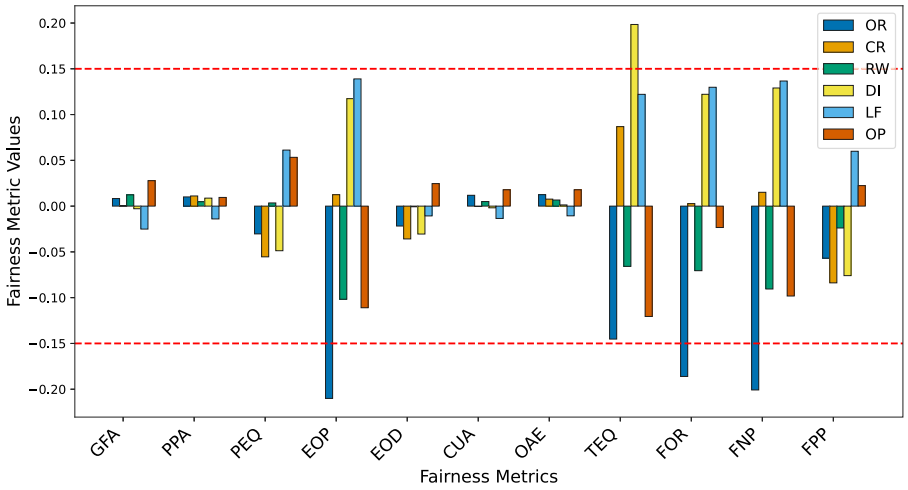


Fig. 2. Fairness metrics (division) after pre-processing mitigation on Alzheimer-sex with EXRT.

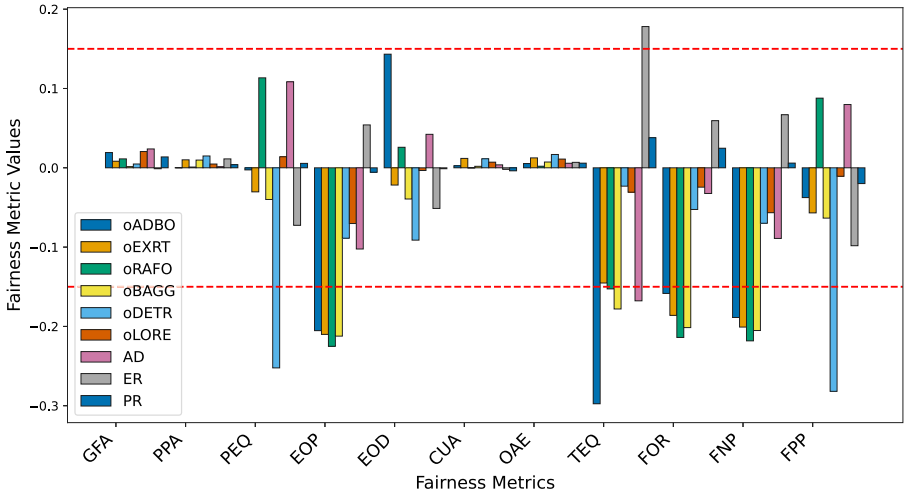


Fig. 3. Fairness metrics (division) after in-processing mitigation on Alzheimer-sex.

Post-processing. Fig. 4 shows fairness metrics results after the application of post-processing techniques RO, CE, TO, and EO. EO is the best choice for post-processing, as it effectively improves EOP, TEQ, FOR and FNP. The performance metrics for this technique are also similar to the original ones: Acc = 0.827 and $F_1 = 0.722$.

For the subtraction comparison, plots are not displayed for space reasons, but they are available in our project repository [11]. With this dataset, all three kinds of processing (pre-, in-, post-) lead to fairness results very similar to the ones obtained with the division approach.

Insight #3. *In our* 1-B *healthcare datasets, such as Alzheimer-sex, OP, PR and EO are the most effective mitigations for improving fairness in the pre-, in-, and post-processing types, respectively.*

7.2.3. S datasets

Table 7 presents the initial fairness results for datasets classified as "skewed" evaluated across all ML models. Each cell indicates the number of satisfied fairness metrics, computed using the "division" comparison. The datasets with the lowest number of satisfied fairness metrics across all models are *Diabetes Women* and *Sepsis-age* (highlighted in orange), while the dataset with the best fairness results is *AIDS-homo* (highlighted in green). Among the models, LORE, EXRT and ADBO—highlighted in green—are the best performers, whereas DETR performs the worst. Compared to the previous category (1-B), the number of satisfied fairness metrics is generally lower for skewed datasets, with one exception: *AIDS-homo* (in the S category) achieves slightly higher fairness values than *Sepsis-sex* (in the 1-B category). For the subtraction comparison, the initial results are not shown for

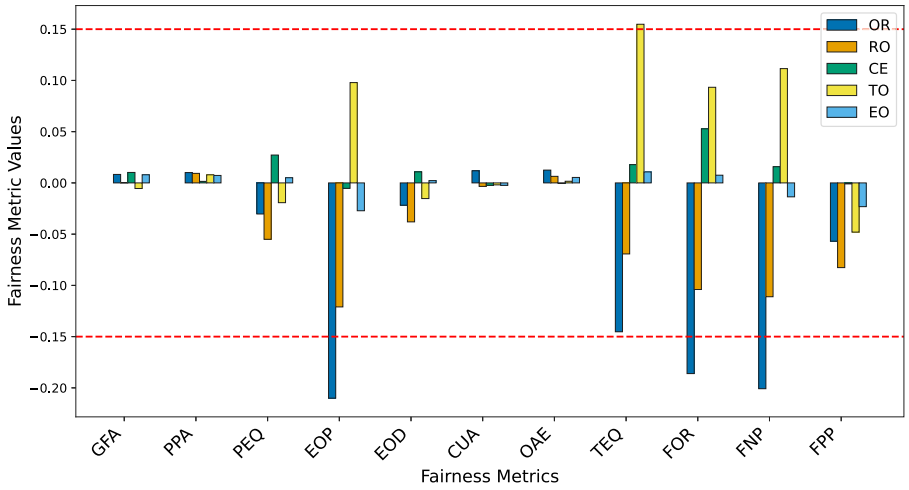


Fig. 4. Fairness metrics (division) after post-processing mitigation on Alzheimer-sex with EXRT.

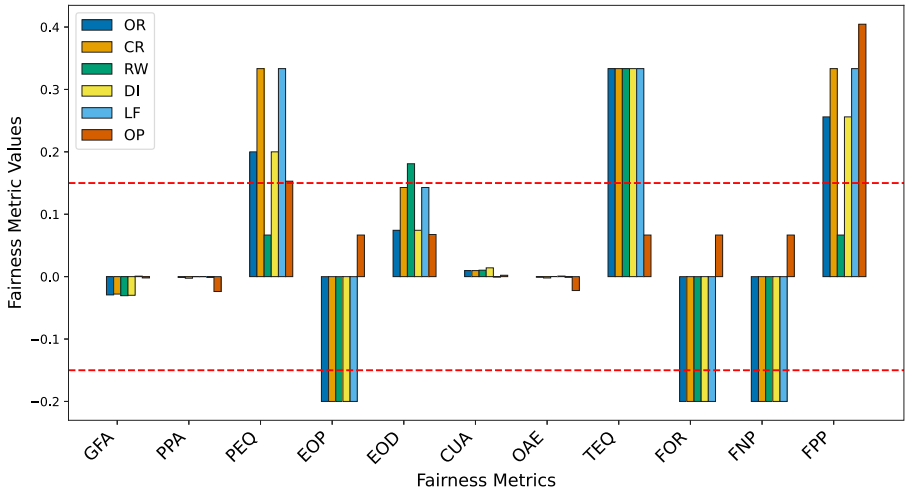


Fig. 5. Fairness metrics (division) after pre-processing mitigation on Myocardial with DETR.

Table 7
Satisfied fairness metrics with division comparison wrt mean and std dev (in brackets, only wrt the mean) for S datasets across all ML models.

	LORE	DETR	BAGG	RAFO	EXRT	ADBO
Diabetes W.	4 (4)	2 (5)	3 (6)	2 (5)	3 (4)	4 (4)
Sepsis-age	3 (4)	3 (8)	3 (5)	3 (4)	3 (4)	3 (8)
Myocardial	4 (8)	4 (5)	4 (6)	5 (5)	4 (5)	4 (6)
AIDS-race	4 (6)	4 (11)	4 (9)	4 (10)	5 (7)	5 (7)
AIDS-homo	5 (7)	5 (7)	5 (6)	5 (7)	5 (11)	4 (11)

space reasons. Later in this section, we shall explicitly compare the two approaches, showing that the subtraction approach is less restrictive than the division approach, possibly leading to more satisfied metrics.

Insight #4. *In our experiments, skewed datasets like AIDS-homo (S category) typically satisfy fewer fairness metrics than partially balanced (1-B) ones. However, a skewed distribution does not always imply severe unfairness, so fairness should still be measured empirically.*

For class S ("skewed") we show multiple datasets and multiple ML models, with both "subtraction" and "division" comparisons:

- pre-processing: *Myocardial*, 3 ML models, only division;
- in-processing: *Diabetes Women* division and subtraction;
- post-processing: *AIDS-race*, BAGG model, division and subtraction.

Pre-processing. Figs. 5, and 6 present fairness results for the *Myocardial* dataset after the application of pre-processing techniques across the DETR, BAGG and RAFO models respectively. In this dataset, considering only the mean values, between 4 and 6 metrics are originally satisfied, but after mitigation 9 metrics out of 11 are satisfied using OP. Modifying the dataset, OP slightly changes accuracy (from $\texttt{Acc} = 0.994 \pm 0.003$ to $\texttt{Acc} = 0.946 \pm 0.014$) and precision (from $\texttt{Pre} = 0.970 \pm 0.026$ to $\texttt{Pre} = 0.975 \pm 0.050$) but it drastically reduces recall and thus F_1 (from $\texttt{Rec} = 0.959 \pm 0.054$ and $F_1 = 0.963 \pm 0.019$ to $\texttt{Rec} = 0.358 \pm 0.073$ and $F_1 = 0.519 \pm 0.078$). This fall-off in Rec is reflected on FPP values, which increased after applying the mitigation.

In-processing. Fig. 7 plots fairness results for *Diabetes Women* with division and subtraction comparison after the application of in-processing techniques. These plots show that no in-processing technique significantly improves fairness metrics; this also occurs in the experiments

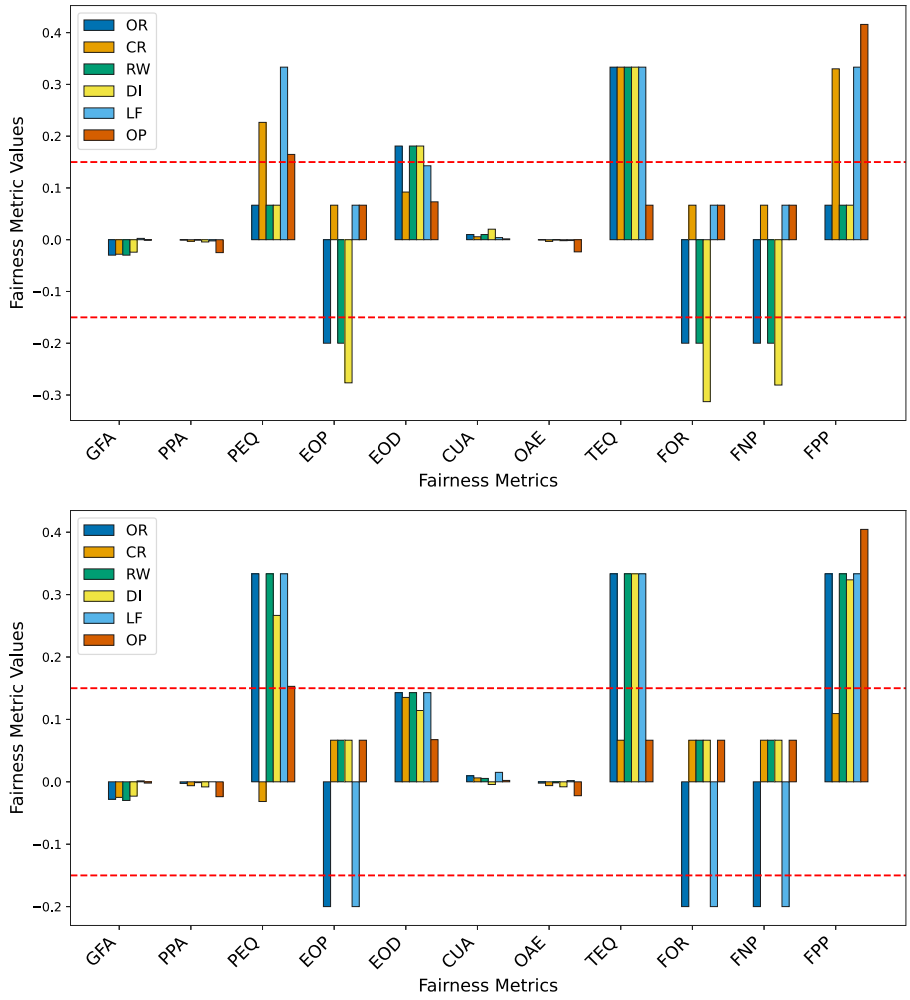


Fig. 6. Fairness metrics (division) after pre-processing mitigation on Myocardial with BAGG (top) and RAFO (bottom).

performed with other S datasets (not shown for brevity). Furthermore, comparing the two plots shows that division and subtraction are generally in agreement, but in some cases, such as FNP and FPP, division is more stringent, and these two metrics are satisfied only by subtraction.

Post-processing. Fig. 8 plots fairness results for *AIDS-race* with BAGG using division and subtraction. Considering only the mean, CE seems the most powerful technique, while taking into account both mean and std dev, the EO technique is the best performer using division. With subtraction, EO is even more powerful, as it increases the number of satisfied fairness metrics from 5 to 10, considering both mean and std dev.

Regarding performances, EO maintains values near the original ones: from Acc $= 0.886 \pm 0.016$ to 0.844 ± 0.023, from Pre $= 0.783 \pm 0.090$ to 0.682 ± 0.112, from Rec $= 0.687 \pm 0.071$ to 0.607 ± 0.090 and from $F_1 = 0.729 \pm 0.068$ to 0.636 ± 0.082.

Insight #5. *In skewed datasets like AIDS-race, (i) OP (pre) improves fairness but significantly reduces* Rec *and* F_1*; (ii) in-processing techniques show limited fairness improvements; (iii) EO (post) improves fairness, with minimal impact on performance; (iv) division is more stringent than subtraction in fairness evaluation.*

8. Findings, considerations and guidelines

Firstly, we illustrate our findings, presenting the best-performing techniques for each performance and fairness metric, briefly presenting the trade-off between them. Secondly, we outline some final considerations along with five practical guidelines for users.

8.1. Findings

In the following tables, results are aggregated over all ML models, and dataset categories. To assess whether differences for fairness and performance metrics among mitigation strategies are statistically significant, we first apply a one-way ANOVA test for each metric. If the resulting p-value is greater than 0.05, we fail to reject the null hypothesis, indicating that no statistically significant difference exists across the mitigation techniques for that particular metric. However, if the p-value is less than 0.05, we conclude that at least one mitigation differs significantly, and we proceed with Tukey's Honest Significant Difference (HSD) test to identify which specific mitigation pairs are significantly different. For interpretability, we visualize the results using box plots and compute rankings for each mitigation technique for each studied aspect. These rankings are extracted doing, for each dataset, the mean of the metric scores across multiple ML models, and then the average for all datasets within its category. Thus, in the following tables, the symbol > means "statistically significantly better than" and the symbol ≈ means "no statistical difference".

8.1.1. Findings regarding performance metrics

Table 8 presents a ranking, along with the best-performing technique (in bold), for each performance metric across different dataset categories and mitigation processing types. For pre-processing techniques, LF is the top-performing mitigation across all datasets (around 80% of configurations for S datasets, 60% for 1-B datasets and 100% for 2-B datasets) and performance metrics. LF is designed to modify

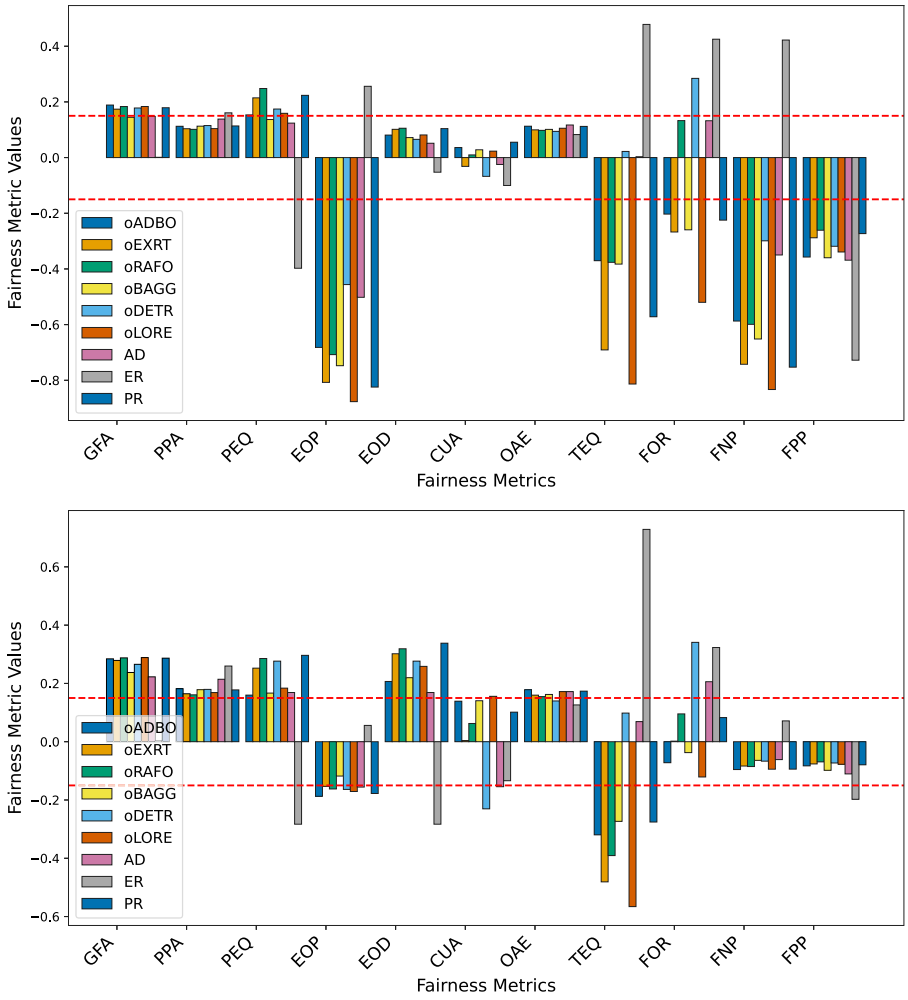


Fig. 7. Fairness metrics after in-processing mitigation on the Diabetes Women dataset. Top: division method. Bottom: subtraction method.

Table 8
Ranking of mitigation techniques for dataset category for all performance metrics.

	Pre-processing	In-processing	Post-processing
S Acc	**LF** > DI ≈ CR ≈ RW ≈ OP	**ER** ≈ PR ≈ AD	**CE** ≈ RO ≈ TO ≈ EO
1-B Acc	**LF** ≈ RW ≈ DI ≈ CR > OP	**PR** ≈ ER ≈ AD	**RO** ≈ CE ≈ TO > EO
2-B Acc	**LF** > DI ≈ CR ≈ RW ≈ OP	**AD** ≈ PR ≈ ER	**CE** ≈ TO ≈ EO ≈ RO
S Pre	**LF** > CR ≈ DI ≈ RW > OP	**ER** ≈ PR ≈ AD	**CE** ≈ RO ≈ TO ≈ EO
1-B Pre	**LF** > RW ≈ DI ≈ CR > OP	**ER** ≈ PR ≈ AD	**RO** ≈ TO ≈ CE ≈ EO
2-B Pre	**LF** ≈ OP ≈ DI ≈ CR ≈ RW	**PR** ≈ AD ≈ ER	**RO** ≈ CE ≈ TO ≈ EO
S Rec	**LF** > RW ≈ DI ≈ CR > OP	**ER** ≈ PR ≈ AD	**EO** ≈ CE ≈ RO ≈ TO
1-B Rec	**LF** > RW ≈ CR ≈ DI > OP	**ER** ≈ PR ≈ AD	**EO** ≈ RO ≈ TO ≈ CE
2-B Rec	**LF** > OP ≈ DI ≈ CR ≈ RW	**ER** ≈ AD ≈ PR	**RO** > TO ≈ EO ≈ CE
S F_1	**LF** > RW ≈ CR ≈ DI > OP	**ER** ≈ PR > AD	**CE** ≈ RO ≈ TO ≈ EO
1-B F_1	**LF** > RW ≈ DI ≈ CR > OP	**ER** ≈ PR ≈ AD	**RO** ≈ TO ≈ CE ≈ EO
2-B F_1	**LF** > OP ≈ DI ≈ CR ≈ RW	**ER** ≈ AD ≈ PR	**RO** > TO ≈ EO ≈ CE

the data distribution in a way that minimizes performance degradation, making it the most effective pre-processing approach. For in-processing, ER is most often the leading technique, followed closely by PR and AD, and all three techniques actually perform worse than the "original" (OR i.e., no mitigation applied). The similar performance of all in-processing techniques suggests that they exhibit comparable behavior across multiple datasets and metrics. Finally, for post-processing, RO is the dominant technique, often emerging as the best for all performance metrics across all datasets and performing slightly better wrt "original" for Pre, Rec and F_1 (in the 60% of cases for S and 1-B datasets and 100% for 2-B datasets). The difference between all post-processing techniques is not statistically significant for the ANOVA and Tukey's HSD test; thus, the difference is negligible for all performance metrics.

Fig. 9 shows the overall performance values on 1-B and S datasets, respectively, making it clear that, for all mitigation techniques, the performance metrics have similar values and that LF is the best mitigation technique regarding performance across all mitigation types (considering at the same time pre-, in- and post-processing).

From the empirical results, we can also extract the best ML models regarding performance: ADBO, RAFO and EXRT are the best performers in all dataset categories (S, 1-B and 2-B). Specifically, ADBO consistently achieves the best Acc. These ensemble methods obtain higher results in terms of all performance metrics (all around 0.98 and 0.95) than the other models (DETR, LORE, BAGG), which obtain slightly lower but still reasonable values (between 0.81 and 0.93).

Insight #6. *Performance Metrics: In our experiments, for pre-processing LF is the best-performing strategy, for in-processing all the mitigations perform similarly, with ER leading in most cases, for post-processing RO consistently offered high performance. ML models: ADBO, RAFO and EXRT are the best performers, and ADBO achieves the highest* Acc.

8.1.2. Findings regarding fairness metrics

Table 9 presents the best-performing technique for each fairness metric (both division and subtraction comparison) across different dataset categories and mitigation processing types. In each cell, if there is only one entry value, that means that it is the same for the division

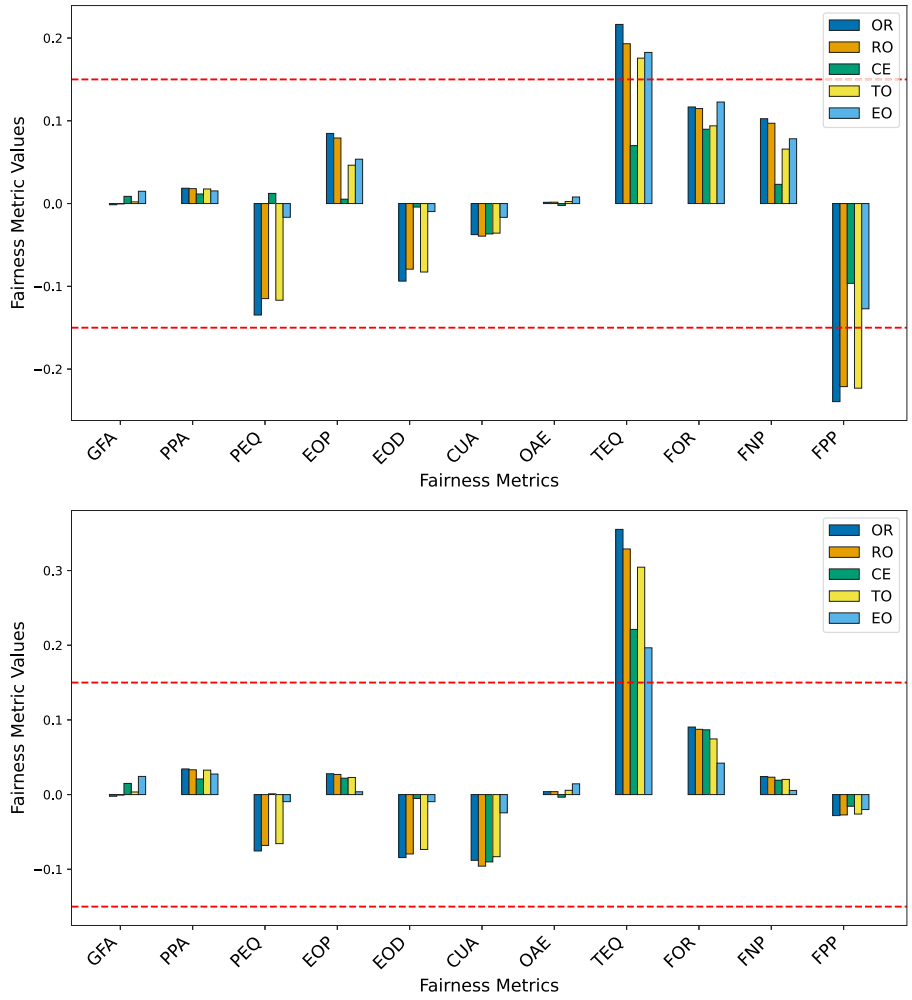


Fig. 8. Fairness metrics after post-processing mitigation on the AIDS-race dataset. Top: division method. Bottom: subtraction method.

and the subtraction comparisons; otherwise the first value reports the results for the division comparison, and the second one (i.e., after the '/') reports the value for the subtraction comparison. The same table also highlights possible statistical differences: the symbol '✓' means that there are no statistical differences, while the symbol '!' indicates that the mitigation techniques or the original baseline (OR) have a statistical difference with other techniques. In this case, the user can choose a technique that is statistically no different but reaches a better trade-off between performance and specific fairness metrics. For the ANOVA and Tukey's HSD test, almost all mitigation techniques do not exhibit statistical differences, especially the in-processing ones.

The results highlight only the best technique, but, in only a few cases, only one technique can improve a specific fairness metric. Some cells have different techniques for division and subtraction (i.e., split with /), suggesting sensitivity to the comparison method. There are distinct behaviors between pre-, in-, and post-processing techniques, and certain techniques appear repeatedly across multiple fairness metrics: for pre-processing, OP shows strength especially for S datasets for 7 out 11 fairness metrics, indicating more consistent effectiveness compared to the results on 1-B datasets; regarding in-processing, for S datasets results are more varied compared to the 1-B configuration. Unlike pre-processing, the number of competing techniques in in-processing is lower, making fairness results' differences between the winning and other techniques less pronounced. In several cases, multiple techniques share the top spot for a given metric. EO is the dominant post-processing technique, outperforming others in 10 out of 11 fairness metrics for 1-B datasets. For S datasets, EO remains the leading mitigation, excelling in 6 out of 11 metrics.

From the empirical results, we can also extract the best ML models regarding fairness metrics. Regarding the best ML models for each fairness metric, there are no statistical differences among all the models; the top performer for all dataset categories is ADBO, which obtains fairness metric values nearest to 0 wrt other models.

To further summarize our findings, we rank mitigation techniques across all fairness metrics by dataset category. Table 10 reports, for each category, an ordered list from the best (in bold) to the worst technique according to the number of satisfied fairness metrics: the more fairness metrics the technique satisfies, the better the ranking. In the table, the symbol > means "better than" and ≈ "as good as". We include 2-B datasets for completeness, although they would not require mitigation. It is worth noting that, for both 1-B and S datasets, the winning techniques are the same within the same processing type: OP is the best for pre-processing, PR for in-processing, and EO for post-processing techniques.

Insight #7. *Fairness Metrics. Pre: OP performs best, especially for S datasets. LF and RW also show good results. In: PR and AD perform similarly in S and 1-B datasets, while ER dominates in 2-B datasets. Post: EO is the most effective technique across the majority of datasets and fairness metrics. ML models. There are no statistical differences among all the models; ADBO obtains the best fairness metric values wrt to the other models.*

8.1.3. Trade-off between performance and fairness metrics

Finally, we briefly describe the trade-off between performance and fairness metrics. For each fairness metric and performance metric, we

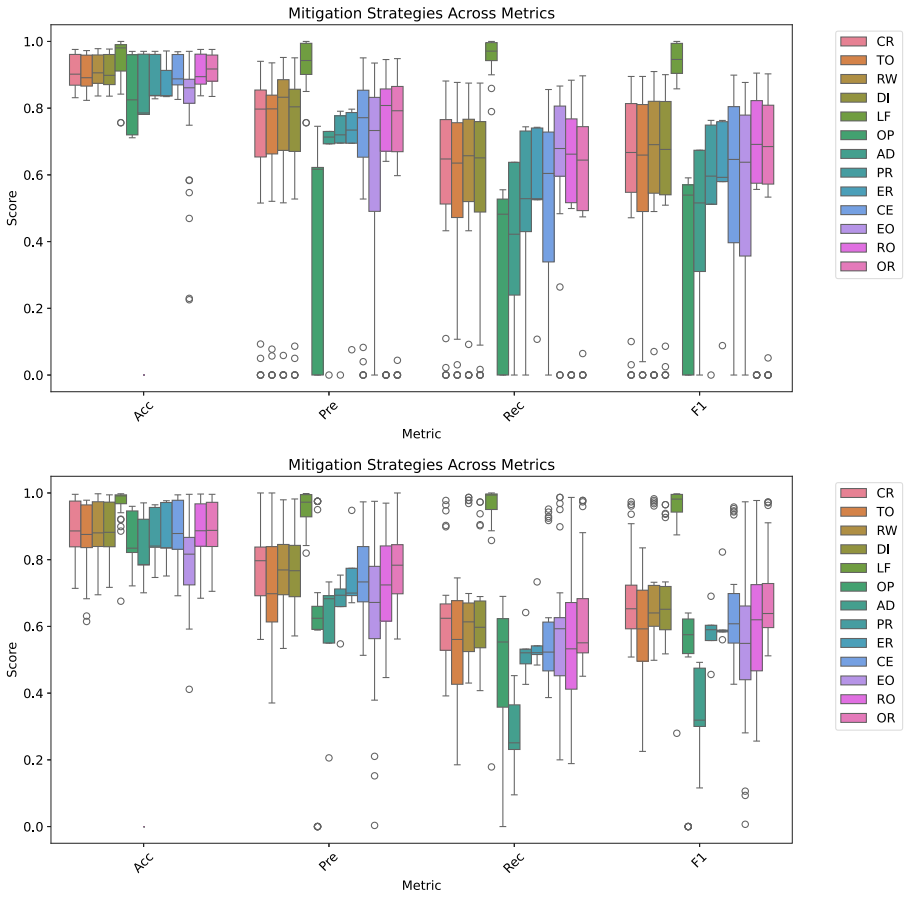


Fig. 9. Box plots of overall performance metrics. Top: 1-B datasets. Bottom: S datasets.

Table 9
Best mitigation techniques by fairness metric across dataset categories.

Metric	Pre 1-B	Pre S	In 1-B	In S	Post 1-B	Post S
GFA	OR !	OP ✓	ER ✓	ER ✓	EO/TO ✓	EO ✓
PPA	CR ✓	OP !	PR/AD ✓	OR !	EO/CE ✓	CE ✓
PEQ	OP/LF ✓	CR ✓	PR/ER ✓	PR/ER ✓	EO ✓	EO/CE ✓
EOP	DI/CR ✓	LF ✓	ER/AD ✓	ER ✓	CE ✓	RO ✓/EO !
EOD	RW !/LF ✓	OP/LF ✓	ER/AD ✓	PR/ER ✓	CE/EO ✓	TO/CE ✓
CUA	LF/DI ✓	OP/LF ✓	ER ✓/OR !	ER/PR ✓	TO/EO ✓	TO !
OAE	LF ✓	OP/LF ✓	OR !	PR ✓	EO/RO ✓	RO ✓
TEQ	OR !/CR ✓	OP/CR ✓	PR/ER ✓	AD/PR ✓	OR !/EO ✓	EO !
FOR	LF/DI ✓	DI ✓/OR !	ER/PR ✓	ER/AD ✓	RO/EO ✓	RO !/EO ✓
FNP	DI ✓/CR !	LF ✓	OR !/PR ✓	ER ✓	EO ✓	EO !
FPP	OP !	OP/LF !	PR/AD ✓	AD ✓/OR !	EO ✓	CE ✓

Table 10
Ranking of mitigation techniques for dataset category for all fairness metrics.

	Pre-processing	In-processing	Post-processing
S	**OP** > LF > RW > DI ≈ CR	**PR** ≈ AD ≈ ER	**EO** > RO ≈ TO > CE
1-B	**OP** > LF > RW > CR > DI	**PR** > AD ≈ ER	**EO** > CE > RO ≈ TO
2-B	**RW** > OP ≈ LF ≈ CR ≈ DI	**ER** > PR ≈ AD	**TO** > EO ≈ CE ≈ RO

plot the trade-off for each dataset category (S, 1-B and 2-B) and for all datasets together; the full set of plots (more than 500) is available in the shared folder. As an example, Fig. 10 presents the trade-off of a specific fairness metric (GFA, often used for comparison) and Acc for S datasets. The different shapes represent the different ML models and in-processing mitigation techniques (which do not depend on a ML model) and the different colors represent the original baseline and the pre-, in- and post-processing techniques. In-processing techniques are both model and mitigation, thus in the scatter plot there will be only one combination for each specific color and shape for each in-processing technique. In this specific example, GFA is satisfied by all mitigation techniques (since its absolute value is always lower than $\tau = 0.15$). The best technique for Acc is LF, while, for almost all ML models, OP, PR, TO, and EO reach the best GFA values.

Generally, after applying the winning mitigation techniques presented before, Acc and F_1 decrease, on average, by around 0.1, although OP in many cases improved both. In the worst case, Acc and F_1 decreased by 0.2 and, resp., 0.6 with OP, by 0.1 and 0.3 with PR, and by 0.7 and 0.3 with EO.

Insight #8. *Fairness-performance trade-off. In our experiments, applying mitigation techniques often reduced* Acc *and* F_1*. However, OP occasionally improved both. LF achieved the highest accuracy overall, while OP, PR, and EO consistently yield the best fairness results—highlighting the need to balance priorities based on the task.*

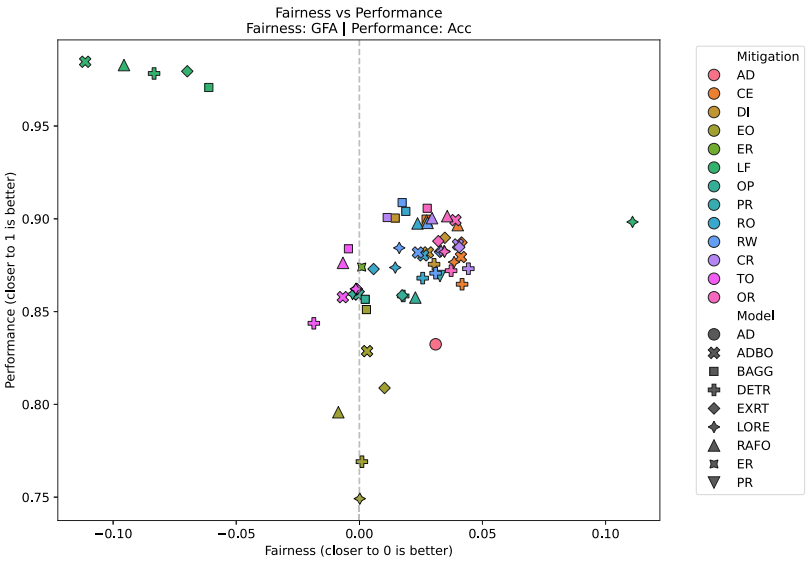


Fig. 10. Scatter plot of trade-off between GFA fairness metric and Acc for S datasets.

8.2. Final considerations

We outline below the general limitations and key considerations encountered in our experiments.

Balanced datasets. We used only one dataset in the 2-B category (i.e., the most balanced one), due to two key reasons: (i) in the healthcare sector, datasets balanced on multiple aspects are rare, as the populations from which the data are drawn are often limited, and the primary focus of such datasets is typically disease prediction, which is inherently imbalanced; and (ii) in most balanced datasets, fairness metrics are often naturally satisfied, reducing or eliminating the need for mitigation techniques.

Reliability of results. Our framework includes a shuffling step during data cleaning to eliminate order bias, followed by k-fold cross-validation to ensure robust estimation. For each configuration, we report both the mean and standard deviation of performance and fairness metrics across the folds. We did not repeat the entire cross-validation process multiple times since repeated full runs tend to yield marginal benefit in reliability [70], and in large-scale experiment settings like ours, it is not feasible due to resource constraints.

Defaults. We employed the techniques as implemented in their respective libraries, using default settings wherever possible, for three main reasons: (i) parameter tuning is often highly context-dependent, and we aimed to ensure generalizability across different scenarios; (ii) certain techniques, such as OP, rely on specific functions to transform and optimize the dataset, making default settings essential for proper functionality; and (iii) using default parameters enhances reproducibility.

Specificities of techniques. Initially, LF showed extremely poor performance values across all datasets. To address this, we implemented several adjustments: reducing the number of splits, modifying the dataset to a balanced subset of ∼1000 samples and selecting only the most relevant attributes. Still, LF remained highly sensitive to parameter tuning, so we iteratively adjusted k, the most influential parameter [35], to optimize model performance. OP required all dataset attributes to be in binary form. Due to suboptimal performance on larger datasets, only the five attributes most strongly correlated with the target class were considered. RO did not converge with large datasets. So for *Diabetes US*, we kept a maximum of 5000 tuples and performed dimensionality reduction keeping the 5 features most correlated with the target. EO frequently encountered a division by zero, particularly when underrepresented demographic groups were present in the training or test datasets (e.g., in *Sepsis*). We then reduced the splits (initially 5) to a value that fixed the error while preserving their utility.

Execution time. The runtime of the mitigation notebooks were recorded for every configuration (i.e., dataset and protected attribute), as measured on Colab using a T4 GPU. Each mitigation notebook took approximately 500 s to execute.

8.3. Practical guidelines

Drawing from our empirical findings, we outline five actionable recommendations to help practitioners effectively apply FAIR-CARE to healthcare datasets.

1. **Start by evaluating the baseline model — it may already be fair enough.** Before assessing fairness, users must ensure that their dataset is appropriately cleaned and the user should identify which fairness metrics align with the ethical priorities of the application domain. Once these steps are complete, evaluating the baseline model may reveal that the chosen fairness metrics are already satisfied and mitigation might not be necessary.
2. **Be aware of the fairness–performance trade-off when applying mitigation.** Fairness mitigation may come at the cost of predictive performance. Users should first clarify which performance metrics are most critical for their task. This helps guide acceptable trade-offs. If fairness is the primary concern, techniques like OP, PR, and EO tend to yield the best improvements. For a better accuracy–fairness balance, consider LF, ER, and RO.
3. **Select mitigation methods based on the analyzed dataset and processing preference.** Select the mitigation type based on your goal and willingness to alter data, models, or outputs. Each mitigation stage has trade-offs: practitioners should choose based on how invasive or transparent they want the mitigation to be. Use also the dataset category to guide your choice: For skewed (S) and partially balanced (1-B) datasets:

   - Pre-processing: OP
   - In-processing: PR
   - Post-processing: EO

   For fully balanced (2-B) datasets:

   - Pre-processing: RW
   - In-processing: ER
   - Post-processing: TO

4. **Select the ML model that best fits the users' needs - if predictive performance is critical, selecting the right ML model is just as important as mitigation.** Among ML models, ADBO, RAFO, and EXRT are top performers. ADBO achieve the highest accuracy and best fairness results. However, these models are less interpretable than simpler models like logistic regression or decision trees. If explainability is less critical in your context, these models may help compensate for performance drops from mitigation.
5. **Choose fairness metric types (division vs. subtraction) based on clarity and context.** Division-based metrics are often easier to interpret for non-technical users and may be preferred in public reporting settings. However, subtraction-based metrics can be useful when small disparities matter. The choice should reflect the intended audience and context.

9. Conclusions

High-risk applications, like healthcare prediction, should grant fairness in the results computed by the ML algorithms. In this paper, we have proposed an evaluation pipeline, called FAIR-CARE, that we used to perform a systematic assessment of numerous unfairness mitigation techniques, through an extensive set of ML algorithms, fairness and performance metrics. Our tests covered many different healthcare datasets, overall largely exceeding the amount of configurations considered by previous studies. Our study highlights that fairness and performance can coexist, and mitigation techniques are a suitable way to achieve this; their effectiveness, however, depends on a number of factors, including the dataset category and the processing stage, whose impact we have analyzed in depth in this paper. Future work includes a deeper investigation of parameter tuning and the integration of FAIR-CARE in ML systems.

CRediT authorship contribution statement

Chiara Criscuolo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Mattia Salnitri:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Davide Martinenghi:** Writing – review & editing, Supervision, Methodology.

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.

Acknowledgments

We thank Professor L. Tanca for her support, T. Dolci for assistance in the initial phase, S. Perini, S. Balleros and D. Eraso for help with experiments, and D. Chicco for guidance on healthcare datasets. This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU and from the Italian PRIN project 2022XERWK9 "S-PIC4CHU" – Semantics-based Provenance, Integrity, and Curation for Consistent, High-quality, and Unbiased data science. We also want to thank the anonymous reviewers for the insightful comments that improved the quality of the paper.

Data availability

Data and code is available at the shared repository and link in the paper.

References

[1] The European Parliament, Artificial Intelligence Act, Official Journal of the European Union, 2024, URL https://artificialintelligenceact.eu/the-act/, Last accessed on January 30, 2025.
[2] Z. Chen, J.M. Zhang, F. Sarro, M. Harman, MAAT: a novel ensemble approach to addressing fairness and performance bugs for machine learning software, in: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2022, pp. 1122–1134.
[3] V. De Martino, G. Voria, C. Troiano, G. Catolino, F. Palomba, Examining the impact of bias mitigation algorithms on the sustainability of ml-enabled systems: A benchmark study, J. Syst. Softw. (2025) 112458.
[4] S. Biswas, H. Rajan, Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 642–653.
[5] A. Fabris, S. Messina, G. Silvello, G.A. Susto, Algorithmic fairness datasets: the story so far, Data Min. Knowl. Discov. 36 (6) (2022) 2074–2152.
[6] A. Parziale, G. Voria, G. Giordano, G. Catolino, G. Robles, F. Palomba, Fairness on a budget, across the board: A cost-effective evaluation of fairness-aware practices across contexts, tasks, and sensitive attributes, in: Across the Board: A Cost-Effective Evaluation of Fairness-Aware Prac- tices Across Contexts, Tasks, and Sensitive Attributes.
[7] A.L. Beam, I.S. Kohane, Big data and machine learning in health care, Jama 319 (13) (2018) 1317–1318.
[8] V. Gulshan, L. Peng, M. Coram, M.C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al., Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs, Jama 316 (22) (2016) 2402–2410.
[9] J. Krause, V. Gulshan, E. Rahimy, P. Karth, K. Widner, G.S. Corrado, L. Peng, D.R. Webster, Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy, Ophthalmology 125 (8) (2018) 1264–1272.
[10] L. Seyyed-Kalantari, H. Zhang, M.B. McDermott, I.Y. Chen, M. Ghassemi, Underdiagnosis bias of artificial intelligence algorithms applied to chest radiographs in under-served patient populations, Nature Med. 27 (12) (2021) 2176–2182.
[11] C. Criscuolo, FAIR-CARE GitHub repository, 2025, https://github.com/chiaracriscuolo/FAIR-CARE, Last accessed on January 30, 2025.
[12] C. Criscuolo, FAIR algorithm - COLAB drive project, 2025, https://drive.google.com/drive/folders/182YKE0bNOltAezFfcEVEy7-FwXemlWX8?usp=sharing, Last accessed on January 30, 2025.
[13] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2021) 1–35.
[14] D. Pessach, E. Shmueli, A review on fairness in machine learning, ACM Comput. Surv. 55 (3) (2022) 1–44.
[15] S. Caton, C. Haas, Fairness in machine learning: A survey, ACM Comput. Surv. 56 (7) (2024) 1–38.
[16] Q. Feng, M. Du, N. Zou, X. Hu, Fair machine learning in healthcare: A survey, IEEE Trans. Artif. Intell. (2024).
[17] A. Balayn, C. Lofi, G.-J. Houben, Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems, VLDB J. 30 (5) (2021) 739–768.
[18] A. Rajkomar, M. Hardt, M.D. Howell, G. Corrado, M.H. Chin, Ensuring fairness in machine learning to advance health equity, Ann. Intern. Med. 169 (12) (2018) 866–872.
[19] I.Y. Chen, E. Pierson, S. Rose, S. Joshi, K. Ferryman, M. Ghassemi, Ethical machine learning in healthcare, Annu. Rev. Biomed. Data Sci. 4 (1) (2021) 123–144.
[20] J. Xu, Y. Xiao, W.H. Wang, Y. Ning, E.A. Shenkman, J. Bian, F. Wang, Algorithmic fairness in computational medicine, EBioMedicine 84 (2022).
[21] S.A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E.P. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 329–338.
[22] G.P. Jones, J.M. Hickey, P.G. Di Stefano, C. Dhanjal, L.C. Stoddart, V. Vasileiou, Metrics and methods for a systematic comparison of fairness-aware machine learning algorithms, 2020, arXiv preprint arXiv:2010.03986.
[23] M. Hort, J.M. Zhang, F. Sarro, M. Harman, Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods, in: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 994–1006.
[24] M.T. Islam, A. Fariha, A. Meliou, B. Salimi, Through the data management lens: Experimental analysis and evaluation of fair classification, in: Proceedings of the 2022 International Conference on Management of Data, 2022, pp. 232–246.
[25] N. Lässig, M. Herschel, O. Nies, FAIRCR–an evaluation and recommendation system for fair classification algorithms, in: 2024 IEEE 40th International Conference on Data Engineering, ICDE, IEEE, 2024, pp. 5473–5476.

[26] Z. Chen, J.M. Zhang, F. Sarro, M. Harman, A comprehensive empirical study of bias mitigation methods for machine learning classifiers, ACM Trans. Softw. Eng. Methodol. 32 (4) (2023) 1–30.

[27] M.M. Churpek, T.C. Yuen, C. Winslow, D.O. Meltzer, M.W. Kattan, D.P. Edelson, Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards, Crit. Care Med. 44 (2) (2016) 368–374.

[28] J. Larson, M. Roswell, Compas US dataset, 2016, URL https://github.com/propublica/compas-analysis.

[29] B. Becker, R. Kohavi, Adult, UCI Machine Learning Repository, 1996, http://dx.doi.org/10.24432/C5XW20.

[30] H. Hofmann, Statlog (German Credit Data), UCI Machine Learning Repository, 1994, http://dx.doi.org/10.24432/C5NC77.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[32] F.J. Provost, R. Kohavi, Guest editors' introduction: On applied research in machine learning, Mach. Learn. 30 (2–3) (1998) 127–132.

[33] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 214–226.

[34] A. Chouldechova, Fair prediction with disparate impact: A study of bias in recidivism prediction instruments, Big Data 5 (2) (2017) 153–163.

[35] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Adv. Neural Inf. Process. Syst. 29 (2016).

[36] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth, Fairness in criminal justice risk assessments: The state of the art, Sociol. Methods Res. 50 (1) (2021) 3–44.

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

[38] L. Baresi, C. Criscuolo, C. Ghezzi, Understanding fairness requirements for ml-based software, in: 2023 IEEE 31st International Requirements Engineering Conference, RE, IEEE, 2023, pp. 341–346.

[39] R.K. Bellamy, et al., AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias, IBM J. Res. Dev. 63 (4/5) (2019) 4–41.

[40] S. Majumder, J. Chakraborty, G.R. Bai, K.T. Stolee, T. Menzies, Fair enough: Searching for sufficient measures of fairness, ACM Trans. Softw. Eng. Methodol. 32 (6) (2023) 1–22.

[41] C. Criscuolo, T. Dolci, M. Salnitri, et al., Mitigating unfairness in machine learning: A taxonomy and an evaluation pipeline, in: CEUR WORKSHOP PROCEEDINGS, 2024, pp. 217–226.

[42] F. Kamiran, T. Calders, Data preprocessing techniques for classification without discrimination, Knowl. Inf. Syst. 33 (1) (2011) 1–33, http://dx.doi.org/10.1007/S10115-011-0463-8.

[43] M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 259–268, http://dx.doi.org/10.1145/2783258.2783311.

[44] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: International Conference on Machine Learning, PMLR, 2013, pp. 325–333.

[45] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, K.R. Varshney, Optimized pre-processing for discrimination prevention, Adv. Neural Inf. Process. Syst. 30 (2017).

[46] H. Weerts, M. Dudík, R. Edgar, A. Jalali, R. Lutz, M. Madaio, Fairlearn: Assessing and improving fairness of ai systems, J. Mach. Learn. Res. 24 (257) (2023) 1–8.

[47] B.H. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 335–340.

[48] T. Kamishima, S. Akaho, H. Asoh, J. Sakuma, Fairness-aware classifier with prejudice remover regularizer, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23, Springer, 2012, pp. 35–50.

[49] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, H. Wallach, A reductions approach to fair classification, in: International Conference on Machine Learning, PMLR, 2018, pp. 60–69.

[50] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, K.Q. Weinberger, On fairness and calibration, Adv. Neural Inf. Process. Syst. 30 (2017).

[51] F. Kamiran, A. Karim, X. Zhang, Decision theory for discrimination-aware classification, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 924–929.

[52] WHO's Global Health Estimates, The top 10 causes of death, 2024, https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, Last accessed on January 10, 2025.

[53] L.P. Garrison Jr., P.J. Neumann, P. Erickson, D. Marshall, C.D. Mullins, Using real-world data for coverage and payment decisions: the ispor real-world data task force report, Value Heal. 10 (5) (2007) 326–335.

[54] N. Shahbazi, Y. Lin, A. Asudeh, H. Jagadish, Representation bias in data: A survey on identification and resolution techniques, ACM Comput. Surv. 55 (13s) (2023) 1–39.

[55] P. Indians, Diabetes dataset, 2022, URL https://www.kaggle.com/datasets/akshaydattatraykhare/diabetes-dataset.

[56] J.W. Smith, J.E. Everhart, W. Dickson, W.C. Knowler, R.S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Annual Symposium on Computer Application in Medical Care, American Medical Informatics Association, 1988, p. 261.

[57] J. Clore, K. Cios, J. DeShazo, B. Strack, Diabetes 130-US Hospitals for Years 1999–2008, UCI Machine Learning Repository, 2014, http://dx.doi.org/10.24432/C5230J.

[58] S.M. Ganie, P.K.D. Pramanik, M. Bashir Malik, S. Mallik, H. Qin, An ensemble learning approach for diabetes prediction using boosting techniques, Front. Genet. 14 (2023) 1252159.

[59] M. Zeinalnezhad, S. Shishehchi, An integrated data mining algorithms and meta-heuristic technique to predict the readmission risk of diabetic patients, Heal. Anal. 5 (2024) 100292.

[60] Fedesoriano, Stroke prediction dataset, 2020, URL https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

[61] G. Sailasya, G.L.A. Kumari, Analyzing the performance of stroke prediction using ML classification algorithms, Int. J. Adv. Comput. Sci. Appl. 12 (6) (2021).

[62] E. Dritsas, M. Trigka, Stroke risk prediction with machine learning techniques, Sensors 22 (13) (2022) 4670.

[63] A.H. Gucyetmez B, C-reactive protein and hemogram parameters for the non-sepsis systemic inflammatory response syndrome and sepsis: What do they mean? PLoS ONE 11 (2) (2016) E0148699, URL https://doi.org/10.1371/journal.pone.0148699.

[64] M. Mollura, D. Chicco, A. Paglialonga, R. Barbieri, Identifying prognostic factors for survival in intensive care unit patients with SIRS or sepsis by machine learning analysis on electronic health records, PLOS Digit. Heal. 3 (3) (2024) e0000459.

[65] S. Hammer, et al., AIDS Clinical Trials Group Study 175, UCI Machine Learning Repository, 1996, http://dx.doi.org/10.24432/C5ZG8F.

[66] S.M. Hammer, D.A. Katzenstein, M.D. Hughes, H. Gundacker, R.T. Schooley, R.H. Haubrich, W.K. Henry, M.M. Lederman, J.P. Phair, M. Niu, M.S. Hirsch, T.C. Merigan, A trial comparing nucleoside monotherapy with combination therapy in HIV-infected adults with CD4 cell counts from 200 to 500 per cubic millimeter. AIDS clinical trials group study 175 study team., New Engl. J. Med. 335 15 (1996) 1081–1090, URL https://api.semanticscholar.org/CorpusID:40754467.

[67] S. Golovenkin, et al., Myocardial Infarction Complications, UCI Machine Learning Repository, 2020, http://dx.doi.org/10.24432/C53P5M.

[68] S.E. Golovenkin, J. Bac, A. Chervov, E.M. Mirkes, Y.V. Orlova, E. Barillot, A.N. Gorban, A. Zinovyev, Trajectories, bifurcations, and pseudo-time in large clinical datasets: applications to myocardial infarction and diabetes data, GigaScience 9 (11) (2020) giaa128.

[69] R.E. Kharoua, Alzheimer's disease dataset, 2024, http://dx.doi.org/10.34740/KAGGLE/DSV/8668279, URL https://www.kaggle.com/dsv/8668279.

[70] X. Bouthillier, P. Delaunay, M. Bronzi, A. Trofimov, B. Nichyporuk, J. Szeto, N. Mohammadi Sepahvand, E. Raff, K. Madan, V. Voleti, et al., Accounting for variance in machine learning benchmarks, Proc. Mach. Learn. Syst. 3 (2021) 747–769.