# Auditing fairness under unawareness through counterfactual reasoning

Giandomenico Cornacchia [a,*], Vito Walter Anelli [a], Giovanni Maria Biancofiore [a], Fedelucio Narducci [a], Claudio Pomo [a], Azzurra Ragone [b], Eugenio Di Sciascio [a]

[a] *Polytechnic University of Bari, Via Orabona, 4, Bari, 70125, Italy*
[b] *Università degli Studi di Bari Aldo Moro, Piazza Umberto I, 1, Bari, 70121, Italy*

ARTICLE INFO

*Keywords:*
Fairness
Discrimination
Counterfactual reasoning
Proxy features

ABSTRACT

Artificial intelligence (AI) is rapidly becoming the pivotal solution to support critical judgments in many life-changing decisions. In fact, a biased AI tool can be particularly harmful since these systems can contribute to or demote people's well-being. Consequently, government regulations are introducing specific rules to prohibit the use of sensitive features (e.g., gender, race, religion) in the algorithm's decision-making process to avoid unfair outcomes. Unfortunately, such restrictions may not be sufficient to protect people from unfair decisions as algorithms can still behave in a discriminatory manner. Indeed, even when sensitive features are omitted (*fairness through unawareness*), they could be somehow related to other features, named proxy features. This study shows how to unveil whether a black-box model, complying with the regulations, is still biased or not. We propose an end-to-end bias detection approach exploiting a counterfactual reasoning module and an external classifier for sensitive features. In detail, the counterfactual analysis finds the minimum cost variations that grant a positive outcome, while the classifier detects non-linear patterns of non-sensitive features that proxy sensitive characteristics. The experimental evaluation reveals the proposed method's efficacy in detecting classifiers that learn from proxy features. We also scrutinize the impact of state-of-the-art debiasing algorithms in alleviating the proxy feature problem.

## 1. Introduction

The Cambridge Dictionary defines *discrimination* as the act of "*treating a person or particular group of people differently, especially in a worse way from the way in which you treat other people, because of their skin color, sex, sexuality, etc.*".[1] Recently, various regulations have been designed to face the discrimination problem. For instance, Article 21 of the EU Charter of Fundamental Rights defines the non-discrimination requirements: "*any discrimination based on any ground such as sex, race, color, ethnic or social origin, genetic features, language, religion or belief, political or any other opinion, membership of a national minority, property, birth, disability, age or sexual orientation shall be prohibited*".[2] In 2015, the United Nations General Assembly set up the Sustainable Development Goals (SDGs) or Global Goals, a collection of 17 interlinked global goals designed to be a "*blueprint for achieving a better and more sustainable future*

* Corresponding author.
*E-mail addresses:* giandomenico.cornacchia@poliba.it (G. Cornacchia), vitowalter.anelli@poliba.it (V.W. Anelli), giovannimaria.biancofiore@poliba.it (G.M. Biancofiore), fedelucio.narducci@poliba.it (F. Narducci), claudio.pomo@poliba.it (C. Pomo), azzurra.ragone@uniba.it (A. Ragone), eugenio.disciascio@poliba.it (E. Di Sciascio).

[1] https://dictionary.cambridge.org/dictionary/english/discrimination

[2] https://fra.europa.eu/en/eu-charter/article/21-non-discrimination

https://doi.org/10.1016/j.ipm.2022.103224

*for all*".[3] Most of the SDGs are somehow related to the discrimination problem, such as *no poverty*, *zero hunger*, *gender equality*, and *reduced inequality*. The discrimination problem is well-known and recognized in the financial domain where, for example, the decision to approve or deny credit has been regulated with precise and detailed regulatory compliance requirements (i.e., Equal Credit Opportunity Act,[4] Federal Fair Lending Act,[5] and Consumer Credit Directive for EU Community[6]). However, these laws were set to prevent discrimination in human decision-making processes and not in automated ones, such as those exploiting Machine Learning (ML) or, more generally, Artificial Intelligence (AI) systems. The EU Commission, in the wake of the GDPR[7] (i.e., a regulation to safeguard personal data), seeks to regulate the use of AI systems with the "Ethics Guidelines for Trustworthy AI" and more recently with "The Proposal for Harmonized Rule on AI". The regulated characteristics are various (e.g., technical robustness, privacy, data governance, transparency, accountability, societal and environmental well-being), and the European legislature deems adopting non-discriminatory AI models crucial. Therefore, the financial domain is the perfect workbench to test these regulations. Indeed, financial services are considered high-risk AI applications on the European AI risk scale (the levels are: minimal, limited, high, and unacceptable risk). As a consequence, a financial AI model must demonstrate fairness concerning sensitive characteristics to protect the social context in which it operates.

Since unfair treatment is strictly related to discriminatory behavior, fairness can be seen as the antonym of discrimination. Unfortunately, finding a strict and formal definition of fairness is challenging, and the subject is still under debate. Mehrabi, Morstatter, Saxena, Lerman, and Galstyan (2021) proposed a definition that could fit the financial domain and its discrimination-derived risks. They defined *fairness* as "*the absence of any prejudice or favoritism towards an individual or a group based on their inherent or acquired characteristics*". Another relevant aspect of fairness is highlighted by Ekstrand, Das, Burke, Diaz, et al. (2022) that refer to *unfairness* when a system treats people, or groups of people, in a way that is considered "unfair" by some moral, legal, or ethical standard. The exciting aspect is that, in that case, "fairness" is related to the normative aspects of the system and its effects. For this work, the *counterfactual fairness* as defined by Pitoura, Stefanidis, and Koutrika (2022) is particularly relevant. The intuition, in this case, is that an output is fair towards an entity if it is the same in both the actual world and a counterfactual world where the entity belongs to a different group. Causal inference is used to formalize this notion of fairness. This definition inspired the design of our model. From a geometrical perspective that considers how a decision model works, Dwork, Hardt, Pitassi, Reingold, and Zemel (2012) say that items that are close in construct space shall also be close in decision space, which is widely known as individual fairness: similar individuals should receive similar outcomes. In contrast to individual fairness, Deldjoo, Jannach, Bellogin, Difonzo, and Zanzonelli (2022) define group fairness that aims to ensure that "similar groups have similar experiences". Typical groups in such a context are a majority or dominant group and a protected group (e.g., an ethnic minority). Following this overview, some critical aspects of this work emerged: the legislation, the counterfact, and the group. More specifically, the legislation is the primary motivation behind this work, the counterfactual generation is the strategy we exploited for detecting unfairness, and the group is the subject of discrimination we want to catch. Although system designers train a model without any discriminatory purpose, several studies demonstrated that using AI systems without considering ethical aspects can promote discrimination (Bickel, Hammel, & O'Connell, 1975; Corbett-Davies, Pierson, Feller, Goel, & Huq, 2017; Dressel & Farid, 2018). Moreover, while the financial domain regulations strictly prohibit using sensitive characteristics for decision-making, some researchers defend their usage and believe the model should avoid unfair treatments (i.e., active bias detection) (Elliott, Fremont, Morrison, Pantoja, & Lurie, 2008; Ruf & Detyniecki, 2020). Nevertheless, only avoiding using sensitive features for training AI models does not guarantee the absence of biases in the outcome (Agarwal & Mishra, 2021). Indeed, there could be features in the dataset that can represent an implicit sensitive feature. In this study, we name these independent features as a *proxy features* for the sensitive one. For instance, education, smoking and drinking habits, pet ownership, and diet can be proxy variables for the feature age. The relationship between proxy and sensitive features generally depends on multicollinearity, namely a highly linear relationship between more than two variables. Unfortunately, non-linear relationships are more challenging to detect.

This investigation relies on the "*Fairness Under Unawareness*" –or "*blindness*" Pitoura et al. (2022)– definition (i.e., "*an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process*" (Chen, Kallus, Mao, Svacha, & Udell, 2019)). The choice of this definition is a logical consequence of current regulations. Indeed, like for other high-risk applications, the law dictates that AI applications in the financial domain cannot use sensitive information.

This work investigates a strategy to detect decision biases in a realistic scenario where sensitive features are absent, and there could be an unknown number of proxy features. We propose to tackle this challenging task by designing a system composed of three main modules. The first module encapsulates the classifier to analyze, named the **outcome classifier**. This predictor, as regulations suggest, is trained without any sensitive features. The second module trains a separate classifier, named **sensitive feature classifier**, on the same features to predict the sensitive characteristics. The third module calculates the minimal counterfactual samples, i.e., variants of the original sample, by modifying the values of non-sensitive features to obtain a different outcome with the outcome classifier. Finally, the sensitive feature predictor classifies the generated samples to check whether the samples do still belong to the original sensitive class. If this does not occur, the outcome predictor is biased, and its unfairness can be quantified.

To better explain the idea behind our approach, let us introduce a simple example regarding the loan granting process. Suppose our goal is to assess whether our loan classifier discriminates against women. In this example, the protected class is women, and

[3] United Nations (2017) Resolution adopted by the General Assembly on 6 July 2017

[4] https://www.ftc.gov/enforcement/statutes/equal-credit-opportunity-act

[5] https://www.fdic.gov/resources/supervision-and-examinations/consumer-compliance-examination-manual/documents/4/iv-1-1.pdf

[6] https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:32008L0048

[7] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

the sensitive feature is gender. The outcome classifier is a whatsoever state-of-the-art classification model trained without gender. The sensitive feature classifier will then distinguish men from women by exploiting the other non-sensitive features in the dataset (e.g., car type, job, education). An event triggers the system's operation: a woman uses the outcome classifier to obtain a loan, and her request is rejected. Therefore, the counterfactual module modifies the values of her non-sensitive attributes until the loan is approved (e.g., increasing income, reducing the loan duration). The sensitive-feature classifier then classifies the new approved counterfactual sample. Is she still classified as a woman by the system? What could we say if the features that approved the loan are the same that classified her as a man? The decision model may still be biased and thus unfair, and since it does not use sensitive features, this is due to proxy features.

Overall, this study proposes an approach for detecting bias in machine learning models using counterfactual reasoning, even when those models are trained without sensitive features, i.e., in the case of *Fairness Under Unawareness*. This setting could be summarized as outlined by Mehrabi et al. (2021): "*An algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process*". This research aims to investigate the presence of bias in an algorithm using counterfactual reasoning as an effective strategy for bias detection and evaluate if different counterfactual strategies have dissimilar efficacy in detecting biases. In detail, with this study, we intend to answer the following research questions:

- **RQ1:** Is there a principled way to identify if proxy features exist in a dataset?
- **RQ2:** Does the Fairness Under Unawareness setting ensure that decision biases are avoided?
- **RQ3:** Is counterfactual reasoning suitable for discovering decision biases?
- **RQ4:** Is our methodology effective for discovering discrimination and biases? Are there limitations in its application?

To provide an answer to the previous RQs, we performed an extensive experimental evaluation on three state-of-the-art datasets, broadly recognized as datasets containing Social Bias. The remainder of the paper is organized as follows: Section 2 provides an overview of the most relevant research in the fields of fairness and counterfactual reasoning, Section 3 provides the preliminaries of the work, while Section 4 describes the methodology. Section 5 introduces the experiments, while results are discussed in Section 6. Conclusion and future work are drawn in Section 8.

## 2. Related work

This study presents a strategy for detecting bias in machine learning models using Counterfactual Reasoning. This section aims to provide the reader with an adequate background, introducing the most relevant works in Fairness and Counterfactual Reasoning research fields.

### *2.1. Fairness, fairness under unawareness, and proxy features*

In machine learning research, fairness is a well-studied topic with a considerable body of knowledge to draw from Ashokan and Haas (2021), Pedreschi, Ruggieri, and Turini (2008) and Zhu, Hu, and Caverlee (2018). The first domains to take an interest in the theme were Financial Services, Banking, and Health. In fact, due to the critical impact of decision-making in these domains on people's well-being, today, the use of sensitive characteristics is strictly prohibited. The decisional tasks, i.e., regression and classification tasks with models deprived of sensitive features, took the name of *Fairness Under Unawareness* assessment. However, companies and institutions must demonstrate the fairness and impartiality of their systems despite the absence of such sensitive characteristics (Chen, 2018).

While designing the decision-making algorithm not to leverage sensitive information is simple, assuring the same accuracy as before and demonstrating that the predictor is unbiased is another matter. In fact, for tasks like granting credit cards or approving loans and mortgages, financial companies should collect and use sensitive features to ensure their tools are non-discriminatory. On this point, the EU Commission proposes a conformity assessment before AI systems are put into service or placed on the market.[8] In fact, their tools are subject to fair and trustworthy audit assessments to check their conformity. However, is a shallow check of the input characteristics sufficient to determine that a predictor will not suggest unfair treatment? Even though the user does not provide protected characteristics, the system could predict sensitive features from variables, i.e., proxy variables, that still represent protected characteristics. The models that infer sensitive features from proxy variables are known as "probabilistic proxy models (Bureau, 2014; Chen et al., 2019)".

Most of the approaches proposed in the literature for identifying proxy features rely on techniques capable of discovering multicollinearity between variables. If the correlation between two independent variables is $1$ or $-1$, we have perfect multicollinearity between them (Agarwal & Mishra, 2021). Methods for discovering multicollinearity are based on Linear Regression, Variance Inflation Factor, and Pearson correlation coefficient (Yeom, Datta, & Fredrikson, 2018). However, the relationships may not be linear. In that case, cosine similarity and mutual information are the most used approaches (Agarwal & Mishra, 2021). Elliott et al. (2009) investigated, in their work, whether from customer characteristics such as name and geolocation information (e.g., residence address) the information about the race can be inferred. Using a Bayesian classifier model, they demonstrated that first-name listings might improve prediction estimates. In particular, they showed that in some Asian and black subgroups, first names tend to have

[8] https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

low sensitivity. Conversely, imputing native American and multiracial identities from surname and residence remains challenging. Chen et al. (2019) studied the relationship between proxy features and sensitive variables (i.e., geolocation and race). In their work, bias seems to depend on the chosen threshold, suggesting an ad-hoc threshold estimation to produce fair thresholded classifiers and probabilistic proxy models.

Fabris, Esuli, Moreo, and Sebastiani (2021) use a quantification approach to measure group fairness when sensitive features are unknown. The advantage is that quantification-based estimates are robust to distribution shifts and do not allow the inference of sensitive attributes at the individual-class level. Biswas and Mukherjee (2021) likewise employ quantification techniques. In detail, they propose a mitigation model in which training and test population subgroups structurally differ. The proposed model, CAPE (i.e., Combinatorial Algorithm for Proportional Equality), aims to minimize a peculiar loss to obtain a Proportional-Equality-fair model.

The exposure of some groups on a geographic and demographic basis is also a problem that impacts the Recommender Systems community. In this direction, there are some attempts to analyze and mitigate this type of issue. One possible solution is the re-ranking strategy (Gómez, Boratto, & Salamó, 2022), to balance the items produced in a continent and the ranking of the items. Another recent proposal is FairLens (Panigutti, Perotti, Panisson, Bajardi, & Pedreschi, 2021), a framework to discover the bias of a generic Decision Support System model. The authors tested the approach in the medical domain. Interestingly, this strategy involves human experts in analyzing misclassifications. Specifically, the expert describes which aspects of the impacted patients' clinical history are responsible for the model error in the considered groups. It is essential to underline that the human expert, who thoroughly analyzes potential fairness issues, plays a crucial role in the operational loop.

### 2.2. *Counterfactual reasoning*

Counterfactual Reasoning is an active and flourishing field in artificial intelligence research (Ginsberg, 1986; Miller, 2019). This research was initially born to investigate causal links (Pearl, 1994), and today it can count on several contributions (Ferrario, 2001). Most of them define and employ counterfactuals as a helpful tools to explain the decisions taken by modern decision support systems. The underlying rationale is that some aspects of past events could predict future events. In detail, some studies focus on identifying causality-related aspects to discover the link between the counterfactuals and the analyzed phenomenon (DeMartino, 2020).

Counterfactual Reasoning finds application in various fields. To summarize what we have briefly detailed before, machine learning research has positively valued these contributions ranging from Explainable AI (Mothilal, Sharma, & Tan, 2020) to the most recent counterfactual fairness measures (Joo & Kärkkäinen, 2020; Kusner, Loftus, Russell, & Silva, 2017). Beyond the theoretical aspects, Counterfactual Reasoning is extensively applied to interactive systems (Bottou et al., 2013; Cornacchia, Narducci, & Ragone, 2021a; Swaminathan & Joachims, 2015; Tavakol, 2020). Unfortunately, this important application showed some limitations. These systems employ machine learning models that reflect the data they use for learning. Consequently, the same information influences the reasoning, and the contribution of Counterfactual Reasoning could be limited or somehow biased. The explaining policy, coming from Counterfactual Reasoning, exhibits a bias towards the implemented learning model. Researchers devoted considerable effort to tackle this issue and proposed new models such as doubly robust estimators (Dudík, Langford, & Li, 2011).

Overall, even though limitations that need a solution, Counterfactual Reasoning is taking over Explainable AI, and it is becoming the de facto standard for explaining decisions taken by autonomous systems. In this respect, the European Union's "right to explanation" played a crucial role in arousing a further interest in this methodologies (Korikov, Shleyfman, & Beck, 2021). Indeed, they are compliant with the regulation and easily interpreted by either a domain expert or a layperson (Sokol & Flach, 2019).

Decision support systems particularly benefited from these models. However, the more the application domain is vital, the more the fairness problem emerges. For instance, the issue cannot be overlooked in sensitive domains such as justice, risk assessment, or clinical risk prediction. This need promoted the most promising research in the Counterfactual Reasoning field to analyze and mitigate this issue. Kusner et al. (2017) proposed a metric exploiting causal inference to assess fairness at an individual level by requiring that a sensitive attribute not be the cause of a change in a prediction. Even though the proposed solution and the involved methodologies differ from ours, the studies take the first steps from the same motivations. Pfohl, Duan, Ding, and Shah (2019) further extended the metric for clinical risk assessment. They aim to mitigate the exposure of medical care disparities due to bias implicitly embedded in data for historically underrepresented and mistreated groups. For what concerns the risk assessment domain, Mishler, Kennedy, and Chouldechova (2021) put forward a similar working hypothesis. They propose a counterfactual equalized odds ratio criterion to train predictors operating in the post-processing phase. They extend and adapt previous post-processing approaches (Hardt, Price, & Srebro, 2016) to the counterfactual setting and employ doubly robust estimators.

In contrast to the majority of the mentioned studies, our investigation aims to leverage a counterfactual generation tool to reveal the presence of implicit biases in a decision support system. Interestingly, this motivation is similar to Bottou et al. (2013). In fact, both aim to answer the question: "How would the system have decided if we had replaced some user characteristics?". Beyond this commonality, the two studies differ significantly. Indeed, they focus on measuring the fidelity level of the system and robustifying the model. Instead, our study is in line with the goal of other investigations (Denton, Hutchinson, Mitchell, Gebru, & Zaldivar, 2019; Mikolajczyk, Grochowski, & Kwasigroch, 2021) that aim to use the counterfactual approach to uncover the bias present in the dataset that plagues the predictive model itself.

**Table 1**
List of the main notational conventions used in this document.

| Notation | Description |
|---|---|
| $\mathbf{x}$ | A vector of values for non-sensitive features $\mathbf{x} =< x_1, x_2, \dots, x_n >$. |
| $\mathbf{s}$ | A vector of binary values for sensitive features $\mathbf{s} =< s_1, s_2, \dots, s_l >$. When no confusion arises, $s$ is reported instead of $s_i$ |
| $y$ | A binary class value from the target domain for a single data point, with $y \in \{0, 1\}$ |
| $\mathbf{p}$ | A vector of values for proxy features, i.e., a subvector of $\mathbf{x}$, with $h(\cdot)$ being an unknown function s.t. $h(\mathbf{p}) = s_i$ |
| $\hat{y}$ | A binary class prediction value from the target domain for a single data point, with $\hat{y} \in \{0, 1\}$ |
| $\hat{s}_i$ | A binary prediction value of the $i$th sensitive feature, with $\hat{s}_i \in \{0, 1\}$ |
| $f(\mathbf{x}) = \hat{y}$ | A binary classification function of the target variable $y$ |
| $f_s(\mathbf{x}) = \hat{s}_i$ | A binary classification function of the sensitive variable $s_i$ |
| $g(\mathbf{x}) = C_{\mathbf{x}}$ | A function that, given a data point $\mathbf{x}$, returns $k$ counterfacts. |
| $\mathbf{c}_{\mathbf{x}} \in C_{\mathbf{x}}$ | A counterfact of $\mathbf{x}$. $\mathbf{c}_{\mathbf{x}}$ is a vector $\mathbf{c}_{\mathbf{x}} =< c_{x_1}, c_{x_2}, \dots, c_{x_n} >= \mathbf{x} \pm \epsilon$, with $\epsilon$ being a perturbation such that $f(\mathbf{c}_{\mathbf{x}}) = 1 - f(\mathbf{x}) = 1 - \hat{y}$. |

### 2.3. *Social, theoretical, and practical implications on information access systems*

The UN Agenda 2030 for Sustainable Development sets out 17 Sustainable Development Goals, which are part of a broader program of actions consisting of 169 associated targets to be achieved in the environmental, economic, social, and institutional domains by 2030. Among them, there are 'gender equality', 'reducing inequalities', and 'responsible consumption and production' –i.e., goals 5, 10, and 12, respectively. As a consequence, current and impending regulations affecting high social impact tasks will comply with the UN Agenda 2030. Among the others, the financial sector is a high-risk domain, as unethical use of AI can have significant repercussions from a social point of view, such as for instance discriminatory access to credit. Several works attempted to tackle the fairness problem or provide model explainability for tasks ranging from classification to loan recommendation (Chen, 2018; Cornacchia et al., 2021a; Cornacchia, Narducci, & Ragone, 2021b; Das et al., 2021). The "Fairness Under Unawareness" setting (i.e., "an algorithm is fair as long as any protected attributes are not explicitly used in the decision-making process") mitigated the discrimination. However, the evaluation and the quantification of bias in a situation of "Fairness Under Unawareness" are of worryingly little interest to researchers. The investigation at hand proposes a theoretical approach to identify the existence of bias even when sensitive information is not exploited in the training of the machine learning model. The proposed approach is general enough to neglect what kind of classifier is adopted under the hood and could be used in any classification task. The whole approach could be practically very useful for any practitioner since it could be used as a black box that measures and returns several pieces of information regarding the potential bias. Finally, the approach is designed to be a support tool for several kinds of Information Access Systems. The prominent potential application of the proposed approach is in Conversational-Agent systems that rely on lending recommendations (e.g., peer-to-peer lending) in which social bias may imply different access to credit. In that setting, the proposed system sheds light not only on the features that are necessary to reverse the decision but also on the potential biases of the decision maker. More generally, every Information Access System exploiting machine learning models that imply life-changing decisions can use our methodology to assess the bias in the models.

### 2.4. *Contextualizing our work*

To the best of our knowledge, and quite unexpectedly, the idea of learning a classifier on sensitive features for discovering biases is unexplored in the financial domain literature. Furthermore, given the regulator's intervention, the concept of *fairness under unawareness* has assumed a crucial role in financial decision-making systems. However, the research on detecting bias for models trained in a fairness-under-unawareness setting is still in a very early stage. The experimental setup adopted in this investigation rigorously follows the best practice proposed in the recent literature and complies with the regulations. Nevertheless, the study shows that removing sensitive features from a decision support system does not guarantee a fair outcome. Concerning existing state-of-the-art approaches, the analysis tackles the fairness theme in the financial domain and proposes a general approach to identify implicit bias in a decision support system. Finally, instead of leveraging Counterfactual Reasoning to explain outcomes, the approach exploits the causal link between the counter-facts and the prediction to reveal the otherwise unnoticed bias.

## 3. Preliminaries

This section introduces some useful notation that is extensively used in the rest of the paper. To ease the reading and for a rapid understanding, the definition of protected groups has some commonalities with Chen et al. (2019), while some other aspects necessarily diverge from it due to the different nature of the study. The notation used is further condensed in Table 1, while in Table 2 we can find the list of acronyms used in the work.

In the following, we will refer to a set $\mathcal{D}$, with $|\mathcal{D}| = m$, of data points whose domain $dom(\mathcal{D})$ is composed by a number $n$ of non-sensitive features and a number $l$ of sensitive ones, i.e., $|dom(\mathcal{D})| = n + l$. Given a data point $d \in \mathcal{D}$, we can represent it as the concatenation of a vector $\mathbf{x}$ containing values of non-sensitive features and a vector $\mathbf{s}$ containing values for sensitive features.

**Non-sensitive Features**: We use $\mathbf{x} = \langle x_1, x_2, \dots, x_n \rangle$ to represent a vector of values for non-sensitive features in $dom(\mathcal{D})$. The value of $x_i$, with $1 \leq i \leq n$, can be categorical (set of discrete values) or numerical (set of continuous values).

**Table 2**
List of the most frequent and useful acronyms used in this document arranged in alphabetic order.

| Context | Acronym | Meaning |
|---|---|---|
| Behavior | AI | Artificial Intelligence |
| | CF | Counterfactual |
| | FAccT | Fairness, Accountability and Transparency |
| | FUU | Fairness Under Unawareness |
| | ML | Machine Learning |
| | SDG | Sustainable Development Goals |
| | XAI | eXplainable Artificial Intelligence |
| Model | AdvDeb | Adversarial Debiasing |
| | CAPE | Combinatorial Algorithm for Proportional Equality |
| | DiCE | Diverse Counterfactual Explanation |
| | DNN | Deep Neural Network |
| | LFERM | Linear Fair Empirical Risk Minimization |
| | LGBM | Light Gradient Boosting Machine |
| | LR | Logistic Regression |
| | SVM | Support-Vector Machines |
| | XGB | eXtreme Gradient Boosting |
| Metric | DAO | Difference in Average Odds |
| | DEO | Difference in Equal opportunity |
| | DI | Disparate Impact |
| | DSP | Difference in Statistical Parity |
| | AUC | Area Under the Receiver Operative Curve |

**Sensitive Features**: We use $\mathbf{s} = \langle s_1, s_2, \dots, s_l \rangle$ to represent a vector of values for sensitive features in $dom(\mathcal{D})$. When no confusion arises, $s$ is reported instead of $s_i$. Without loss of generality, we assume the value of $s_i$, with $1 \leq i \leq l$, as binary, i.e., $s_i \in \{0, 1\}$. Based on the value of $s_i$, the advantaged group is referred to as *privileged* and associated with $s_i = 1$, the disadvantaged group is referred to as *unprivileged* and associated with $s_i = 0$.

**Target Labels**: Given a target feature $y \in \{0, 1\}$, we use $y^*$ to represent the positive outcome $y = 1$ (the negative outcome is associated to $y = 0$).

**Proxy Features**: Let $\mathbf{p} \subseteq \mathbf{x}$ be a subset of $\mathbf{x}$, and $h(\cdot)$ be a such that $h(\mathbf{p}) = s_i$, i.e., the value returned by $h$ applied to the values associated to the features in $\mathbf{p}$ is equal to the values associated to $s_i$. We say that $\mathbf{p}$ is a set of proxy features for the sensitive feature $s_i$.

In practical terms, if we knew $h(\cdot)$, a set of proxy features could be used to predict a certain sensitive feature.

**Outcome prediction**: Let $\hat{y} \in \{0, 1\}$ be the prediction for a given data point. The notation $\hat{y} = 1$ denotes a *favorable* prediction (e.g., loan application approved), while $\hat{y} = 0$ an *unfavorable* one (e.g., loan application rejected). Let $f(\cdot)$ be a function such that $f(\mathbf{x}) = \hat{y}$.

**Sensitive Feature Prediction**: Let $\hat{s}_i \in \{0, 1\}$ be the prediction of the $i$th sensitive feature. The notation $\hat{s}_i = 1$ denotes the prediction to belong to a *privileged* group, while $\hat{s}_i = 0$ denotes the prediction to belong to an *unprivileged* group.

Let $f_s(\cdot)$ be a function able to predict the value of a sensitive feature given the value of non sensitive ones, i.e., $f_s(\mathbf{x}) = \hat{s}_i$. Since the set of proxy features $\mathbf{p}$ is unknown, we can use $f_s(\cdot)$ to predict the value of $s_i$.

**Counterfactual samples**: Given a vector $\mathbf{x}$ and a perturbation $\epsilon$, we say that a vector $\mathbf{c_x} = \langle c_{x_1}, c_{x_2}, \dots, c_{x_n} \rangle = \mathbf{x} \pm \epsilon$ is a counterfactual of $\mathbf{x}$ if $f(\mathbf{c_x}) = 1 - f(\mathbf{x}) = 1 - \hat{y}$. We use the set $C_\mathbf{x}$, with $|C_\mathbf{x}| = k$, to denote the set of possible counterfactuals for $\mathbf{x}$. A function $g(\mathbf{x})$ is used to compute $k$ counterfactuals for $\mathbf{x}$.

Our investigation focuses on unfavorable outcome predictions. Consequently, all the generated counterfactuals are associated with a favorable $f(\mathbf{c_x}) = 1$. When no confusion arises, $\mathbf{c}$ and $C$ are reported instead of $\mathbf{c_x}$ and $C_\mathbf{x}$, respectively.

## 4. Methodology

The *fairness under unawareness* setting (see Section 2.1) poses several challenges to the identification of discriminatory behaviors performed by intelligent systems. On the one hand, the prohibition of exploiting sensitive features makes it extremely difficult to guarantee fair treatment for the various categories of users. On the other hand, proxy features can be non-linearly correlated with sensitive ones, thus making the commonly used statistical approaches useless. This section aims to define a model to identify discriminatory behaviors put in place by applications that make a decision that impacts, in some way, users' lives.

Fig. 1 depicts the principal components of our model, namely the *decision-maker*, the *counter-fact generator*, and the *sensitive-feature* classifier. As a relevant case study, the model has been specialized in the financial domain, considering the tasks of predicting loan-repayment default and individual income. However, its generality remains.

### 4.1. Decision-maker

The *decision-maker* is the key component of the decision support system. Even though the nature of the decisions can be heterogeneous, the decision-maker implements a machine-learning algorithm trained using past human decisions. Although it does

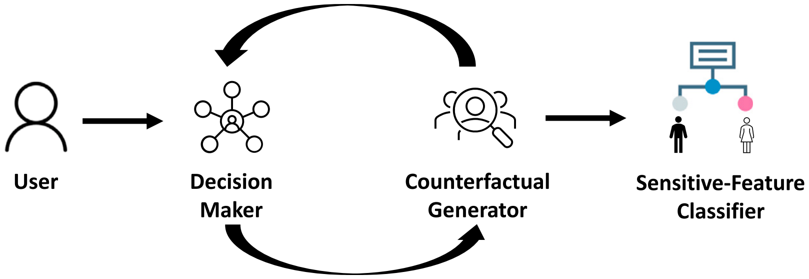

**Fig. 1.** The process of bias identification. The user's characteristics feed the target decision-maker. If the classifier returns a negative outcome, the counterfactual generator creates counterfactuals (CFs) to unveil how the user could achieve a positive outcome. A potential discriminatory behavior is identified if the CFs are classified into another demographic group.

not use sensitive features in the learning phase, we assume the predictive model is not necessarily bias-free, thanks to current regulations. This phenomenon could be due to proxy features.

To keep the approach as general as possible, to implement the *decision-maker*, we have chosen four largely adopted approaches to tackle the classification task. As far as possible, we avoided domain-specific models, preprocessing steps, and operations, and we relied on the general best practices that apply to a broader set of machine learning domains. Our choice was to sacrifice a small quantity of accuracy (even though the performance remains highly competitive) to gain the generality of the approach. In detail, we opted for Logistic Regression (LR), Support-Vector Machines (SVM), XGBOOST (XGB),[9] and LightGBM (LGBM).[10] LR is a linear statistical model that predicts the probability of one event taking place through a linear combination of independent variables. SVM is a pattern classification technique aiming to minimize an upper bound of the generalization error by maximizing the margin between the separating hyperplane and data instances (Boser, Guyon, & Vapnik, 1992). We exploited LR and SVM's Scikit-learn[11] implementation. XGB stands for eXtreme Gradient Boosting, and it implements gradient boosting machines guaranteeing high computational speed and performance. XGB learns both classification and regression models employing gradient-boosted decision trees. LGBM stands for Light Gradient Boosting Machine and uses an approach similar to XGB, thus favoring speed to robustness. Since the two approaches are state-of-the-art solutions yielding the best results in many competitions, we considered them despite their similarity.

### 4.1.1. *Debiased decision-makers*

To evaluate whether debiasing algorithms can reduce discriminatory behavior even in a "Fairness under unawareness" setting, we also considered *decision-makers* that exploit debiasing approaches. The overall system is the same as the one depicted in Fig. 1. This variation aims to assess whether debiasing models guarantee fair behavior and counterfactual reasoning can help discover discrimination even when these models are chosen as decision-makers. The debiasing algorithms we chose to investigate are *Adverarial Debiasing* (Zhang, Lemoine, & Mitchell, 2018) and *Linear Fair Empirical Risk Minimization* (Donini, Oneto, Ben-David, Shawe-Taylor, & Pontil, 2018).

*Adversarial debiasing.* Zhang et al. (2018) propose an adversary framework for debiasing algorithms (AdvDeb). The model comprises two elements: a target predictor and an adversary. The target label predictor consists of a Deep Neural Network that, given a general input $\mathbf{x}$, tries to predict the target label $y$. The adversary is a simple Neural Network that, fed by the predicted output of the DNN $\hat{y}$, tries to predict the sensitive label $s$. The DNN and the Adversary Network (AN) are trained to optimize both their model weights, $W$ (for DNN) and $U$ (for AN), by minimizing the losses $L_P(\hat{y},y)$ and $L_A(\hat{s},s)$, respectively. $L_P(\hat{y},y)$ is the target discrimination loss of the classification task, typically a CrossEntropy loss. $L_A(\hat{s},s)$ is the loss the adversary aims to maximize to predict the sensitive label. To ease the understanding of the adversarial learning process, $L_A(\hat{s},s)$ is herein used with an opposite sign with respect to the original paper, in which the adversary aims to minimize $L_A(\hat{s},s)$.

$$\underbrace{\min_W L_P(\hat{y},y) - \underbrace{\left[\max_U proj_{\nabla_W L_A(\hat{s},s)} L_P(\hat{y},y) + \alpha L_A(\hat{s},s)\right]}_{\text{best-case loss } L_A = \textbf{optimal prediction of the sensitive feature}}}_{\textbf{robust classification} \text{ against the prediction of the sensitive feature}} \tag{1}$$

The overall learning process resembles a min–max game in which the discriminator tries to minimize the loss of the predictor while the adversary tries to maximize its utility (see Eq. (1)). The middle term (i.e., $proj_{\nabla_W L_A(\hat{s},s)}$) limits the predictor from moving in a direction that promotes the adversary's loss reduction. For reproducibility, we adopt the IBM implementation available in the AIF360[12] framework.

[9] https://github.com/dmlc/xgboost

[10] https://github.com/microsoft/LightGBM

[11] https://scikit-learn.org/

[12] https://github.com/Trusted-AI/AIF360

*Linear fair empirical risk minimization.* Donini et al. (2018) propose a method that applies a fairness constraint to the loss function of an SVM classifier. In detail, they constrain the Hinge-loss to respect the "Equality of Opportunity" condition. The underlying goal is to remove the discrepancy between the false-negative rates of the privileged and unprivileged groups. The fairness condition is implemented by imposing an orthogonality constraint directly on the sample. Specifically, the sample vector is required to be orthogonal to the vector formed by the difference between the barycenters of the positive input samples in the two groups. Let $\mathbf{u} = u_{priv} - u_{unpriv}$ be the difference between the two barycenter vectors of the privileged and unprivileged groups, respectively, and let $|u_i|$ be the maximum valued feature in the vector, and $x$ be a sample in the original space. The fairness-constrained representation $\tilde{x}$ is then calculated as follows:

$$\tilde{x}_j = x_j - x_i \frac{u_j}{u_i}, \quad j \in \{1, \dots, i-1, i+1, \dots, d\} \tag{2}$$

with $d$ being the number of features. In this study, to ensure the reproducibility of the results, the implementation provided by the authors[13] is used. Specifically, the reader can refer to the linear implementation of Fair SVM, named *linear fair empirical risk minimization* (LFERM) therein.

### 4.2. Counterfactual generator

This study leverages the counterfactual reasoning approach to explore the decision-maker boundary in the feature space. Thanks to the sample generation process, this strategy can ease the analysis of the decision boundary even though the decision-maker is a black-box model. Moreover, the proposed model is utterly agnostic about the algorithm chosen as the decision-maker. The input of the counterfactual generator is the same sample previously evaluated by the decision-maker. When the system takes a decision adverse to the user (e.g., loan request rejected, income under a given threshold), the counterfactual generator is called in, and it produces new samples that would lead to a favorable outcome, as we discussed in Section 3. Under the hood, it modifies user characteristics following various strategies (e.g., increasing savings or changing education level). Each generated counterfactual feeds the decision-maker, and all the counterfactuals that switch the decision outcome, e.g., granting the loan, constitute the input of the next module of the system. For the sake of reproducibility and reliability, the counterfactuals are generated with an external counterfactual framework. We opted for DiCE (Mothilal et al., 2020), an open-source framework developed by Microsoft.[14] Mothilal et al. (2020) built their framework to satisfy two fundamental requirements. The generated counterfactuals should be (1) plausible and associated with actions that could be actionable by users and (2) diverse from each other. Both requirements fit the goals of our work. The first ensures that generated counterfactuals are close to the original sample and thus realistic. The second guarantees that they are all different, thus suggesting various strategies to solve the problem. The diversity requirement is fulfilled thanks to determinantal point processes (DPP), commonly used in selection problems with diversity constraints (Kulesza & Taskar, 2012).

For the sake of completeness, we briefly introduce the DiCE counterfactual generation process using the notation adopted in this study. Let $\mathbf{x}$ be a candidate sample, $C_{\mathbf{x}} = \{\mathbf{c}_{\mathbf{x}}^1, \mathbf{c}_{\mathbf{x}}^2, \dots, \mathbf{c}_{\mathbf{x}}^k\}$ be a set of $k$ candidate counterfactual samples, with $k$ being the desired number of counterfactuals, and $f(\cdot)$ being a predictor function, i.e., a machine learning model. The optimization function, the module generates counterfactual samples on, is then the following:

$$g(\mathbf{x}) = \underset{\mathbf{c}_{\mathbf{x}}^1, \dots, \mathbf{c}_{\mathbf{x}}^k}{\arg\min} \frac{1}{k} \sum_{i=1}^{k} yloss(f(\mathbf{c}_{\mathbf{x}}^i), y^*) + \frac{\lambda_1}{k} \sum_{i=1}^{k} dist(\mathbf{c}_{\mathbf{x}}^i, \mathbf{x}) - \lambda_2 dppd(\mathbf{c}_{\mathbf{x}}^1, \dots, \mathbf{c}_{\mathbf{x}}^k) \tag{3}$$

where $yloss(\cdot)$ is a metric (e.g., $\ell_1$-loss, $\ell_2$-loss, or hinge-loss) minimizing the distance between the predicted output of $\mathbf{c}_{\mathbf{x}}^i$ and the desired $y^*$; $dist$ is a proximity function that quantifies the distance between $\mathbf{c}_{\mathbf{x}}^i$ and $\mathbf{x}$; $dppd(\cdot)$ is the *determinantal point processes diversity*, i.e., the determinant of the kernel matrix of the inverse distance between counterfactuals. More formally:

$$dppd = det(\mathbf{K}), \quad \text{with} \quad \mathbf{K_{i,j}} = \frac{1}{1 + dist(\mathbf{c}_{\mathbf{x}}^i, \mathbf{c}_{\mathbf{x}}^j)} \tag{4}$$

where $dist$ in the previous equation denotes a generic distance metric between counterfactuals. Finally, $\lambda_1$ and $\lambda_2$ are hyperparameters that balance the contribution of the distance and the diversity part, respectively.

DiCE offers several strategies for generating candidate counterfactual samples. We decided to use three different approaches: Random, Genetic, and KDtree generation. The choice of these strategies allows (i) to assess whether it is possible to generate a large enough number of counterfactuals from a sample; (ii) to investigate which strategy is most effective for our purposes, and (iii) to find the most robust and valid method in generating plausible counterfactuals. The Random strategy randomly selects a set of features to perturb and replace the original sample. The perturbation goes ahead until the counterfactual satisfies the requirement $f(\mathbf{c}_{\mathbf{x}}) = y^*$. The KDtree strategy computes a tree-based distance between all the dataset samples; it chooses the samples that are close to the original one and that switch the outcome prediction to $y^*$. The Genetic strategy can start with a Random initialization or a KDtree initialization and then iterates by generating new samples close to the original one that switches the outcome prediction to $y^*$.

[13] https://github.com/jmikko/fair_ERM

[14] https://github.com/interpretml/DiCE

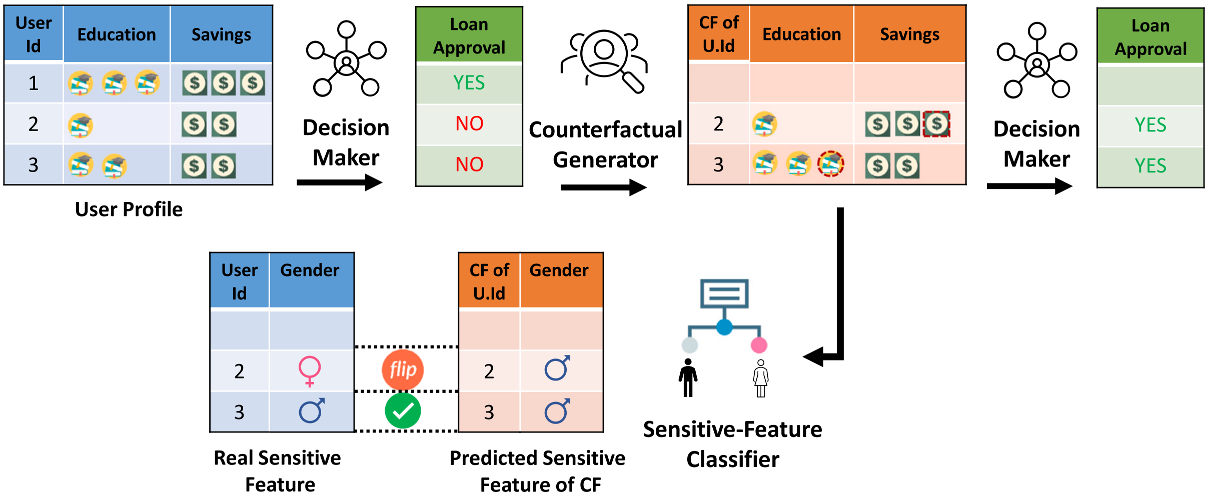

**Fig. 2.** An example of a loan-approval decision is analyzed through the model. Users' features are taken as inputs of the decision-maker module that decides whether the loan request is approved or not. The counterfactual generator creates new samples for all those with a negative outcome that would lead to approval. The Sensitive-Feature classification module predicts the counterfactuals' gender. If the predicted gender differs from the gender of the original sample, a warning is raised.

### 4.3. Sensitive-feature classifier

The *sensitive-feature classifier* performs a classification of the sample generated by the *counterfactual generator* (that caused a decision flip) into one of the sensitive categories. This component plays a crucial role in our methodology since it allows the system to discover hidden discriminatory models. For each sensitive feature (e.g., gender, race, etc.), a classifier is thus learned. In Fig. 1, the counterfactual sample that caused the flip becomes the input of the sensitive-feature classifier. If the sensitive-feature classification predicts a category different from the one initially (i.e., before generating counterfactuals) associated with the sample (e.g., from female to male), a bias in the decision-making process could occur. In fact, a change in the sensitive-feature classification means that there are some non-sensitive features (whose values have been changed by the counterfactual generator) that allow the system to recognize the counterfactual sample as belonging to the *privileged* class (i.e., male). Hence, the *sensitive-feature classifier* gives us an indication of the existence of a function that links non-sensitive features to sensitive ones, namely a proxy feature.

### 4.4. The model at work

Fig. 2 exemplifies the operation of the model. In the depicted example, the system provides a preliminary decision to grant a loan or not. The user profile is preprocessed and feeds the decision-maker. It determines whether to approve the user request or not by analyzing her characteristics. Suppose the request is rejected (in the example, this is the case for users #2 and #3). The counterfactual generator begins to craft counterfactuals by modifying the user characteristics. Concerning user 2, the counterfactual generator reduces the education level and increases the capital gain. For user 3, it increases the education level and reduces the capital gain. User 1 is not involved in the counterfactual generation step since its request has been accepted. The counterfactual samples for users 2 and 3 feed the decision-maker. When the decision-maker returns a different outcome (i.e., we have a decision flip from rejected to accepted), those samples are analyzed and classified by the sensitive-feature classifier. For simplicity, the example reported only one classifier for all the sensitive characteristics. Behind the curtains, each sensitive feature has a dedicated classifier. The classifier predicts the sensitive feature that could be different from the user's actual sensitive feature. If a mismatch occurs, it raises a warning since the counterfactual sample that received the loan approval is identified as belonging to another class (e.g., another gender). The following section introduces a metric – "Counterfactual Flips" – to assess how often the counterfactuals generated by a decision model represent individuals of another sensitive class. The metric intuitively gives an intuition of the potentially discriminatory behavior of the model.

**Definition 4.1** (*Counterfactual Flips*)**.** Let $\mathbf{x}$ be a sample belonging to a demographic group associated with the sensitive feature value $s$ whose model output is denoted as $f(\mathbf{x})$. Suppose the counterfactual generator produced a set $C_{\mathbf{x}}$ of $k$ counterfactuals with desired $y^*$ outcome $f(\mathbf{c}_{\mathbf{x}}^i) = y^* \quad \forall \mathbf{c}_{\mathbf{x}}^i \in C_{\mathbf{x}}$, with $i \in \{1 \dots k\}$. The Counterfactual Flips indicate the percentage of counterfactual samples belonging to another demographic group (i.e., $f_s(\mathbf{c}_{\mathbf{x}}^i) \neq f_s(\mathbf{x})$, with $f_s(\mathbf{x}) = s$).

$$\mathrm{CFlips}(\mathbf{x}, C_{\mathbf{x}}, f_s(\cdot)) = \frac{\sum_{i=1}^{k} (\mathbb{1}(\mathbf{c}_{\mathbf{x}}^i))}{k} \quad \text{where } \mathbb{1}(\mathbf{c}_{\mathbf{x}}^i) = \begin{cases} 1 & \text{if } f_s(\mathbf{c}_{\mathbf{x}}^i) \neq f_s(\mathbf{x}) \\ 0 & \text{if } f_s(\mathbf{c}_{\mathbf{x}}^i) = f_s(\mathbf{x}) \end{cases} \text{ with } \mathbf{c}_{\mathbf{x}} \in C_{\mathbf{x}} \tag{5}$$

**Algorithm 1:** Algorithm for model training and counterfactual generation

```
Input:
  • the Train and Test datasets $D_{train}$ and $D_{test}$, where $D_{train} = \{\mathcal{X}_{train}, \mathcal{Y}_{train}, \mathcal{S}_{train}\}$, and $D_{test} = \{\mathcal{X}_{test}, \mathcal{Y}_{test}, \mathcal{S}_{test}\}$,
  • the target label Classifier = $f(\cdot)$,
  • the sensitive label Classifier = $f_s(\cdot)$,
  • the classification loss $\text{Loss}(\cdot)$
  • the number of train epochs $Epochs$,
  • the number of counterfactuals to be generated for each sample $N_{CF}$,
  • the counterfactual generator $g(\cdot)$.
Result:
  • the set $\mathcal{A}$ composed of tuples of objects related to the samples associated with a **negative outcome** (i.e., $f(\mathbf{x}) = 0$),
    belonging to an **privileged group** (i.e., $s = 1$), and correctly predicted to belong to the same sensitive class
    (i.e., $f_s(\mathbf{x}) = 1$),
  • the set $\mathcal{B}$ composed of tuples of objects related to the samples associated with a **negative outcome** (i.e., $f(\mathbf{x}) = 0$),
    belonging to an **unprivileged group** (i.e., $s = 0$), and correctly predicted to belong to the same sensitive class
    (i.e., $f_s(\mathbf{x}) = 0$).
Randomly initialize $\theta_1$ for target output classifier $f(\cdot)$, and $\theta_2$ for sensitive label classifier $f_s(\cdot)$;
for $epoch \leftarrow 1$ to $Epochs$ do
    $\mathcal{X}_{train}, \mathcal{Y}_{train}, \mathcal{S}_{train} \leftarrow D_{train}$;
    $\hat{\mathcal{Y}}_{train} \leftarrow f(\mathcal{X}_{train})$;
    $\hat{\mathcal{S}}_{train} \leftarrow f_s(\mathcal{X}_{train})$;
    $\theta_1 \leftarrow \arg\min_{\theta_1} Loss(\hat{\mathcal{Y}}_{train}, \mathcal{Y}_{train})$;
    $\theta_2 \leftarrow \arg\min_{\theta_2} Loss(\hat{\mathcal{S}}_{train}, \mathcal{S}_{train})$;
endfor
for $d^{(i)} \in D_{test}$ do
    $\mathbf{x}^{(i)}, y^{(i)}, s^{(i)} \leftarrow d^{(i)}$;
    $\hat{y}^{(i)} \leftarrow f(\mathbf{x}^{(i)})$;
    $\hat{s}^{(i)} \leftarrow f_s(\mathbf{x}^{(i)})$;
    $\mathcal{C}_{\mathbf{x}^{(i)}} \leftarrow g(\mathbf{x}^{(i)})$ with $f(\mathbf{c}_{\mathbf{x}}) = y^*$;
    $\hat{\mathbf{s}}_{CF} \leftarrow f_s(\mathbf{c}_{\mathbf{x}}^{(i)})$ for $\mathbf{c}_{\mathbf{x}}^{(i)} \in \mathcal{C}_{\mathbf{x}^{(i)}}$;
    if $\hat{y}^{(i)} = 0$ then
        if $\hat{s}^{(i)}$=1 $\wedge$ $s^{(i)}$=1 then
            $\mathcal{A} \cup \{\langle \mathbf{x}^{(i)}, \mathcal{C}_{\mathbf{x}^{(i)}}, \hat{y}^{(i)}, \hat{s}^{(i)}, \hat{\mathbf{s}}_{CF} \rangle\}$;
        end
        if $\hat{s}^{(i)}$=0 $\wedge$ $s^{(i)}$=0 then
            $\mathcal{B} \cup \{\langle \mathbf{x}^{(i)}, \mathcal{C}_{\mathbf{x}^{(i)}}, \hat{y}^{(i)}, \hat{s}^{(i)}, \hat{\mathbf{s}}_{CF} \rangle\}$;
        end
    end
endfor
```

The higher the CFlips value is, the more severe the discriminations between privileged and unprivileged groups are. To evaluate the fairness of the models, we propose a Counterfactual approach to produce counterfactuals for each sample. For reproducibility reasons, the framework adopted for the generation of the Counterfactual samples is DiCE (see Section 4.2) with three different strategies: Random, Genetic, and KDtree generation. We first train the models for the target label binary classification task $f(\cdot)$, i.e., the decision-maker. Analogously, we train the models for the sensitive classification task $f_s(\cdot)$, i.e., the sensitive-feature classifiers. The counterfactual module generates $k$ counterfactuals for each original sample. Whether the sample is associated with a negative outcome (i.e., $f(\mathbf{x}) = 0$), it belongs to a privileged group (i.e., $s = 1$), and it is correctly predicted to belong to the same sensitive class (i.e., $f_s(\mathbf{x}) = 1$), then the sample and its counterfactuals are added to the set $\mathcal{A}$. Alternatively, if the sample is associated with a negative outcome (i.e., $f(\mathbf{x}) = 0$), it belongs to an unprivileged group (i.e., $s = 0$), and it is correctly predicted to belong to the same sensitive class (i.e., $f_s(\mathbf{x}) = 0$), then the sample and its counterfactuals are added to the set $\mathcal{B}$. In detail, for each sample, a tuple of objects is stored, including: (i) the original sample $\mathbf{x}$, (ii) the predicted target label $f(\mathbf{x})$, (iii) the sensitive feature of the sample as it is predicted by the dedicated classifier $f_s(\mathbf{x})$, (iv) the set of counterfactual samples $\mathcal{C}_{\mathbf{x}}$, (v) and the predictions of the sensitive labels performed on the counterfactuals $f_s(c_{\mathbf{x}})\ \forall\ c_{\mathbf{x}} \in \mathcal{C}_{\mathbf{x}}$. The process is summarized by Algorithm 1.

**Algorithm 2:** Counterfactual Flips Evaluation

**Input:**

- the number of counterfactuals to be generated for each sample $N_{CF}$,
- the set $\mathcal{A}$ composed of tuples of objects related to the samples associated with a **negative outcome** (i.e., $f(\mathbf{x}) = 0$), belonging to an **privileged group** (i.e., $s = 1$), and correctly predicted to belong to the same sensitive class (i.e., $f_s(\mathbf{x}) = 1$),
- the set $\mathcal{B}$ composed of tuples of objects related to the samples associated with a **negative outcome** (i.e., $f(\mathbf{x}) = 0$), belonging to an **unprivileged group** (i.e., $s = 0$), and correctly predicted to belong to the same sensitive class (i.e., $f_s(\mathbf{x}) = 0$).

**Result:**

- the vector **priv** of size $N_{CF}$ that contains averaged CFlips values, across all samples, of counterfactuals in $\mathcal{A}$ sorted in descending order of similarity, as returned by the counterfactual generator. The $i$th element of **priv** is the average of CFlips values considering $i$ counterfactuals for all the samples.
- the vector **unpriv** of size $N_{CF}$ that contains averaged CFlips values, across all samples, of counterfactuals in $\mathcal{B}$ sorted in descending order of similarity, as returned by the counterfactual generator. The $i$th element of **unpriv** is the average of CFlips values considering $i$ counterfactuals for all the samples.

```
Initialize priv = $\{0, 0, \dots, 0\}$, and unpriv = $\{0, 0, \dots, 0\}$;
for $k \leftarrow 1$ to $N_{CF}$ do
    $n_p \leftarrow 0$;
    for $l_p^i \in \mathcal{A}$ do
        $\mathbf{x}^{(i)}, \mathcal{C}_{\mathbf{x}^{(i)}}, \hat{y}^{(i)}, \hat{s}^{(i)}, \hat{\mathbf{s}}_{CF} \leftarrow l_p^i$;
        $n_p \leftarrow n_p + 1$;
        $\mathbf{priv}[k] \leftarrow \mathbf{priv}[k] + \text{CFlips}(\mathbf{x}^{(i)}, \text{sorted}(\mathcal{C}_{\mathbf{x}^{(i)}})[:k], \hat{\mathbf{s}}_{CF}[:k])$;
    end
    $\mathbf{priv}[k] \leftarrow \mathbf{priv}[k]/n_p$;
    $n_{unp} \leftarrow 0$;
    for $l_{unp}^i \in \mathcal{B}$ do
        $\mathbf{x}^{(i)}, \mathcal{C}_{\mathbf{x}^{(i)}}, \hat{y}^{(i)}, \hat{s}^{(i)}, \hat{\mathbf{s}}_{CF} \leftarrow l_{unp}^i$;
        $n_{unp} \leftarrow n_{unp} + 1$;
        $\mathbf{unpriv}[k] \leftarrow \mathbf{unpriv}[k] + \text{CFlips}(\mathbf{x}^{(i)}, \text{sorted}(\mathcal{C}_{\mathbf{x}^{(i)}})[:k], \hat{\mathbf{s}}_{CF}[:k])$;
    end
    $\mathbf{unpriv}[k] \leftarrow \mathbf{unpriv}[k]/n_{unp}$;
end
```

The sets set $\mathcal{A}$ and set $\mathcal{B}$ are evaluated using the counterfactual metric CFlips (see Eq. (5)). Specifically, the metric CFlips applies for each tuple in $\mathcal{A}$ and $\mathcal{B}$ to all the counterfactuals therein. The CFlips values are then averaged to obtain an overall value for $\mathcal{A}$ and $\mathcal{B}$, respectively. The evaluation pipeline is graphically depicted in Fig. 3. The procedure can be repeated for different values of $k$ and the different counterfactual generation strategies. To efficiently compute the metric CFlips for several values of $k$, two vectors (i.e., for $\mathcal{A}$ and $\mathcal{B}$) of size $k$ can be created to accumulate the CFlips values before averaging them. These vectors can be used to plot how CFlips vary over the number of considered counterfactuals (see plots in Section 6). The optimized procedure is condensed into Algorithm 2.

## 5. Experimental evaluation

This section details our experimental settings, designed to answer the research questions defined in Section 1. Two different models are trained: on the one hand, we train a model for making decisions for a specific task (i.e., income prediction or loan prediction), and on the other hand, we train the sensitive-feature classifiers to predict the sensitive group the samples belong to.

Specifically, we focus on the samples predicted as negative by the main task classifier. Next, we exploit counterfactual reasoning: starting from these samples classified as negative, we aim to modify features to cause a *flip* concerning the final prediction class (i.e., the prediction class goes from 0 to 1 by modifying one or more features). Subsequently, these new counterfactual samples feed the classifier for the sensitive features to predict the demographic group they belong to. In this way, we check if the counterfactual modifications have caused a flip concerning the sensitive group to which the sample belongs. The intuition here is that counterfactual-generated data are more explanatory in showing the model unfairness resulting from proxy features. The system's fairness can be evaluated by analyzing, for each test sample, any existing correlations between the target classification task and the protected classes inferred from counterfactuals.

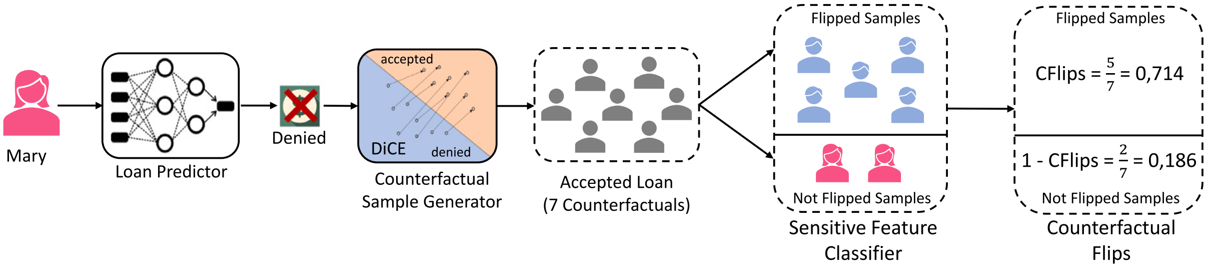

**Fig. 3.** Pipeline to detect bias, leveraging the proposed approach for a loan granting task. Mary, a woman, asks for a loan. Since the decision-maker operates in a *fairness under unawareness* setting, Mary does provide her gender information. The decision-maker denies the loan to Mary. Due to the denial, the counterfactual generator creates a group of seven potential profiles similar to Mary, who would have the loan request accepted. The Sensitive Feature Classifier reveals that most (five) of the profiles correspond to male profiles. This insight is summarized by the Counterfactual Flips (CFlips) metric, which measures the ratio of the counterfactuals identified as belonging to another demographic group. The pipeline resembles the evaluation workflow we applied for the German dataset (cf. Section 5.1.2).

**Table 3**
Adult and German datasets characteristics.

(a) Adult

| | Samples | $n^*$ | Target | $Y = 1$ |
|---|---|---|---|---|
| Train | 40699 | 6 | Income | $\geq \$50,000$ |
| Test | 4523 | 6 | Income | $\geq \$50,000$ |

(b) German

| | Samples | $n^*$ | Target | $Y = 1$ |
|---|---|---|---|---|
| Train | 900 | 17 | Credit score | Good |
| Test | 100 | 17 | Credit score | Good |

*Number of non-sensitive features in the dataset.

## 5.1. Datasets and preprocessing

Experiments are conducted on three state-of-the-art datasets, used as benchmarks in several works (Balunovic, Ruoss, & Vechev, 2022; Das et al., 2021; Donini et al., 2018; Pedreschi et al., 2008). Despite their small dimension, as stated by Rossini, Croce, Mancini, Pellegrino, and Basili (2020), these datasets are useful to evaluate fairness approaches because they represent real-world problems and provide a wide range of attributes that can be used to develop ethical standards. These are Adult and German, two real-world datasets used for income prediction and default prediction respectively. Now we provide a preliminary analysis of these datasets.

### 5.1.1. Adult dataset

Adult[15] is a popular UCI Machine Learning dataset extracted from the 1994 US Census database. The prediction task is to determine whether a person earns more than 50K a year. The sensitive attributes consider for this dataset are *gender* which indicates the sex of an individual, and *marital status*, whether an individual is married or not.

In the Adult dataset, there are other sensitive characteristics (i.e., *age*, *relationship*, and *race*). Since Fairness Under Unawareness, the setting most coherent with current AI regulations, requires bereaving the dataset of sensitive information during training, we decided not to use these features to learn the model. From the whole set of sensitive features we chose to investigate but not to use in the training phase, only *gender* and *marital-status* as classic sensitive information for benchmarking debiasing models (Donini et al., 2018; Guntzel, 2022; Oneto & Chiappa, 2020). As regards the non-sensitive features used for training the models, 6 out of 15 were used: *education num*, *occupation*, *work class*, *capital gain*, *capital loss*, *hours per week*. The remaining non-sensitive features are filtered out because they show a high correlation with the sensitive features (Pearson's correlation coefficient greater than 0.35). Furthermore, the feature *work class* is condensed into three classes: *Private*, *Public*, and *Unemployed*. We replace the categories in *work class Private*, *SelfEmpNotInc*, *SelfEmpInc*, with *Private*, the categories *FederalGov*, *LocalGov*, *StateGov*, with *Private*, and the category *WithoutPay* with *Unemployed*. This choice is taken to simplify the calculation of distances between counterfactual samples and **x** samples (see Section 6.3). The Adult dataset is imbalanced, as shown in Table 4a. This can emphasize some biases (Donini et al., 2018; Zemel, Wu, Swersky, Pitassi, & Dwork, 2013; Zhang et al., 2018). The target label *income* >= 50K is strongly unbalanced towards the *privileged* class (male, married). More detailed statistics, including the number of samples, the sensitive feature distribution, and the ex-ante statistical parity, are summarized in Table 3a, Table 4a, and Table 4c.

[15] https://archive.ics.uci.edu/ml/datasets/adult

**Table 4**
Overview of relevant dataset information, including sensitive feature distribution (a, b), name of privileged group (a, b), ex-ante Statistical parity (a, b), sample distribution over the target class and sensitive feature in the form of a confusion matrix (c, d).

(a) Adult sensitive feature distribution and ex-ante Statistical Parity

| | Sensitive-feature | Frivileged | Distribution† | ex-ante statistical parity‡ |
|---|---|---|---|---|
| $S_1$ | Gender | Male | 0,675/0,325 | 0.199 |
| $S_2$ | Marital status | Married | 0,478/0,522 | 0.378 |

(b) German sensitive feature distribution and ex-ante Statistical Parity

| | Sensitive-feature | Privileged | Distribution† | ex-ante Statistical Parity‡ |
|---|---|---|---|---|
| $S_1$ | Gender | Male | 0,690/0,310 | 0.075 |

†Probability distribution of the *advantaged* and *disadvantaged* group:
$\mathbb{P}(S_i = 1)/\mathbb{P}(S_i = 0)$.
‡A priori Statistical Parity probability, based on Independence Statistical Criteria:
$\mathbb{P}(Y = 1 \mid S_i = 1) - \mathbb{P}(Y = 1 \mid S_i = 0)$.

(c) Adult sensitive feature distribution over the target class *income*

| | | Marital-Status | | Gender | | |
|---|---|---|---|---|---|---|
| | | $s = 0$ | $s = 1$ | $s = 0$ | $s = 1$ | Samples |
| Income | $y = 0$ | 2201 | 1201 | 1303 | 2099 | 3402 |
| | $y = 1$ | 158 | 963 | 167 | 954 | 1121 |
| | Samples | 2359 | 2164 | 1470 | 3053 | |

(d) German sensitive feature distribution over the target class *credit score*

| | | Gender | | |
|---|---|---|---|---|
| | | $s = 0$ | $s = 1$ | Samples |
| Credit score | $y = 0$ | 11 | 19 | 30 |
| | $y = 1$ | 20 | 50 | 70 |
| | Samples | 31 | 69 | |

### 5.1.2. German dataset

German[16] is another popular UCI Machine Learning dataset extracted from a German bank loan approval history. Demographic and financial characteristics of individuals who applied for a loan are collected in this dataset, along with the decision to grant them a loan or not. The prediction task is the binary decision of approving a loan based on the probability of repaying it. The sensitive characteristic take into account is *gender*. As for the Adult dataset, German contains other sensitive characteristics (i.e., age and race) beyond those exploited in this study. Also, in this case, we do not include these features for learning the model for guaranteeing the *fairness under awareness* setting. We exploit 17 non-sensitive features to train the predictive models (i.e., *existingchecking*, *duration*, *credithistory*, *purpose*, *creditamount*, *savings*, *employmentsince*, *installmentrate*, *otherdebts*, *residencesince*, *property*, *otherinstallmentplans*, *housing*, *existingcredits*, *job*, *peopleliable*, *telephone*). As for the Adult dataset, German is imbalanced (Donini et al., 2018; Zemel et al., 2013; Zhang et al., 2018). Table 4b shows that the privileged group is overrepresented for both the sensitive features. Moreover, the ex-ante statistical parity metric indicates that the advantaged target label ($Y = 1$) is strongly associated with the privileged group ($S_i = 1$) compared to the unprivileged group ($S_i = 0$), which confirms that the data is imbalanced and strongly biased. Useful statistical details are reported in Table 3b, Table 4b, and Table 4d.

## 5.2. Evaluation metrics

The evaluation includes two different groups of metrics: accuracy-based and bias-based metrics. The accuracy-based metrics are mainly based on the confusion matrix, which quantifies how many samples are correctly classified or misclassified for both the negative and positive classes. For self-consistency, this section details all the considered metrics. Some are just recalled, reporting the formulas. The others, used in cutting-edge fairness research, are described. The first metric is the Accuracy, which quantifies the overall number of correct classifications over the predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{6}$$

The Recall metric measures the number of positive correctly classified samples with respect to all the real positive ones:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

[16] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Precision measures the ratio of samples correctly classified as positive over the ones classified as positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

The F1 score is the harmonic mean between recall and accuracy:

$$\text{F1} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

The primary goal of the F1 score is to combine the precision and recall metrics into a single metric. Indeed, this metric is useful for evaluating classification methods when dealing with imbalanced data. The Area Under the Receiver Operating Characteristic Curve (AUC) is a metric that measures the capability of a classifier to separate the positive class from the negative class correctly. It can be formulated as follows:

$$\text{AUC} = \frac{\sum_{x^- \in X^-} \sum_{x^+ \in X^+} (\mathbb{1}(f(x^-) < f(x^+)))}{|X|^- + |X|^+} \quad \text{where } \mathbb{1}(\cdot) = 1 \text{ if } f(x^-) < f(x^+) \text{ else } \mathbb{1}(\cdot) = 0; \tag{10}$$

where $X^+$ is the set of positive sample, $X^-$ is the set of negative sample, $f(\cdot)$ is the result of model prediction, and $\mathbb{1}(\cdot)$ an indicator function (Calders & Jaroszewicz, 2007).

To quantify the presence of bias in the decision of the two classifiers several fairness metrics were used that consider the *Independence* and *Separation* statistical criteria. For the Independence statistical criteria, we used *Difference in Statistical Parity* (DSP) and *Disparate Impact* (DI). DSP measures the difference between the probability that samples belonging to the *privileged* group and to the *unprivileged* group are classified in a positive outcome class (Hardt et al., 2016). It is the equivalent of the difference between the sum of the TP rate and FP rate of the *privileged* and *unprivileged* group (see Eq. (11)). A model is considered Fair w.r.t. DSP if the measure is equal or, at least, very close to zero.

$$\text{DSP} = \left|\mathbb{P}(\hat{Y} = 1|s = 1) - \mathbb{P}(\hat{Y} = 1|s = 0)\right| = \left|(\text{TPrate}_{priv} + \text{FPrate}_{priv}) - (\text{TPrate}_{unpriv} + \text{FPrate}_{unpriv})\right| \tag{11}$$

The latter, i.e., DI, measures the ratio between the probability that samples belonging to the *unprivileged* group and to the *privileged* group are classified in a positive outcome class (Das et al., 2021). It can also be formulated as the ratio between the sum of the TP and FP rate for each group (see Eq. (12)). A model is considered Fair w.r.t. DI if the considered measure is near the value of one.[17]

$$\text{DI} = \frac{\mathbb{P}(\hat{Y} = 1|s = 0)}{\mathbb{P}(\hat{Y} = 1|s = 1)} = \frac{\text{TPrate}_{unpriv} + \text{FPrate}_{unpriv}}{\text{TPrate}_{priv} + \text{FPrate}_{priv}} \tag{12}$$

For the *Separation* statistical criteria, we used *Difference in Equal Opportunity* (DEO) and *Difference in Average Odds* (DAO). The former, i.e., DEO, measures the difference between the probability of instances in a *privileged* group and the probability of instances in an *unprivileged* group being correctly classified in a positive outcome class (Hardt et al., 2016). The formulation of the DEO metric is shown in Eq. (13).

$$\text{DEO} = \left|\mathbb{P}(\hat{Y} = 1|Y = 1, s = 1) - \mathbb{P}(\hat{Y} = 1|Y = 1, s = 0)\right| = \left|\text{TPrate}_{priv} - \text{TPrate}_{unpriv}\right| \tag{13}$$

The latter, i.e., DAO, measures the difference between the probability of instances in a *privileged* group and the probability of instances in an *unprivileged* group being correctly classified in a positive outcome class, as DEO does. Furthermore, DAO also considers the difference between the probability of instances in a *privileged* group and the probability of instances in a *privileged* group being incorrectly classified in a positive outcome class. DAO gives a broader intuition of how much imbalanced the classifier accuracy is between the two groups (Hardt et al., 2016). The formulation of the DAO metric is shown in Eq. (14).

$$\begin{aligned}\text{DAO} &= \frac{\left|\mathbb{P}(\hat{Y} = 1|Y = 0, s = 1) - \mathbb{P}(\hat{Y} = 1|Y = 0, s = 0)\right| + \left|\mathbb{P}(\hat{Y} = 1|Y = 1, s = 1) - \mathbb{P}(\hat{Y} = 1|Y = 1, s = 0)\right|}{2} \\ &= \frac{\left|\text{FPrate}_{priv} - \text{FPrate}_{unpriv}\right| + \left|\text{TPrate}_{priv} - \text{TPrate}_{unpriv}\right|}{2}\end{aligned} \tag{14}$$

In either case, for DEO and DAO, a model is considered fair if the measure is equal or, at least, very close to zero.

### *5.3. Evaluation protocol and reproducibility*

**Dataset Splitting.** The dataset was split with the random 90/10 hold-out method to partition train and test sets, with stratification based on the target variable $\mathcal{Y}$ and the sensitive features $\mathcal{S}$. For the Adult dataset, we have 40699 train samples and 4523 test samples (see Table 3a), and for the German dataset, 900 train samples and 100 test samples (see Table 3b). For reproducibility, we used the Scikit-learn implementation for splitting with a random seed set to 42.[18]

[17] In an employment context in the US, the regulation of The Equal Opportunity Act is known as "80% rule" or as a "rule of thumb" for measuring disparate impact (Das et al., 2021); DI value should be between 0.8 and 1.2

[18] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

**Table 5**
Hyperparameter list, values and type for the classification models reported in this work.

| Algorithm | Hyperparameter | Values | Type |
|---|---|---|---|
| Logistic Regression | seed | {42} | Integer |
| | penalty | {l1,l2} | String |
| | tol | {0.0001,0.00001} | Float |
| | C | $\{10^{-4+(\frac{8}{20}i)}$ for $i$ in $range(1,21)\}$ | Float |
| | fit_intercept | {True, False} | Boolean |
| | class_weight | { dict, balanced, None} | String |
| | solver | {newton-cg, lbfgs, liblinear, sag, saga} | String |
| | warm_start | {True, False} | Boolean |
| Support Vector Machines | seed | {42} | Integer |
| | C | {0.1, 1, 10} | Float |
| | class_weight | {balanced, None} | String |
| | gamma | {scale, auto} | String |
| | kernel | {linear, rbf, sigmoid} | String |
| eXtreme Gradient Boosting | seed | {42} | Integer |
| | min_child_weight | {1, 5, 10} | Integer |
| | gamma | {0.01, 0.1, 0.5} | Float |
| | learning_rate | {0.1, 0.01, 0.001} | Float |
| | max_depth | {3, 5, 6} | Integer |
| | subsample | {0.4,0.6,0.8,1.0} | Float |
| | colsample_bytree | {0.6, 0.8, 1} | Float |
| | n_estimators | {50, 100, 300,500} | Integer |
| | reg_alpha | {0.1, 0.01, 0.02} | Float |
| Light Gradient Boosting | seed | {42} | Integer |
| | learning_rate | {0.1, 0.05} | Float |
| | num_leaves | {3, 10, 30, 50, 100, 200} | Integer |
| | reg_alpha | {None, 0.01, 0.05, 0.1} | Float |
| | colsample_bytree | {0.6, 0.8,1} | Float |
| | max_depth | {−1, 3, 5, 8, 10} | Integer |
| | reg_lambda | {None, 0.01, 0.02, 0.03} | Float |
| | n_estimators | {50, 100, 300} | Integer |
| Adversarial Debiasing | seed | {42} | Integer |
| | adversary_loss_weight | {0.01, 0.05, 0.1} | Float |
| | num_epochs | {50, 70, 150, 250, 500} | Integer |
| | batch_size | {64, 128, 256, 512} | Integer |
| | hidden_units | {64, 128, 256} | Integer |
| | number_of_layers | {1}[a] | Integer |
| Linear Fair Empirical Risk Minimization | seed | {42} | Integer |
| | C | {0.01, 0.1, 1} | Float |
| | kernel | {linear} | String |

[a] AIF360 implementation of Adversarial Debiasing does not allow to change the number of layers.

**Decision-Maker Hyperparameter Tuning and optimization.** The target label classifiers, i.e., LR, SVM, XGB, and LGB (see Section 4.1), have been tuned using a grid search strategy.[19] For hyperparameter tuning and validation, the train data was further split using a k-fold cross-validation strategy, with the number of folds set to five. The best models hyperparameter has been chosen to optimize the Area under the ROC curve metric (AUC) since AUC indicates how well the classifier can separate the positive from the negative class (see Eq. (10)). For reproducibility, the list of explored hyperparameter values is reported in Table 5.

**Debiased Decision-Makers Hyperparameter Tuning and optimization.** The Debiasing classifiers, i.e., AdvDeb and LFERM (see Section 4.1.1), have been tuned using the same evaluation protocol, with a grid search for the hyperparameter values and a 5-fold cross-validation strategy. Conversely, in this evaluation, the best models have been chosen to optimize AUC and Fairness with an overall metric that considers both:

$$\mathrm{AUC_{FAIR}} = \mathrm{AUC} \cdot (1 - \mathrm{DAO}) \tag{15}$$

It is straightforward noticing that any other Fairness metric could replace DAO. In this work, DAO is chosen to balance fairness in terms of correct predictions for negative and positive samples. The list of explored hyperparameter values is reported in Table 5.

**Sensitive Feature Classifier Hyperparameter Tuning and optimization.** The sensitive label classifiers, i.e., XGB (see Section 4.3), are tuned using the same approach, exploiting a grid search exploration[19] for hyperparameter values and a 5-fold cross-validation strategy. Due to the imbalanced nature of the datasets concerning the sensitive classes, the models optimizing the F1 score are chosen (see Eq. (9)). Explored hyperparameter values are shown in Table 5.

[19] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

**Counterfactual generation.** For the sake of reproducibility, the generation of counterfactual samples makes use of DiCE, as discussed in Section 4.2. To avoid the results depending on a single counterfactual generation strategy, we considered three different strategies, i.e., Random, Genetic, and KDtree. For the Random strategy, the *seed* has been set to 42, the *posthoc sparsity parameter* to 0.1, and the *posthoc sparsity algorithm* to *linear search*. For the Genetic strategy, we set the *initialization* to *kdtree*, the *proximity weight* to 0.2, the *sparsity weight* to 0.2, the *diversity weight* to 5, the *categorical penality* to 0.1, the counterfactual generation loss to *hinge-loss*, the *feature weights* to *inverse Mean Absolute Deviation* (MAD), the *posthoc sparsity parameter* to 0.1, the *posthoc sparsity algorithm* to *binary search*, and the *max iterations* to 500. For the KDtree strategy, we set the *sparsity weight* to 1, the *feature weights* to *inverse Mean Absolute Deviation* (MAD), the *posthoc sparsity parameter* to 0.1, and the *posthoc sparsity algorithm* to *linear search*. For each sample in the test set, an overall number of 100 counterfactuals was requested (see Algorithm 1). For reproducibility reasons, we use all the previously listed default parameter values of the DiCE tool, except for the *posthoc sparsity algorithm* set to *binary search* in the Genetic strategy for speeding up the search due to the expensive experimental time.

**Distance between counterfactuals and original samples.** To assess the quality of the counterfactuals generated with the various strategies, the distance vector between the original samples and the corresponding counterfactuals are calculated (see Section 6.3). Counterfactuals that belong to *privileged* samples (i.e., $\forall \mathbf{c_x} \in C_\mathbf{x} \wedge f_s(\mathbf{x}) = s = 1$) and counterfactuals that belong to *unprivileged* samples (i.e., $\forall \mathbf{c_x} \in C_\mathbf{x} \wedge f_s(\mathbf{x}) = s = 0$) were analyzed separately. To calculate the distance vector, we follow the principle of *Credit Risk scorecard* models[20] in the financial domain to transform the categorical features of the dataset into continuous variables.

For the Adult dataset, we have two categorical features: *workclass* and *occupation*. The *workclass* categories, i.e. *Public*, *Private*, and *Unemployed*, has been substitute respectively with the values 1, 2, and 0. For the *occupation* feature, the category *Other-service* has been substituted with the value 1, the categories *Adm-clerical*, *Handlers-cleaners*, *Sales*, *Transport-moving*, *Farming-fishing*, *Machine-op-inspct*, *Craft-repair*, and *Priv-house-serv* with the value 2, the categories *Prof-specialty*, *Tech-support*, and *Protective-serv* with the value 3, and the categories *Exec-managerial* and *Armed-Forces* with the value 4. This operation was necessary to quantify the polarity of discrimination in the categorical features analogously to how we quantify it for numerical features.

For the German dataset, we replace the category of each categorical feature with the actual scorecard value.[21] The metric used to calculate the distance vector for both *privileged* and *unprivileged* samples is formalized as follows (Eq. (16)):

$$\Delta_{dist} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k} dist(\mathbf{x}^i, \mathbf{c}^j_{\mathbf{x}^i})}{n \cdot k} \quad \text{where } dist(\mathbf{x}, \mathbf{c_x}) = \frac{\mathbf{c_x} - \mathbf{x}}{\mathbf{x}} \tag{16}$$

## 6. Discussion of the results

This Section depicts, describes, and discusses the experimental results. The rationale of the discussion is to provide the reader with an in-depth understanding of the critical classifiers and unveil how the proposed method highlights potential biases. For clarity, the discussion follows the research questions introduced in Section 1:

- **RQ1:** Is there a principled way to identify if proxy features exist in a dataset?
- **RQ2:** Does the Fairness Under Unawareness setting ensure that decision biases are avoided?
- **RQ3:** Is counterfactual reasoning suitable for discovering decision biases?
- **RQ4:** Is our methodology effective for discovering discrimination and biases? Are there limitations in its application?

### *6.1. RQ1: Is there a practical way to identify if proxy features exist in a dataset?*

The goal of this experimental evaluation aims to assess the capability of our methodology to predict sensitive features from non-sensitive ones. In fact, as analyzed earlier, more is needed to exclude sensitive features during the training phase to guarantee that a decision model is not affected by biases and does not implement discrimination. In order to answer RQ1, we trained a sensitive-feature classifier as introduced in Section 4 for both datasets. The insight is that, if we are able to predict with reasonable accuracy sensitive features from non-sensitive ones, it is very likely that proxy features occur in the dataset. Accordingly, we evaluate the presence of proxy features $\mathbf{p}$ in the data $\mathcal{X}$ by assessing the performance of the XGB sensitive-feature classifier.

We trained the XGB sensitive-feature classifiers for the Adult and German datasets. As mentioned in Section 5.3, both models have been tuned to maximize the F1 score to balance the precision and the recall as a common strategy for imbalanced datasets. For the Adult dataset, we predicted gender and marital status as sensitive features; for the German dataset, gender is predicted (other sensitive features are not sufficiently represented for learning a classification model). The performance of these models is shown in Table 6a for the Adult dataset, and in Table 6b for the German. On the Adult dataset, the best performance is shown for the *gender* feature for all the metrics, except for AUC, which has very similar values for both gender and marital status. The gender classifier on the German dataset shows a higher recall than the Adult (Table 6b) even though the F1 is essentially the same. Overall, the accuracy (ACC) is around 70% (best value on Adult for gender, $\sim 0.74$), showing a good capability of predicting the three sensitive features we focused on.

[20] Credit Risk scorecards are probabilistic models that evaluate the creditworthiness of a credit applicant, giving a score for specific values or category based on a probabilistic threshold risk tolerance.

[21] We used the Credit scorecard values available at https://online.stat.psu.edu/stat857/node/222/ for German categorical features.

**Table 6**
AUC, Accuracy, F1 score, and Recall performance of the XGB sensitive feature classifiers for Adult (a) and German (b) dataset.

(a) Performance on Adult's sensitive features.

| | Gender | Marital-Status |
|---|---|---|
| AUC | 0.7803 | 0.7736 |
| ACC | 0.7404 | 0.6920 |
| F1 | 0.8067 | 0.6779 |
| Recall | 0.8022 | 0.6774 |

(b) Performance on German's sensitive feature.

| | Gender |
|---|---|
| AUC | 0.7139 |
| ACC | 0.6900 |
| F1 | 0.8025 |
| Recall | 0.9130 |

**Table 7**
Accuracy and Fairness results of Classic (i.e., LR, SVM, XGB, and LGBM) and Debiasing (i.e., AdvDeb and LFERM) classifiers for the Adult dataset stratified with respect to the *income* target and the sensitive label. We mark the best-performing method for each metric in bold font;.

(a) Accuracy and Fairness results, considering *income* as target label and *gender* as sensitive label

| | Classifier | | | | Debiasing | |
|---|---|---|---|---|---|---|
| | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| AUC | 0.8233 | 0.8189 | 0.8592 | **0.8596** | 0.8309 | 0.8017 |
| ACC | 0.7367 | 0.7395 | **0.8393** | 0.8371 | 0.8203 | 0.7953 |
| F1 | 0.5726 | 0.5735 | **0.5862** | 0.5796 | 0.5276 | 0.4176 |
| Recall | **0.7119** | 0.7065 | 0.4594 | 0.4532 | 0.4050 | 0.2962 |
| DSP | 0.1567 | 0.1062 | 0.1056 | 0.1093 | 0.0957 | **0.0639** |
| DI | 0.6263 | **0.7328** | 0.3919 | 0.3766 | 0.4179 | 0.4864 |
| DEO | 0.1515 | 0.1425 | 0.0991 | 0.0969 | 0.0852 | **0.0563** |
| DAO | 0.0783 | 0.0894 | 0.0548 | 0.0546 | 0.0484 | **0.0320** |

(b) Accuracy and Fairness results, considering *income* as target label and *marital status* as sensitive label

| | Classifier | | | | Debiasing | |
|---|---|---|---|---|---|---|
| | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| AUC | 0.8209 | 0.8177 | 0.8608 | **0.8633** | 0.8265 | 0.7444 |
| ACC | 0.7396 | 0.7433 | 0.8441 | **0.8455** | 0.8211 | 0.7590 |
| F1 | 0.5771 | 0.5804 | 0.6046 | **0.6062** | 0.5316 | 0.1780 |
| Recall | **0.7172** | 0.7163 | 0.4808 | 0.4799 | 0.4095 | 0.1053 |
| DSP | 0.1793 | 0.1493 | 0.1663 | 0.1702 | 0.1241 | **0.0336** |
| DI | 0.6116 | **0.6620** | 0.2832 | 0.2707 | 0.3755 | 0.4654 |
| DEO | 0.2537 | 0.2515 | 0.1729 | 0.1733 | 0.1412 | **0.0457** |
| DAO | 0.1640 | 0.1768 | 0.0904 | 0.0882 | 0.0792 | **0.0289** |

#### 6.1.1. Observations

The results from this first experimental evaluation laid the foundation for going ahead with our investigation. In fact, we demonstrated that it is possible to learn a classifier that is able, with quite a good accuracy, to predict sensitive features even though these are not exploited during the training phase. The motivation behind this result is that *the classifier is able to discover hidden patterns in the non-sensitive features that allow attributing a user to the privileged or the unprivileged class. We can now answer the RQ1 positively* and collect clues on RQ2, namely that the Fairness Under Unawareness setting is very likely not enough to guarantee that biases are avoided.

### 6.2. RQ2: Does the fairness under unawareness setting ensure that decision biases are avoided?

The *Fairness Under Unawareness* setting tries to ensure fairness of treatment by removing the direct link between prediction and sensitive features. Sensitive features are then excluded from training data. However, as demonstrated previously, it is possible to predict sensitive information when proxy features occur in the data. Motivated by the results of the previous experimental evaluation, here we provide a deep analysis of two decision-makers, namely the income and credit-score predictor for the Adult and German datasets, respectively. The analysis is carried out in terms of the models' accuracy and fairness.

The results of the income predictor for the Adult dataset are shown in Table 7, whereas those of the credit-score predictor for the German dataset are shown in Table 8. For the Adult dataset, in order to maintain in the test set the same distribution of the original dataset, for each sensitive feature, we performed two different stratifications: one for *income* and *gender* (Table 7a) and one for *income*

**Table 8**
Accuracy and Fairness results of Classic (i.e., LR, SVM, XGB, and LGBM) and Debiasing (i.e., AdvDeb and LFERM) classifiers for the German dataset stratified with respect to the *credit score* target and the sensitive label. We mark the best-performing method for each metric in bold font;.

(a) Accuracy and Fairness results, considering the *credit-score* as target label and the *gender* as sensitive label

| | Classifier | | | | Debiasing | |
|---|---|---|---|---|---|---|
| | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| AUC | **0.8186** | 0.8109 | 0.8014 | 0.7871 | 0.7910 | 0.7686 |
| ACC | 0.7600 | 0.7600 | 0.7700 | **0.8200** | 0.7800 | 0.7200 |
| F1 | 0.8235 | 0.8442 | 0.8477 | **0.8800** | 0.8493 | 0.8313 |
| Recall | 0.8000 | 0.9286 | 0.9143 | 0.9429 | 0.8857 | **0.9857** |
| DSP | 0.1187 | 0.0449 | 0.0986 | 0.0374 | 0.1197 | **0.0355** |
| DI | 1.1900 | 1.0540 | 0.8826 | 0.9539 | 0.8499 | **0.9634** |
| DEO | **0.0299** | 0.0538 | 0.1328 | 0.0683 | 0.1973 | 0.0650 |
| DAO | 0.0594 | 0.0762 | 0.0835 | 0.0496 | 0.1374 | **0.0472** |

and *marital status* (Table 7b). In either case, we can observe that all classifiers work well in terms of accuracy-based metrics, with gradient boosting-based classic models (i.e., XGB and LGBM) that outperform other methods. The only metric on which XGB and LGBM do not show excellent performance is recall, although F1 is comparable to LR and SVM due to a high Precision of gradient boosting-based models. By comparing non-debiased with debiased models, non-debiased models (i.e., LR, SVM, XGB, LGBM) are generally more accurate than the debiased ones (i.e., AdvDeb., LFERM). Regarding the fairness metrics, the debiased models show an overall better performance for both statistical criteria (i.e., DI and DSP). The fairest model is LFERM, which is also the one with the worst accuracy performance. The *Adversarial Debiasing* algorithm shows fairness performance similar to LR, SVM, XGB, and LGBM models. No algorithm can be considered fair in terms of DI metric. Only SVM approaches the minimum acceptable fairness value. The DI metric tells us that a model is fair if and only if its value is between 0.8 and 1.2, in contrast with the DSP, which does not overly highlight unfair behavior. This discrepancy between DSP and DI, both belonging to the Independence statistical criterion, can be due to the low probability of having a positive outcome for the *privileged* and *unprivileged* groups. Furthermore, the DSP does not highlight the proportion between the two probabilities, which DI instead measures. Regarding the score-based metrics, i.e., DEO and DAO, for the Separation's statistical criterion metrics, LFERM shows better performance than all the other algorithms.

For the German dataset, the *credit score* predictor results are available in Table 8. In contrast with Adult, no algorithm generally outperforms the others considering all the accuracy-based metrics. In fact, if we consider the Accuracy and F1, the best algorithm is LGBM, while, considering AUC, LR slightly exceeds SVM and XGB. The best algorithm in terms of Recall is LFERM, while there is a similar performance between LGBM and SVM. However, LFERM generally has the worst performance in terms of accuracy and AUC. Regarding fairness, LFERM shows the best performance for all the metrics except for DEO, where LR and SVM outperform LFERM. Both LGBM and SVM have performance very similar to LFERM. The Adversarial Debiasing algorithm shows the worst performance. The cause is probably the dataset size, given the deep-learning-based nature of the classifier. On the contrary, LFERM turned out to be an excellent model with consistently effective results. Compared to Adult, the DI values for each algorithm meet the "rule of thumb" requirements. In fact, the DI value in each case is in the range 0.8–1.2, and the best value is achieved by LFERM, which is the closest to 1. Finally, non-debiased models consistently outperform the debiased ones in terms of accuracy for both datasets. Yet, they turn out to have significant shortcomings in terms of fairness.

#### 6.2.1. Observations

The outcome of this experimental evaluation is twofold. First of all, *fairness metrics confirm that all the decision-makers we tested are more o less affected by biases*; secondly, *debiased algorithms are not sufficiently robust to ensure fair behaviors and decisions*. Therefore, neither avoiding the use of sensitive features nor exploiting debiased algorithms is enough to keep from discrimination. The cause is, in our opinion, the presence of proxy features in the data. Both debiased and not-debiased algorithms make positive predictions unbalanced towards privileged groups. *We can now answer the RQ2 and confirm that the Fairness Under Unawareness setting is not sufficient to avoid decision biases. We get further confirmation that the Fairness Under Unawareness setting turns out to be of no help when proxy features are in the data.*

### 6.3. RQ3: Is counterfactual reasoning suitable for discovering decision biases?

As introduced in Section 4.1, the counterfactual generator is a crucial component of the proposed methodology. Its role is to modify the original sample to reverse the decision made by the decision-maker. However, since the counterfactual generator must meet both feasibility and diversity of counterfactuals (CFs), it is not obvious that it succeeds in generating new counterfactuals. For this reason, in this experimental evaluation we evaluated the counterfactual generation $C_{\mathbf{x}}$ provided by DiCE. Two perspectives are scrutinized: (i) the capability of generating a certain number of counterfactuals and (ii) their likelihood compared to the original samples. This analysis also considers the different behavior between the privileged and unprivileged groups on the gender-sensitive feature for both datasets. For each generation strategy and decision model, we report the total generated CFs, the average number of generated CFs for each sample in the test set, and the number of generated CFs for unprivileged and privileged groups. The statistics for the Adult dataset are reported in Table 9, with Table 9a considering the *gender* as the sensitive feature and Table 9b

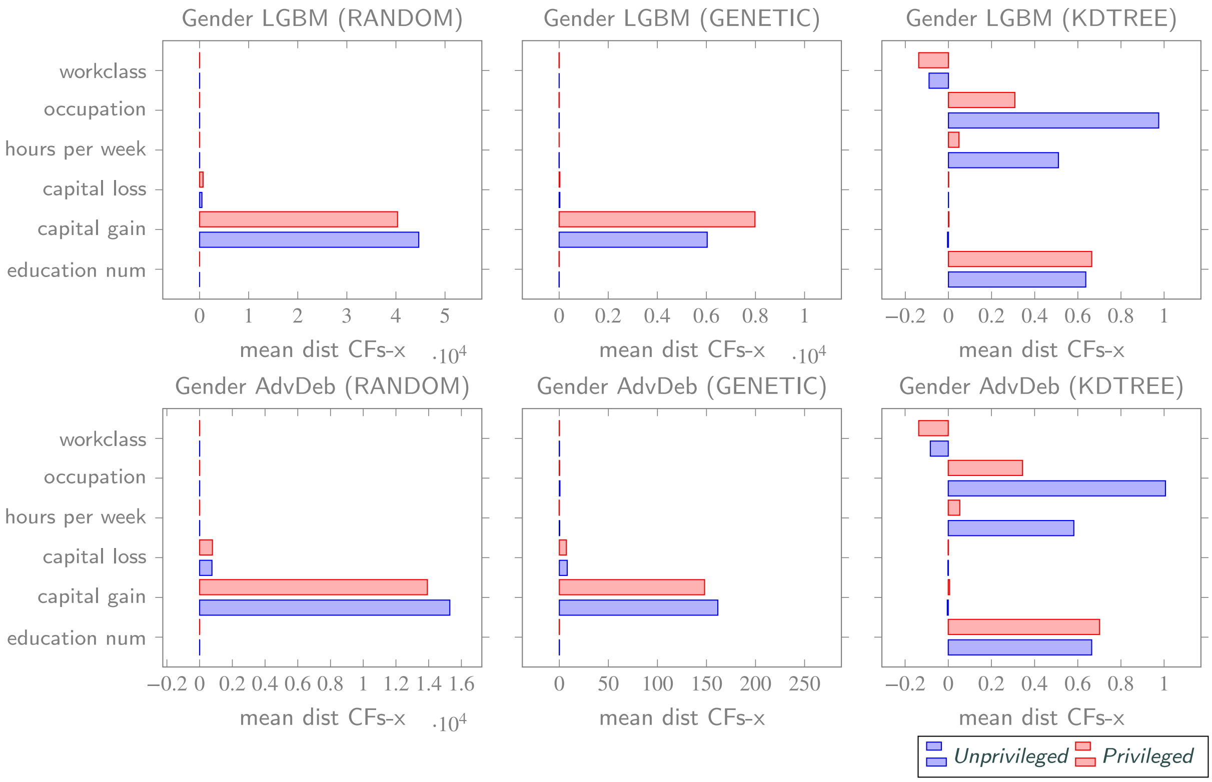

**Fig. 4.** Average distance of counterfactuals from the original samples (i.e., $\mathbf{x}, \forall \mathbf{x} \in \mathcal{X}_{test}$), for both *unprivileged* (blue bar) and *privileged* (red bar) *gender* group samples of Adult dataset. A bigger distance denotes a more substantial effort for that demographic group to reverse the decision-maker decision. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

considering the *marital status* as the sensitive feature. The strategy able to generate the most significant number of counterfactuals is Random. It generates 98–100 CFs for a sample on average. The Genetic strategy has a similar behavior, while the worst approach is KDtree. In fact, it generates 63–100 CFs per sample on average by varying the predictive model. The main reason is that KDtree searches CFs in the original sample space. Therefore, it fails to find CFs that are similar enough to the original sample (due to the proximity constraint) and guarantees a reverse prediction. A hindrance to the counterfactual generation can also be the predictive model. Indeed, since the CFs must reverse the decision, this result generally depends on the decision boundary of each model. From that perspective, the more a model is robust, the harder the generation of the counterfactuals is. In detail, SVM with KDTree is the model that shows the lowest number of generated counterfactuals.

Statistics on the German dataset are reported in Table 10. The Random strategy generates the maximum possible number of counterfactuals, except for AdvDeb. A possible motivation might be that DiCe fails to generate CFs that reverse the decision, probably due to the proximity requirement. However, this aspect requires further investigation in the future. The same behavior is for LFERM and Kdtree. The problem might be that Kdtree explores the sample space for founding CFs, and this search might fail, as explained earlier. Another remarkable case is the number of generated CFs for LFERM, which is 0. This is due to the fact that LFERM does not predict a negative outcome for any sample belonging to the unprivileged group. The same is for SVM.

The second analysis is related to the counterfactual likelihood. For the sake of clarity, this analysis involved two models (LGBM and AdvDeb) for each dataset on which the generation strategy is compared. Fig. 4 reports, for each feature, the average distance of the generated counterfactuals from the original samples for the Adult dataset, whereas Fig. 5 reports the average distance for German. The distance has been computed following Eq. (16). These charts give us a snapshot of the effort required to change the original sample to get a reverse decision for the privileged and unprivileged groups. For the Adult dataset, as expected, Random is the strategy that generated the furthest counterfactuals from the original samples. We can say that Random has more ‘freedom of movement’ in the feature space compared to other strategies since CFs are randomly generated, of course. On the contrary, Kdtree generates the closest counterfactuals given the search-based approach adopted. Genetic is a middle ground between Random and KDtree: it generates samples not so far as random and not so close as Kdtree. However, as mentioned above, Genetic shows better coverage in generating CFs compared to Kdtree. Looking at the features on which changes have more impact, the KDtree behavior is the opposite of Random and Genetic. Random and Genetic enforce a change of the *capital gain* feature. KDtree has minimal change

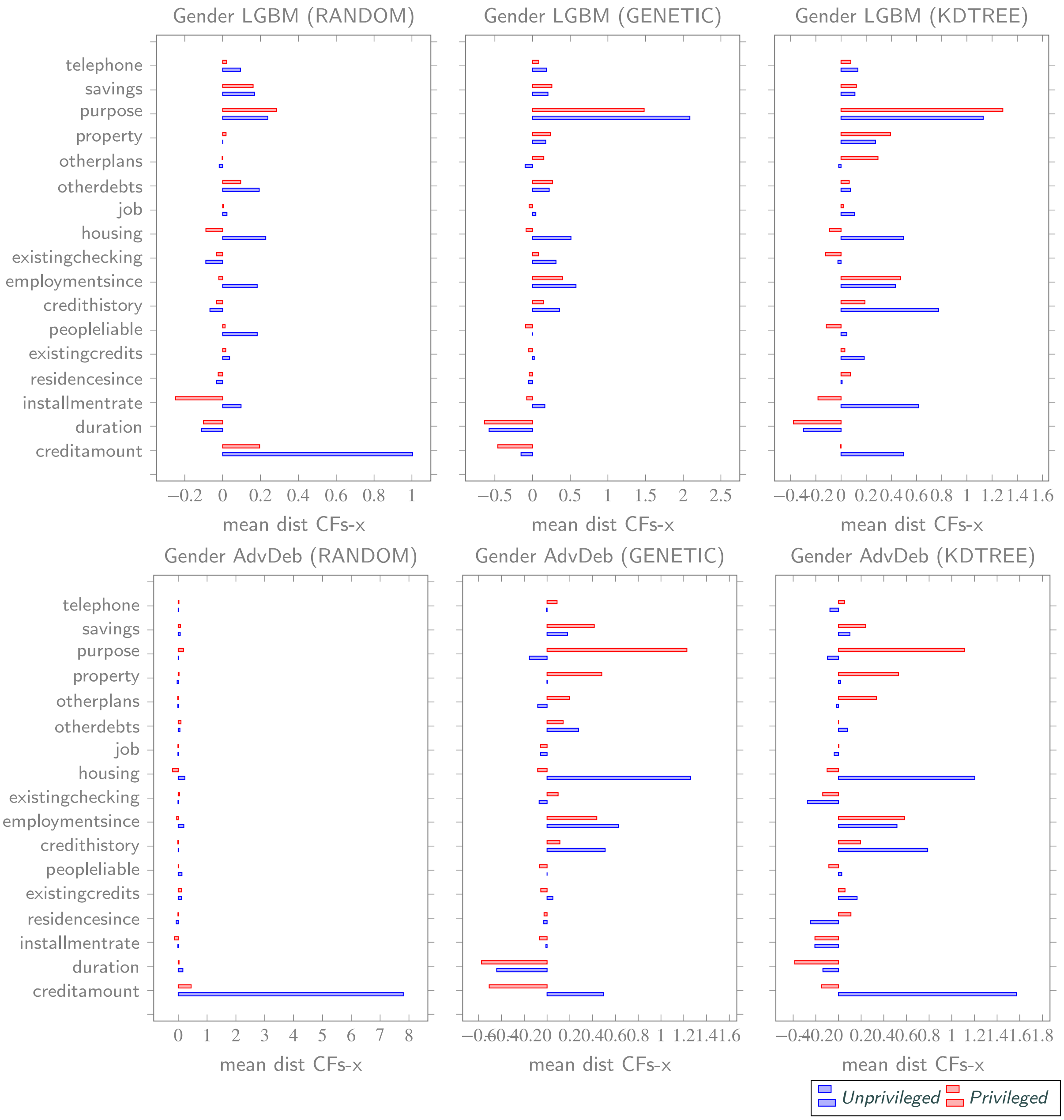

**Fig. 5.** Average distance of counterfactuals from the original samples (i.e., $\mathbf{x}, \forall \mathbf{x} \in \mathcal{X}_{test}$), for both *unprivileged* (blue bar) and *privileged* (red bar) *gender* group samples of German dataset. A bigger distance denotes a more substantial effort for that demographic group to reverse the decision-maker decision. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

on that feature and impacts more *occupation*, *hours per week*, and *education num*. From the decision-model perspective, *Adversarial Debiasing* requires more minor changes than LGBM for reversing the decision, except for the KDtree strategy that is the same for both algorithms (as we steadily underlined, the explored sample space is the same for both the algorithms). By comparing the unprivileged with the privileged group, the former needs more significant changes than the latter to reverse the decision, except for *workclass* with Kdtree and capital gain with LGBM and Genetic. For Kdtree, the motivation is that probably the closest sample that allows the reverse decision has a lower *workclass* than the unprivileged group. For LGBM and Genetic, the effect is not trivial to explain. Probably, LGBM learns a model where men (privileged group for gender) with income >= 50,000 usually have a significant capital gain. Surprisingly, Random has a distance between −0.2 and +1 on average for the German dataset, thus generating counterfactuals

**Table 9**
Adult statistics for *gender* and *Marital Status*, for each decision-maker, i.e., LR, SVM, XGB, LGBM, AdvDeb, and LFERM, and for each Counterfactual generation strategy. Statistics include the number of generated Counterfactuals (i.e., the sum of all the Counterfactuals samples generated for all the samples), the number of test set samples with at least one Counterfactual, and the percentage of generated Counterfactuals for each sample (with respect to the required 100), the number of Counterfactuals for negatively predicted samples and correctly predicted as unprivileged or privileged and collected in the sets $\mathcal{B}$ and $\mathcal{A}$, respectively (see Algorithm 1).

(a) Adult statistics considering *gender* as sensitive feature

| CF method | Statistics | Gender | | | | | |
|---|---|---|---|---|---|---|---|
| | | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| Random | Total generated CFs | 442292 | 452300 | 452300 | 452300 | 392000 | 442805 |
| | Mean generated CFs for a sample | 97.78 | 100 | 100 | 100 | 100 | 97.9 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 87 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 81400 | 81200 | 86900 | 86300 | 86100 | 85700 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 125500 | 132700 | 19800 | 195400 | 198200 | 211000 |
| Genetic | Total generated CFs | 444796 | 438912 | 435069 | 44577 | 392000 | 449241 |
| | Mean generated CFs for a sample | 98.34 | 97.04 | 96.19 | 98.55 | 100 | 99.3 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 87 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 79882 | 79378 | 84145 | 85452 | 86200 | 85198 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 122411 | 129674 | 188601 | 192063 | 198300 | 209348 |
| Kdtree | Total generated CFs | 327693 | 285787 | 422504 | 402782 | 376729 | 452300 |
| | Mean generated CFs for a sample | 72.45 | 63.18 | 93,412 | 89,051 | 83,291 | 100 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 60749 | 52966 | 84129 | 80556 | 71069 | 111670 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 76824 | 63291 | 180078 | 167322 | 160654 | 207885 |

(b) Adult statistics considering *marital-status* as sensitive feature

| CF method | Statistics | Marital Status | | | | | |
|---|---|---|---|---|---|---|---|
| | | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| Random | Total generated CFs | 441897 | 452300 | 452300 | 452300 | 391700 | 452300 |
| | Mean generated CFs for a sample | 97.70 | 100 | 100 | 100 | 100 | 100 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 86 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 135700 | 133900 | 161000 | 161300 | 156600 | 159600 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 54800 | 61000 | 97900 | 97900 | 105400 | 133400 |
| Genetic | Total generated CFs | 444888 | 440106 | 446287 | 447058 | 391700 | 452262 |
| | Mean generated CFs for a sample | 98.36 | 97.30 | 98.67 | 98.841 | 100 | 99.99 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 86 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 133033 | 131132 | 159104 | 159722 | 156700 | 159581 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 53372 | 59565 | 96486 | 96552 | 105300 | 133393 |
| Kdtree | Total generated CFs | 326152 | 287017 | 401305 | 412508 | 342077 | 452300 |
| | Mean generated CFs for a sample | 72.11 | 63.45 | 88,725 | 91,202 | 75,630 | 100 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 93032 | 77567 | 148915 | 154010 | 112764 | 159600 |
| | No. of CFs for $f(\mathbf{x}) = 0 \wedge s = f_s(\mathbf{x}) = \mathit{privileged}$ | 36479 | 34002 | 79265 | 81629 | 79214 | 133400 |

quite close to the original ones. A reason could be found for this behavior by analyzing the feature space: the more significant number of features and the generally limited range of feature values of this dataset than the Adult dataset. Indeed, the only feature with a large range of values is the credit amount, which also shows the most significant changes from the original samples, especially for the AdvDeb model. For Genetic and KDtree the distances are similar and not too different from Random. The feature with a non-constant behavior is the credit amount for both models. A common trend is not observable regarding the difference between privileged and unprivileged groups. However, most cases require more significant changes for the unprivileged group than the privileged.

### *6.3.1. Observations*

In this experimental evaluation, we got evidence that, in most cases, starting from a given sample $\mathbf{x}$, the counterfactual generation is not a problem. However, if we impose demands on the capability of reversing the decision and maintaining the counterfactual entirely close to the original sample, the generation may fail. In the second analysis, we tried to measure the sort of 'effort' required to implement changes in the original sample to get a reverse decision. Here, the outcome is not pretty straightforward. *Some methods maintain counterfactuals quite close to the actual samples, others not. Generally, unprivileged groups require more effort to reverse the decision than privileged.* Therefore, the answer to RQ3 is, most of the time, yes, *counterfactual reasoning is effective for discovering decision biases*. However, we must be aware of some exceptions mentioned above.

## *6.4. RQ4: Is our methodology effective for discovering discrimination and biases? Are there limitations in its application?*

The previous experimental evaluation have assessed that it is possible:

**Table 10**
German statistics for *gender* and *Marital Status*, for each decision-maker, i.e., LR, SVM, XGB, LGBM, AdvDeb, and LFERM, and for each Counterfactual generation strategy. Statistics include the number of generated Counterfactuals (i.e., the sum of all the Counterfactuals samples generated for all the samples), the number of test set samples with at least one Counterfactual, and the percentage of generated Counterfactuals for each sample (with respect to the required 100), the number of Counterfactuals for negatively predicted samples and correctly predicted as unprivileged or privileged and collected in the sets $\mathcal{B}$ and $\mathcal{A}$, respectively (see Algorithm 1).

(a) German statistics considering *gender* as a sensitive feature

| CF method | Statistics | Gender | | | | | |
|---|---|---|---|---|---|---|---|
| | | LR | SVM | XGB | LGBM | AdvDeb | LFERM |
| Random | Total generated CFs | 10000 | 10000 | 10000 | 10000 | 2400 | 10000 |
| | Mean generated CFs for a sample | 100 | 100 | 100 | 100 | 100 | 100 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 24 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 100 | 0 | 200 | 100 | 200 | 0 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{privileged}$ | 2500 | 1200 | 1100 | 1200 | 1100 | 200 |
| Genetic | Total generated CFs | 9993 | 9999 | 9967 | 9946 | 2400 | 10000 |
| | Mean generated CFs for a sample | 99.93 | 99.99 | 99.67 | 99.46 | 100 | 100 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 24 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 100 | 0 | 200 | 100 | 200 | 0 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{privileged}$ | 2498 | 1200 | 1100 | 1200 | 1100 | 200 |
| Kdtree | Total generated CFs | 10000 | 10000 | 10000 | 10000 | 10000 | 2224 |
| | Mean generated CFs for a sample | 100 | 100 | 100 | 100 | 100 | 22.24 |
| | Percentage of $\mathcal{X}$ with CFs (%) | 100 | 100 | 100 | 100 | 100 | 100 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{unprivileged}$ | 100 | 0 | 200 | 100 | 200 | 0 |
| | No. of CFs for $f(\mathbf{x}) = 0 \land s = f_s(\mathbf{x}) = \mathit{privileged}$ | 2500 | 1200 | 1100 | 1200 | 1100 | 200 |

- to predict sensitive features from non-sensitive ones;
- to measure the discrimination levels that predictive models (also unbiased) implement;
- to verify the possibility of generating counterfactuals (CFs) with different strategies and predictive models;
- to measure how many counterfactuals are close to the original samples.

In this last experimental evaluation, we complete our analysis by providing a further strategy for assessing the discrimination that predictive models implement. More specifically, thanks to the sensitive-feature classifier, we verify whether the generated counterfactuals that allow reversing the decision belong to the original sensitive class. If the sensitive class changes with the decision (CF flip), we are dealing with a bias problem definitively.

Fig. 6 shows the result of our analysis for the Adult dataset and the *gender* feature. On the $x$-axis, there is the number of generated counterfactuals, and on the $y$-axis, the percentage of flips. We observe a small percentage of flips with the random strategy. This behavior is probably due to the random generation of new samples (with the biggest distance from the original sample) for which the classifier does not recognize a clear pattern to change the sensitive class. On the opposite, both Genetic and KDtree get many flips. An interesting result is that flips are more significant for the unprivileged group than the privileged. This confirms that members of the unprivileged group have to show characteristics of the privileged group to reverse the decision. Accordingly, the counterfactuals for the *female* group show *male* characteristics to reach a favorable decision. The result of the KDtree strategy is very significant because, since KDtree searches counterfactuals in the real sample space, flips mean that it needs to find a sample belonging to the opposite sensitive class to reverse the decision. From the algorithm perspective, the debiased models (LFERM, AdvDeb) generally have a lower flip rate than the LR, XGB, and LGBM models. We also observe that the number of counterfactuals does not generally impact the flips. This means that by increasing the number of counterfactuals, the probability of having a flip does not grow respectively.

The results of the *marital status* on Adult are reported in Fig. 7. Even for this feature, the unprivileged group shows more flips than the privileged. Here we bring attention to LFERM. This debiased algorithm shows a behavior apparently against the general one: more flips for the privileged class than the unprivileged. However, we should remember that LFERM shows very low performance in terms of accuracy (see Table 7); thus, we can suppose that this error has been propagated in the flip count as well. This is a very relevant result in the application of our model: the bias detection based on the sensitive-feature classifier is effective only if the analyzed predictive model gets a reasonable accuracy. KDtree has the largest number of flips for the unprivileged group, confirming the need to find a sample belonging to the privileged group to reverse the decision. From other perspectives, there are no significant outcomes to be noticed that have not already been discussed for the gender feature.

Fig. 8 presents the results for the *gender* feature on the German dataset. At first glance, we observe a more significant number of flips for the unprivileged group than the privileged for the Adult dataset. From the algorithm perspective, there is a good coherence between fairness metrics and the number of flips, particularly for LGBM, XGB, and AdvDeb. LFERM and SVM do not have *unprivileged* samples with a negative outcome. Thus, no flips occur. Increasing the number of counterfactuals does not proportionally increase the number of flips. This means that the model is effective even with a quite small ($< 20$) number of counterfactuals. Again, KDtree has the most significant number of flips that, further confirms the need to change the sensitive feature to get a beneficial prediction.

### *6.4.1. Observations*

Through this last experimental evaluation, we confirmed the effectiveness of the proposed methodology for discovering discrimination and decision biases. *The results of the proposed approach are generally coherent with fairness metrics. In contrast to them*

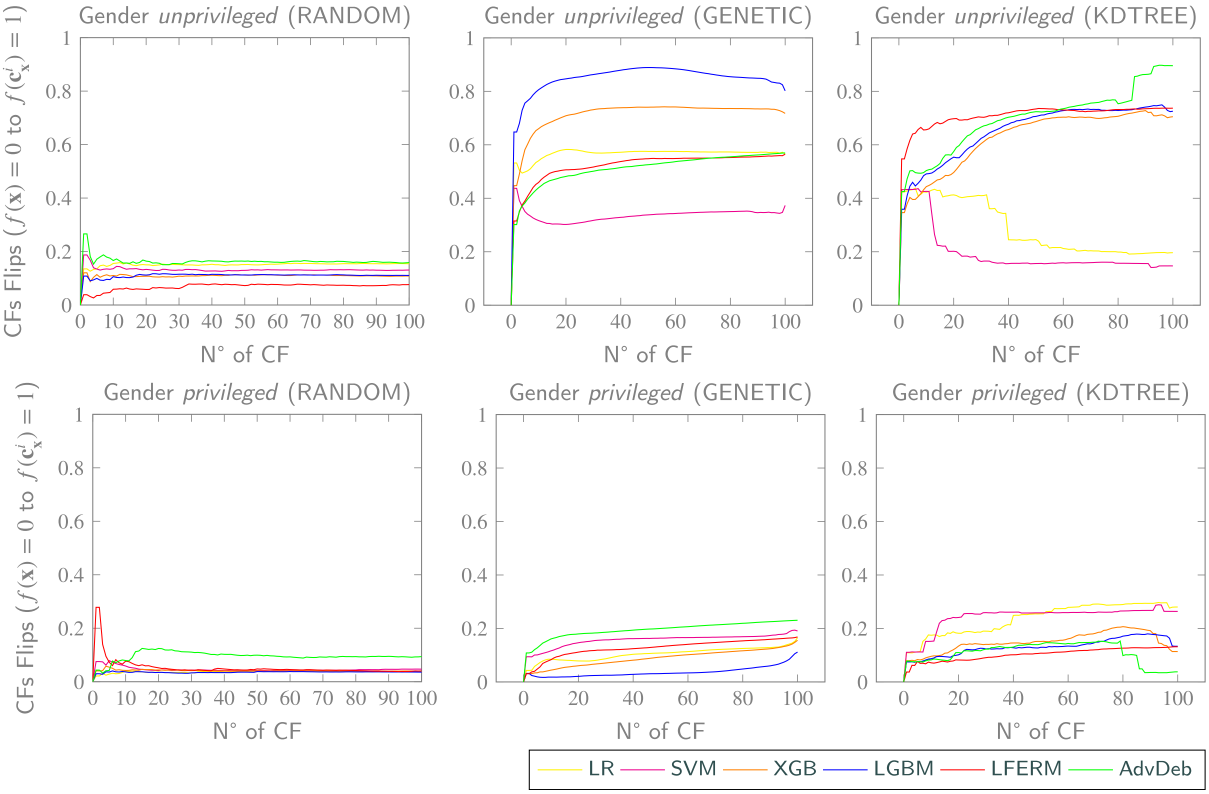

**Fig. 6.** Ratio of counterfactuals for the German dataset that are identified as belonging to another gender (w.r.t. the original sample). The first row analyzes counterfactuals for the *unprivileged gender* group, i.e., *female* group in the German dataset. The second row analyzes counterfactuals for the *privileged gender* group, i.e., *Male* group. The plot includes all the decision-makers for the three counterfactual generation strategies and an increasing number of considered counterfactuals.

*which provide an overall value, the proposed model deeply analyzes the characteristics that the user should have to reverse the decision. However, there is an obvious limitation in the application of the model: the decision maker must have good predictive accuracy. Another interesting result is that a quite small* ($< 20$) *number of counterfactuals is enough for discovering biases.*

## 7. Limitations and future work

Our work proposes a new methodology for exploring and investigating bias by exploiting advances in counterfactual reasoning. Even though the outcomes presented are a notable achievement in bias identification, our work is not exempt from limitations. For instance, Section 6.3 explores the distances between counterfactuals classified as privileged and underprivileged. However, an overall distance does not highlight features that are the most important in the decision-making process and are, at the same time, proxy features (i.e., $\mathbf{p} \subseteq \mathbf{x}$). A future version of the system could integrate a feature importance methodology like SHAP to address this limitation and bring these features out. This future system should combine sensitive feature classifier outcomes and SHAP values. However, it has yet to be possible to identify these proxy features clearly. Identifying the hidden proxy features is a key objective of our future work and deserves a specific investigation. Further limitations include (i) the strategy for generating counterfactuals, (ii) the quality of sensitive feature models, and (iii) the validation metrics. In this investigation, we relied on existing counterfactual generation strategies, thus keeping the details outside the paper's scope. However, we noticed that some strategies seldom succeed in generating counterfactuals; therefore, in future work, investigating other counterfactual generation models will be necessary. Another frustrating shortcoming is due to the pair dataset/sensitive feature classifier. The most used datasets, like German, are very small, impacting the quality of the trained classifier. Unfortunately, the sensitive feature classifier is a critical component of our system, and its high accuracy is crucial for correctly investigating the bias. Future work will focus on other domains that could grant larger datasets even though they are not datasets of reference for fairness and bias research. Furthermore, the choice of the objective to maximize/minimize – and to select the best models – deserves a specific investigation. Since this kind of study would have fallen in the scope of multi-objective optimization, we avoided facing this aspect in this work. Instead, we prioritized a better separability between positive and negative samples for the decision-maker through AUC. For the sensitive feature classifier, we preferred using

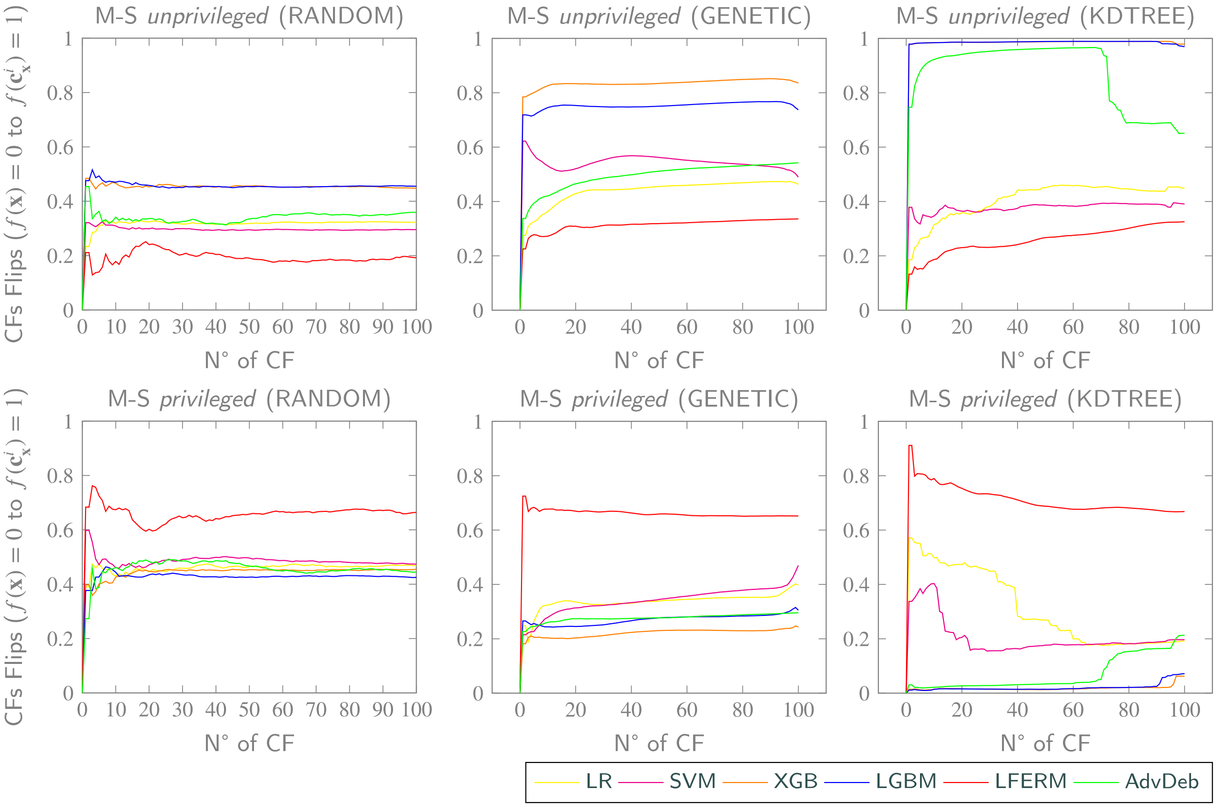

**Fig. 7.** Ratio of counterfactuals for the Adult dataset that are identified as belonging to another marital-status (w.r.t. the original sample). The first row analyzes counterfactuals for the *unprivileged marital-status* group, i.e., *not married* group in the Adult dataset. The second row analyzes counterfactuals for the *privileged marital-status* group, i.e., *married* group. The plot includes all the decision-makers for the three counterfactual generation strategies and an increasing number of considered counterfactuals.

F1 to avoid specialization on privileged or non-privileged classes and thus balance the predictions between the classes. The debiased decision-makers are a special case, where we tuned the models by optimizing AUC and minimizing DAO through the proposed metric (see Eq. (15)). The choice of DAO is dictated by the will to consider not only true positives (e.g., DEO) but also the true negatives. In a future investigation, we would like to study the impact of adopting other fairness – and, in general, other validation – metrics. Finally, the discussion regarding the validation metrics introduces a further limitation of the study. Indeed, recent literature explored a wide range of metrics for accuracy, beyond-accuracy, fairness, and bias. In this study, we limited our analysis to some well-known and mainstream metrics. However, several other metrics would shed light on the various aspects and behavior of the models. Future investigations will explore these dimensions.

## 8. Conclusion

This study introduces a novel methodology for detecting and assessing biases in decision-making models, even if they operate in the context of "fairness under unawareness", and thus do not use sensitive features. The role of counterfactual reasoning in the proposed approach is crucial. Adopting counterfactual reasoning in the proposed approach is crucial since it allows unveiling the characteristics of original samples that could reverse the decision-makers prediction. When the counterfactual sample is identified as a different demographic group (compared to the original sample), it could be a sign of discriminatory behavior, and we refer to it as a counterfactual flip. We tested this counterfactual approach to detect bias with two state-of-the-art datasets for two financial domain tasks: predicting loan-repayment default and individual income.

The experimental results show that the "fairness under unawareness" setting is insufficient to mitigate bias due to proxy features. Moreover, the results confirmed that the proposed approach is effective for auditing bias. The proposed model complements the state-of-the-art statistical metrics commonly adopted to evaluate the fairness level. In fact, the approach analyzes the characteristics that a given sample should have to achieve a positive outcome, such as receiving a loan or other life-changing decisions. Furthermore, the results show that even debiased algorithms are not enough to avoid discriminatory behaviors completely.

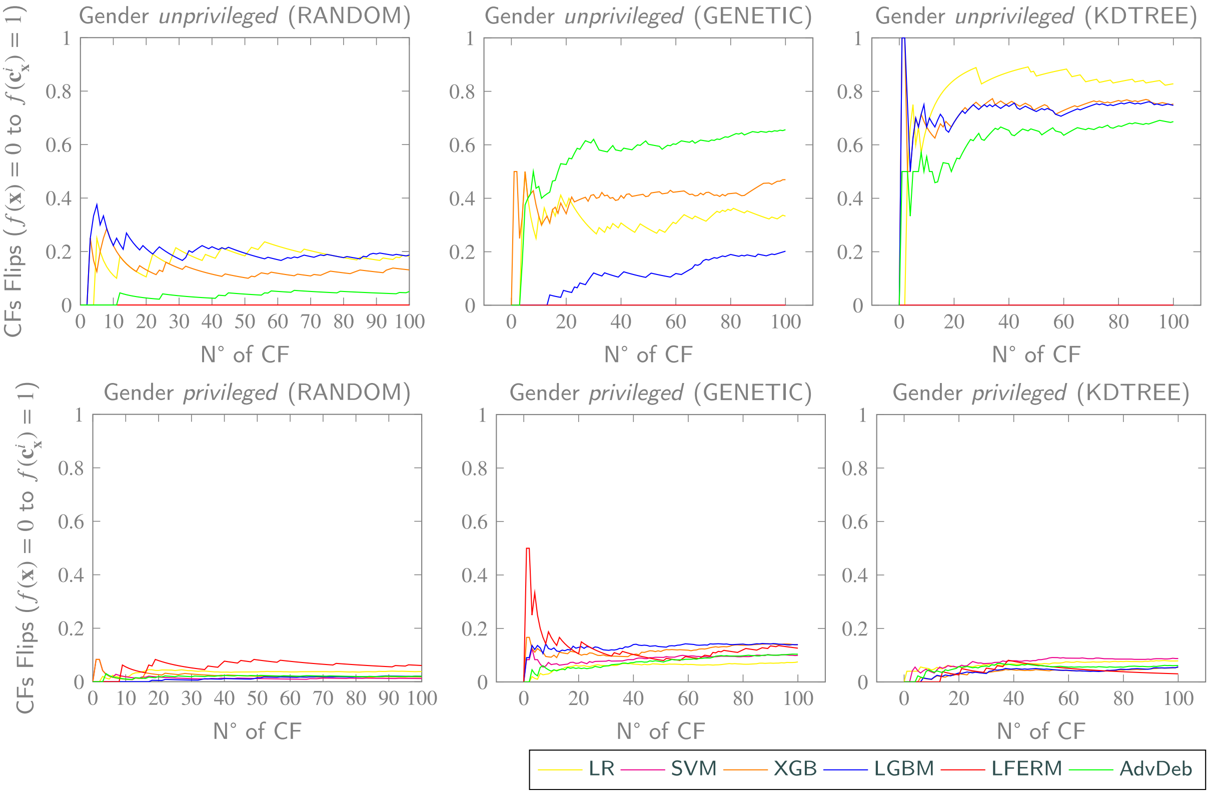

**Fig. 8.** Ratio of counterfactuals for the German dataset that are identified as belonging to another gender (w.r.t. the original sample). The first row analyzes counterfactuals for the *unprivileged gender* group, i.e., *female* group in the German dataset. The second row analyzes counterfactuals for the *privileged gender* group, i.e., *Male* group. The plot includes all the decision-makers for the three counterfactual generation strategies and an increasing number of considered counterfactuals.

Lastly, this investigation paves the way for the integration of counterfactual reasoning with fairness research. The insights that emerged in the study gave us the idea that we have just scratched the surface of the potential of applying counterfactual reasoning to tasks that impact user lives critically.

## CRediT authorship contribution statement

**Giandomenico Cornacchia:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Software. **Vito Walter Anelli:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing. **Giovanni Maria Biancofiore:** Conceptualization, Methodology, Writing – original draft. **Fedelucio Narducci:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing. **Claudio Pomo:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Software. **Azzurra Ragone:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing. **Eugenio Di Sciascio:** Conceptualization, Methodology, Writing – original draft, Supervision.

## Data availability

Data will be made available on request.

## Acknowledgments

This research was partially supported by the following projects: VHRWPD7 – CUP B97I19000980007 – COR 1462424 ERP 4.0, Grant Agreement Number 101016956 H2020 PASSEPARTOUT, Secure Safe Apulia, Codice Pratica 3PDW2R7 SERVIZI LOCALI 2.0, MISE CUP: I14E20000020001 CTEMT - Casa delle Tecnologie Emergenti Comune di Matera, PON ARS01_00876 BIO-D, CT_FINCONS_II.

## References

Agarwal, S., & Mishra, S. (2021). *Responsible AI*. Springer.

Ashokan, A., & Haas, C. (2021). Fairness metrics and bias mitigation strategies for rating predictions. *Information Processing & Management*, *58*(5), Article 102646.

Balunovic, M., Ruoss, A., & Vechev, M. T. (2022). Fair normalizing flows. In *The tenth international conference on learning representations, ICLR 2022, virtual event, April 25-29, 2022*. OpenReview.net, URL https://openreview.net/forum?id=BrFIKuxrZE.

Bickel, P. J., Hammel, E. A., & O'Connell, J. W. (1975). Sex bias in graduate admissions: Data from berkeley. *Science*, *187*(4175), 398–404. http://dx.doi.org/10.1126/science.187.4175.398, URL https://www.science.org/doi/abs/10.1126/science.187.4175.398. arXiv:https://www.science.org/doi/pdf/10.1126/science.187.4175.398.

Biswas, A., & Mukherjee, S. (2021). Ensuring fairness under prior probability shifts. In M. Fourcade, B. Kuipers, S. Lazar, & D. K. Mulligan (Eds.), *AIES '21: AAAI/ACM conference on AI, ethics, and society, virtual event, USA, May 19-21, 2021* (pp. 414–424). ACM, http://dx.doi.org/10.1145/3461702.3462596.

Boser, B. E., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *COLT* (pp. 144–152). ACM.

Bottou, L., Peters, J., Candela, J. Q., Charles, D. X., Chickering, M., Portugaly, E., et al. (2013). Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, *14*(1), 3207–3260.

Bureau, C. F. P. (2014). Using publicly available information to proxy for unidentified race and ethnicity: A methodology and assessment. URL https://www.consumerfinance.gov/data-research/research-reports/usingpublicly-available-information-to-proxy-for-unidentified-race-and-ethnicity/.

Calders, T., & Jaroszewicz, S. (2007). Efficient AUC optimization for classification. In J. N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič, & A. Skowron (Eds.), *Knowledge discovery in databases: PKDD 2007* (pp. 42–53). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, J. (2018). Fair lending needs explainable models for responsible recommendation. In *FATREC'18 proceedings of the second workshop on responsible recommendation*. Vancouver, British Columbia, Canada: arXiv:1809.04684.

Chen, J., Kallus, N., Mao, X., Svacha, G., & Udell, M. (2019). Fairness under unawareness: Assessing disparity when protected class is unobserved. In *FAT* (pp. 339–348). ACM.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017). Algorithmic decision making and the cost of fairness. In *KDD* (pp. 797–806). ACM.

Cornacchia, G., Narducci, F., & Ragone, A. (2021a). A general model for fair and explainable recommendation in the loan domain (short paper). In *CEUR workshop proceedings*: *vol. 2960*, *KaRS/ComplexRec@RecSys*. CEUR-WS.org.

Cornacchia, G., Narducci, F., & Ragone, A. (2021b). Improving the user experience and the trustworthiness of financial services. In *Lecture notes in computer science*: *12936*, *INTERACT (5)* (pp. 264–269). Springer.

Das, S., Donini, M., Gelman, J., Haas, K., Hardt, M., Katzman, J., et al. (2021). Fairness measures for machine learning in finance. *The Journal of Financial Data Science*, *3*(4), 33–64. http://dx.doi.org/10.3905/jfds.2021.1.075, URL https://jfds.pm-research.com/content/3/4/33, arXiv:https://jfds.pm-research.com/content/3/4/33.full.pdf.

Deldjoo, Y., Jannach, D., Bellogin, A., Difonzo, A., & Zanzonelli, D. (2022). A survey of research on fair recommender systems. arXiv preprint arXiv:2205.11127.

DeMartino, G. F. (2020). The confounding problem of the counterfactual in economic explanation. *Review of Social Economy*, 1–11.

Denton, E., Hutchinson, B., Mitchell, M., Gebru, T., & Zaldivar, A. (2019). Image counterfactual sensitivity analysis for detecting unintended bias. arXiv preprint arXiv:1906.06439.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J., & Pontil, M. (2018). Empirical risk minimization under fairness constraints. In *NeurIPS* (pp. 2796–2806).

Dressel, J., & Farid, H. (2018). The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, *4*(1), eaao5580. http://dx.doi.org/10.1126/sciadv.aao5580, URL https://www.science.org/doi/abs/10.1126/sciadv.aao5580, arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.aao5580.

Dudík, M., Langford, J., & Li, L. (2011). Doubly robust policy evaluation and learning. In *ICML* (pp. 1097–1104). Omni Press.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference* (pp. 214–226).

Ekstrand, M. D., Das, A., Burke, R., Diaz, F., et al. (2022). Fairness in information access systems. *Foundations and Trends® in Information Retrieval*, *16*(1–2), 1–177.

Elliott, M., Fremont, A., Morrison, P., Pantoja, P., & Lurie, N. (2008). A new method for estimating race/ethnicity and associated disparities where administrative records lack self-reported race/ethnicity. *Health Services Research*, *43*, 1722–1736. http://dx.doi.org/10.1111/j.1475-6773.2008.00854.x.

Elliott, M. N., Morrison, P. A., Fremont, A. M., McCaffrey, D. F., Pantoja, P. M., & Lurie, N. (2009). Using the census bureau's surname list to improve estimates of race/ethnicity and associated disparities. *Health Services and Outcomes Research Methodology*, *9*, 69–83.

Fabris, A., Esuli, A., Moreo, A., & Sebastiani, F. (2021). Measuring fairness under unawareness via quantification. *CoRR*, https://arxiv.org/abs/2109.08549, arXiv:2109.08549.

Ferrario, R. (2001). Counterfactual reasoning. In *Lecture notes in computer science*: *2116*, *CONTEXT* (pp. 170–183). Springer.

Ginsberg, M. L. (1986). Counterfactuals. *Artificial Intelligence*, *30*(1), 35–79.

Gómez, E., Boratto, L., & Salamó, M. (2022). Provider fairness across continents in collaborative recommender systems. *Information Processing & Management*, *59*(1), Article 102719.

Guntzel, M. H. (2022). Fairness in machine learning: An empirical experiment about protected features and their implications.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. In *NIPS* (pp. 3315–3323).

Joo, J., & Kärkkäinen, K. (2020). Gender slopes: Counterfactual fairness for computer vision models by attribute manipulation. In *Proceedings of the 2nd international workshop on fairness, accountability, transparency and ethics in multimedia* (pp. 1–5).

Korikov, A., Shleyfman, A., & Beck, J. C. (2021). Counterfactual explanations for optimization-based decisions in the context of the GDPR. In *IJCAI* (pp. 4097–4103). ijcai.org.

Kulesza, A., & Taskar, B. (2012). Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, *5*(2–3), 123–286.

Kusner, M. J., Loftus, J. R., Russell, C., & Silva, R. (2017). Counterfactual fairness. In *NIPS* (pp. 4066–4076).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*(6), 1–35.

Mikolajczyk, A., Grochowski, M., & Kwasigroch, A. (2021). Towards explainable classifiers using the counterfactual approach - global explanations for discovering bias in data. *Journal of Artificial Intelligence and Soft Computing Research*, *11*(1), 51–67.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38.

Mishler, A., Kennedy, E. H., & Chouldechova, A. (2021). Fairness in risk assessment instruments: Post-processing to achieve counterfactual equalized odds. In *FAccT* (pp. 386–400). ACM.

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* '20, Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 607–617). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3351095.3372850.

Oneto, L., & Chiappa, S. (2020). Fairness in machine learning. *CoRR*, arXiv:2012.15816.

Panigutti, C., Perotti, A., Panisson, A., Bajardi, P., & Pedreschi, D. (2021). FairLens: Auditing black-box clinical decision support systems. *Information Processing & Management*, *58*(5), Article 102657.

Pearl, J. (1994). Causation, action and counterfactuals. In *ECAI* (pp. 826–828). John Wiley and Sons, Chichester.

Pedreschi, D., Ruggieri, S., & Turini, F. (2008). Discrimination-aware data mining. In *KDD* (pp. 560–568). ACM.

Pfohl, S. R., Duan, T., Ding, D. Y., & Shah, N. H. (2019). Counterfactual reasoning for fair clinical risk prediction. In *Proceedings of machine learning research*: *106*, *MLHC* (pp. 325–358). PMLR.

Pitoura, E., Stefanidis, K., & Koutrika, G. (2022). Fairness in rankings and recommendations: An overview. *VLDB Journal*, *31*(3), 431–458.

Rossini, D., Croce, D., Mancini, S., Pellegrino, M., & Basili, R. (2020). Actionable ethics through neural learning. In *AAAI* (pp. 5537–5544). AAAI Press.

Ruf, B., & Detyniecki, M. (2020). Active fairness instead of unawareness. *CoRR*, arXiv:2009.06251.

Sokol, K., & Flach, P. A. (2019). Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety. In *CEUR workshop proceedings*: *vol. 2301*, *SafeAI@AAAI*. CEUR-WS.org.

Swaminathan, A., & Joachims, T. (2015). Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research*, *16*, 1731–1755.

Tavakol, M. (2020). Fair classification with counterfactual learning. In *SIGIR* (pp. 2073–2076). ACM.

Yeom, S., Datta, A., & Fredrikson, M. (2018). Hunting for discriminatory proxies in linear regression models. *Advances in Neural Information Processing Systems*, *31*.

Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. In *JMLR workshop and conference proceedings*: *vol. 28*, *ICML (3)* (pp. 325–333). JMLR.org.

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society* (pp. 335–340). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3278721.3278779.

Zhu, Z., Hu, X., & Caverlee, J. (2018). Fairness-aware tensor-based recommendation. In *CIKM* (pp. 1153–1162). ACM.