Managing Bias in Machine Learning Projects

Tobias Fahse
Universität St.Gallen

Viktora Huber
Universität St.Gallen

Benjamin van Giffen
Universität St.Gallen

Managing Bias in Machine Learning Projects

Tobias Fahse[1], Viktoria Huber[1], and Benjamin van Giffen[1]

[1] University of St.Gallen, Institute of Information Management, St. Gallen, Switzerland
{tobias.fahse,benjamin.vangiffen}@unisg.ch, viktoria.huber@student.unisg.ch

Abstract. This paper introduces a framework for managing bias in machine learning (ML) projects. When ML-capabilities are used for decision making, they frequently affect the lives of many people. However, bias can lead to low model performance and misguided business decisions, resulting in fatal financial, social, and reputational impacts. This framework provides an overview of potential biases and corresponding mitigation methods for each phase of the well-established process model CRISP-DM. Eight distinct types of biases and 25 mitigation methods were identified through a literature review and allocated to six phases of the reference model in a synthesized way. Furthermore, some biases are mitigated in different phases as they occur. Our framework helps to create clarity in these multiple relationships, thus assisting project managers in avoiding biased ML-outcomes.

Keywords: Bias, Machine Learning, Project Management, Risk Management, Process Model

1 Introduction

Progress in artificial intelligent (AI) technologies such as machine learning (ML) lead to a wide implementation of intelligent systems in companies and institutions. The ability to learn and act autonomously makes AI different from other technologies and allows for automated decisions and solutions [1]. ML, as a field of AI, refers to algorithms that learn patterns from data without being explicitly programmed [2]. ML-applications support or take over human tasks and decisions in many industries, including issuing of credit loans, determination of insurance rates or provision of health care [3]. Simultaneously, the potential of AI is widely recognized, but there remains a significant uncertainty for organizations in how to manage negative consequences and challenges resulting from AI [4]. Due to the increasing complexity of AI, their usage can lead to negative consequences such as wrong decisions, unfairness, and discrimination [5, 6]. If firms cannot understand the underlying mechanisms of their ML-models, organizations face a loss of trust in their technologies [7] leading to questioning of their accountability and reliability and, in the long term, impact investments into AI in organizations [5, 8]. Many of these obstacles can arise from bias incorporated in the ML developing process [9].

16th International Conference on Wirtschaftsinformatik,
March 2021, Essen, Germany

Today, IS research provides only little theory and guidance to systematically identify and mitigate different forms of bias that can occur in ML-projects. There is a need for awareness about possible biases in the ML-project lifecycle and respective mitigation methods to tackle negative consequences [10]. A systematic approach for addressing potential biases is yet missing [11]. Therefore, the objective of this research is to systematically review and integrate available knowledge to improve the management of bias in ML-projects. To frame our research, we pose the following question:

RQ: What types of bias emerge in machine learning projects and how can they be managed?

We perform a systematic literature review to identify different types of biases and respective mitigation methods. We then integrate our findings into CRISP-DM, an established ML-project management framework. CRISP-DM is the most widely adopted process model [12, 13] and provides a practice-oriented and structured approach, including consecutive activities for developing ML-applications that are transferable to different industries [14].

Our research contributes to the emerging body of literature that studies the implementation of ML in organizations and its intended and unintended consequences. First, this paper enhances clarity concerning existing biases and mitigation methods. Hence, a shared terminology can be achieved, which supports further development of solutions. Second, the holistic view of ML-model development and operation improves the understanding of inherent connections between ML-project and bias. Furthermore, temporal dependencies between bias occurrence and mitigation can be detected. Thus, interdependencies in ML-applications that are embedded in broader organizational information systems can be recognized. This makes the management of bias in ML-projects an important aspect of IS development that differs from the development of deterministic software systems.

Our framework serves practitioners as a holistic framework which can be applied to the specific ML-projects at hand.

2 Theoretical Foundation

The adoption of ML-technologies across organizations is growing rapidly and firms are increasingly recognizing the practical opportunities arising from their ability to perform human-like tasks such as learning autonomously, making decisions, or gaining valuable analytical insights from large datasets [1, 4]. In this context, bias describes an unintended or potentially harmful property of data [15] that results in a systematic deviation of algorithmic results [16]. In a broader sense, bias can be defined as unwanted effects or results which are evoked through a series of subjective choices and practices that the ML developing process involves [15].

ML-algorithms differ substantially from deterministic, rule-based algorithms that have been used in the past to perform decision support in organizational context. ML-algorithms, such as Neural Networks, follow a probabilistic approach in which decisions are not made by following programmed rules but by learning patterns from

historical data and applying these to new input data. The decision support from ML-algorithms is provided in the form of probabilities, leading to different levels of uncertainty and therefore increased susceptibility to systematic biases. As the learned patterns are non-transparent to the users of the algorithms, existing bias is difficult to identify or mitigate and therefore requires different approaches than deterministic algorithms [17].

A series of potentially subjective choices and actions must be made in the process of an ML-project, any of which can introduce bias and lead to unwanted effects. If datasets incorporate bias, ML-applications will reflect those biases. Even if input data is perfectly unbiased, the decision on how to build the model can introduce bias. Particularly from technical considerations, design decisions must be made, such as which fairness definition or forms of measurement and performance metrics to use. Even if assuming the resulting ML-application is free from bias introduced through data or design decisions, an inappropriate context of use may nevertheless lead to bias [16, 18, 19].

To address the problem of bias, prior work has used fairness metrics as quantification of unwanted bias. This raises concerns about how social goals are abstracted so that they can be used in a prediction task [20]. Such metrics make the implicit assumption that an underlying mathematical concept of fairness can be formulated and operationalized to create a bias-free system [21, 22]. However, there are at least 21 different definitions of fairness [23], and different definitions lead to entirely different results, which makes it impossible to satisfy all fairness definitions simultaneously [24]. Corbett-Davies and Goel [25] and Mitchell et al. [20] demonstrate how popular classes of fairness definitions can, perversely, have a discriminant effect on minorities and also majorities. To prevent this, technical approaches should be complemented with non-technical mitigation approaches that consider more than merely good performance results.

Prior work has emphasized the importance of interpretable outcomes to increase fairness in ML-applications. By increasing the interpretability of outcomes, harmful patterns that have been learned by the ML-algorithm can be revealed and consequently be tackled [7].

IS research is starting to address the negative consequences resulting from bias but the literature about the origins of biases and possible mitigation methods is yet scattered and a systematic approach for addressing potential biases missing [11]. Few existing articles provide a framework with incorporated biases and mitigation methods where the respective terminologies differ substantially and the incorporation is not made in well-established ML-frameworks [15, 16, 26]. To address this, we choose CRISP-DM as an underlying framework for this work as it is the most widely adopted process model [12, 13]. It provides a practice-oriented and structured approach, including consecutive activities for developing ML-applications that are transferable to different industries.

The initial phase of CRISP-DM, Business Understanding (BU), focuses on the understanding of the business objective and translation into data mining goals in order to define the design plan and necessary resources. The Data Understanding (DU) phase then collects and explores initial data to gain insights into data quality and possible concerns. The final dataset is then created from the raw dataset through various

activities in the Data Preparation (DP) phase, such as the selection of records and features or transformation and cleaning of data for the modeling tools. Several modeling techniques are then selected in the Modeling (MO) phase and applied to the prepared dataset. The model's performance is evaluated in the Evaluation (EV) phase and put into the context of the business objectives. The Deployment (DE) phase then describes the process of implementing the model in the context of the end-user [14].

Overall, a systematic approach for managing potential biases with a shared terminology and clarity about their occurrence in ML-project lifecycles is yet missing. The challenge is to understand what biases potentially arise in which phase and when to apply which prevention or mitigation method [27]. This underlines the need for guidance to identify sources of harm throughout the full ML-project lifecycle.

3 Methodology

3.1 Research Design

The present paper aims to examine how project managers can systematically identify and mitigate bias in ML-projects. To this end, we perform a systematic literature review to understand existing knowledge about bias through a conceptual lens (CRISP-DM) proposed in former research [28, 29]. This type of literature review is problem-centered and aims to group distinctive problem sources as well as different solution approaches using well-known concepts [29, 30]. First, different types of biases are identified, their terminologies understood and consolidated into distinct categories. Second, possible mitigation methods that address these biases are evaluated. Finally, the findings are incorporated into the CRISP-DM framework regarding their occurrence and application in different project steps.

3.2 Data Collection and Analysis

We determined a threefold data collection strategy. First, a keyword search was conducted considering only leading journals in the field of Information Systems (IS). It focused on 39 journals ranked A+, A, or B based on JourQual3 ranking that are represented in the databases EBSCOhost, AIS Electronic Library, ACM Digital Library, and ScienceDirect. We defined relevant key terms by conceptualizing the topic based on an initial search in IS literature and seminal publications on bias in ML. This resulted in the following search strings: AB ("machine bias" OR "data bias" OR "algorithmic bias" OR "model bias" OR "biased data" OR "biased machine*" OR "biased algorithm*" OR "biased model" OR "bias" or "prejudice" OR "discrimination" OR "stereotypes" OR "historical bias" OR "representation bias" OR "selection bias" OR "measurement bias" OR "aggregation bias" OR "deployment bias" AND TX ("Artificial Intelligence" OR "AI" OR "Algorithm*" OR "Machine Learning")); TX ("machine bias" OR "data bias" OR "algorithmic bias" OR "model bias") AND TX ("artificial intelligence" OR "AI" OR "machine learning" OR "deep learning"). These search strings revealed 57 hits in total, including the initial search. We then conducted

a practical screening with inclusion and exclusion criteria. We only considered articles that (i) are peer-reviewed and/or conference proceedings, (ii) define at least one relevant bias (iii) explain at least one relevant identification or mitigation method (iv) address the bias challenge in AI-, or ML-context. After applying these criteria, 22 relevant hits remained. As a third step we performed forward and backward search on the relevant hits using Web of Science [31, 32]. This step added 31 articles based on the same inclusion/exclusion criteria. In a third step, relevant literature was exchanged with senior scholars in the research team [33]. In total, 55 articles were included in the literature review and subsequently examined full text.

In the data analysis phase, types of biases and mitigation methods were subsequently extracted from the articles. Because different synonyms exist in the literature for the same type of bias, biases were then descriptively synthesized based on their mechanism [34]. This allowed us to code the biases in eight distinct categories. Based on an in-depth understanding of the CRISP-DM project steps, the distinct biases were assigned to the phases by matching the mechanism that leads to a specific bias to the tasks of the phases of CRISP-DM. If a different process model was used in the underlying paper, we matched the phases with CRISP DM. We then identified 25 methods and allocated them to the respective bias they address regarding the methods' mechanisms. Finally, the methods were incorporated into the CRISP-DM model steps according to their detailed specifications of where the methods can be applied in the project lifecycle. The allocation was independently conducted by two researchers to enhance inter-coder reliability [35].

4 Results

4.1 Emergence of Bias in Machine Learning Projects

Bias can occur multiple times and in any of the six phases of a ML-project lifecycle. The eight identified biases are located in the CRISP-DM model based on their origin (see Figure 1) and explained and illustrated in the following.

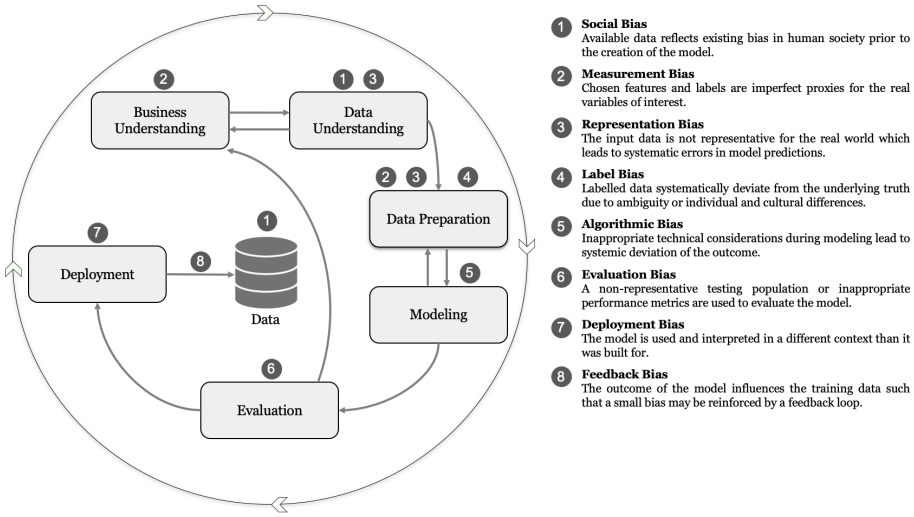

Figure 1. Emergence of Bias in CRISP-DM Process Model (Chapman et al. 2000)

Social bias occurs when available data mirrors existing biases in society at large. When data embodies social biases that exist independently and prior to the development of an ML-model, the model will most likely lead to unwanted outcomes. Even if the data is perfectly measured and sampled, a normative concern with the current state of the world may exist, that should not be reinforced by the ML-model [9, 19, 33, 36].

Illustration: In 2018, statistical evaluations revealed that only 5% of the Fortune 500 companies' CEOs were women [37]. This unequal distribution was consequently reflected in Google's image search of CEOs that showed only a small fraction of women. Google has recently adapted the search results on images of CEOs showing a higher proportion of women in order not to reinforce gender inequality [15].

Measurement bias can be introduced by humans in the BU-phase through subjective choices about model design. When defining the target variable and necessary features for the data mining problem, they may choose imperfect proxies for the true underlying value or include protected attributes. Protected attributes refer to attributes such as race, gender, ethnicity, etc. that partition a population into different groups that should be treated equally. Using protected attributes as proxies for other features that truly carry the signal of interest may result in a discriminant or inaccurate classifier. But even if the protected attribute is excluded, the discriminant effect can still exist due to the redlining effect. The redlining effect states that protected attributes can correlate with non-protected attributes [9, 15, 25, 38]. Measurement bias can also occur in the DP-phase when features and outcome variables are created. Often, features have to be constructed, derived, or transformed where they may omit important factors or introduce additional noise. When features are inaccurate or if the decision is reduced to only a small number of inappropriate features, the prediction accuracy may vary across groups [9, 38].

Illustration: In a crime prediction application, the feature "number of arrests" is used to predict future criminal activity. Assuming African American and Caucasian defendants commit the same number of drug sales, they have a similar true risk. But arrest rates are possibly recorded differently across ethnic groups, leading to differential predictive power of the application. In minority neighborhoods with heavier policing, African American defendants are likely to have more drug arrests [39]. Despite the similar true risk, the ML-application would consequently rate the African American a higher risk than the Caucasian [40].

Representation bias arises during collection and sampling of data. It emerges if the probability distribution of the development population differs from the true underlying distribution. The algorithm subsequently fails to make good predictions for this group of feature values. The over- or underrepresentation can have several reasons, including difficult or expensive availability of required data. Subsequently, the model will result in less robust outcomes for different subpopulations [9, 15, 41]. Representation bias can also occur when the training data is no longer representative of the data found present when the model is deployed. It arises when the world as it is at the time the application is used is inconsistent with the world as it was when the training data was collected [36].

Illustration: Data can be traumatized by one-time phenomena. An algorithm built for credit card applications uses historical data about the chance of default. In case of an unsuspected event during the collection of data, such as a natural catastrophe in a certain area, people might not be able to pay back their debts. Therefore, applicants from this area will most likely be classified as potential defaults. Thus, the one-time phenomenon is imprinted into the ML-application [36].

Label bias arises when training data is assigned to class labels in the *DP-phase*. Data scientists often face the difficulty of deciding which available label best applies to the present data. Due to ambiguity, cultural or individual differences, labels might systematically deviate. Existing class labels may also fail to precisely capture meaningful differences between classes [9, 36].

Illustration: Assuming a certain number of pictures are to be labeled as "wedding". A person that is educated in the western culture, will likely only label pictures with brides in white dresses and grooms in dark suits as "wedding". An Indian wedding with its colorful dresses and special decorations might then not be labeled as a wedding [36].

Algorithmic bias is introduced during the *MO-phase* and results from inappropriate technical considerations. It can emerge when formulating the optimization problem, in which developers make data and parameters amenable to computers [18, 21, 42]. Resulting ML-models may fail to treat groups fairly under given conditions. The probability of misclassification, i.e., false-positive and false-negative rates, should be equal among groups [19, 27, 38, 41].

Illustration: In COMPAS, a predictive policing application to assess the "risk of crime recidivism", minorities exhibited a higher false-positive rate than majority groups [40, 43].

Evaluation bias can occur, if the population in the benchmark set is not representative of the use population. An algorithmic model is trained and optimized on its training data but evaluated on a test-or benchmark data set in the *EV-phase*.

ML-models are often tested on the same benchmark to allow for an objective comparison. If the benchmark itself is not representative, models could be preferred that perform only well on a subset of the population [15, 42].

Illustration: Choosing the wrong benchmark set can lead to overlooking a potential bias. For example, if a facial recognition algorithm is trained on a dataset with underrepresented dark-skinned females and is tested on a similarly unbalanced benchmark, the bias will remain unrecognized [15].

Deployment bias arises when the system is used or interpreted in inappropriate ways, even if none of the before-mentioned biases are present. This can occur when the ML-application is built and evaluated assuming it operates fully autonomous, but in reality, it works in a complex socio-technical environment and is followed by human decisions. The assumed use population may differ in a significant dimension from the actual use population. They may have a different knowledge base or values and interpret the algorithmic output according to their internalized biases [15, 27, 43].

Illustration: Risk assessment applications are models that aim to predict the likelihood of someone committing a future crime. However, in practice, these models are often seen to be used in different contexts, such as determining the length of defendants sentences [44].

Feedback Bias can arise after the *DE-phase* and after project deployment. It emerges when the output of the ML-application influences features that are used as new inputs and algorithms are refined over time (e.g., through re-training). If the outcome of the ML-application has an influence on the training data, an initially small bias is potentially reinforced through a feedback loop [9, 27, 45].

Illustration: Once a certain content got a good ranking by a rating algorithm based on the number of times it has been clicked, it will affect the position and the promotion of this content, thus leading to even more clicks. A reinforcing feedback loop is created and can lead to decreased user satisfaction as not the best content is promoted to the web user [16, 36].

4.2 Managing Bias in Machine Learning Projects

This section outlines potential mitigation methods for addressing the aforementioned biases and indicates their application within the CRISP-DM project phases. Figure 2 illustrates that a single bias can be mitigated by several methods, and one method can mitigate multiple biases. The methods presented either support project teams in identifying biases or mitigate their unwanted effects. A method that is applied in one phase can address biases that occur in the respective phase or possibly also in later stages of the project. In order to optimally avoid negative consequences resulting from bias, a sociotechnical approach is fostered by including technical and non-technical methods.

CRISP-DM Phase / Bias	Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Social Bias			Rapid Prototyping Reweighing Data Massaging Disparate Impact Remover Learning Fair Representation Optimized Pre-Processing	Prejudice Remover Adversarial Debiasing Equalized Odds Multiple Models Latent Variable Model Model Interpretability		
Measurement Bias	Diversity in Teams Exchange with Domain Experts	Proxy Estimation	Rapid Prototyping			
Representation Bias	Diversity in Teams	Data Plotting Exchange with Domain Experts	Reweighing Data Augmentation	Model Interpretability		
Label Bias		Exchange with Domain Experts	Data Massaging			
Algorithmic Bias			Rapid Prototyping	Exchange with Domain Experts Resampling Model Interpretability Multitask Learning		
Evaluation Bias				Resampling	Representative Benchmark Subgroup Validity Data Augmentation	
Deployment Bias	Diversity in Teams Consequences in Context		Rapid Prototyping			Monitoring Plan Human Supervision
Feedback Bias						Human Supervision Randomness

Figure 2. Bias Prevention and Mitigation Methods in ML-Project Phases

In the *BU-phase*, the emergence of three biases can be prevented by understanding the business objectives and undertaking actions to ensure a precise translation into data mining problems. Three bias mitigation approaches are relevant in the *BU-phase*.

It is advisable to start addressing bias with the awareness of the project team about different bias types and understand their occurrences. Acknowledging that data does not necessarily represent the world perfectly is helpful to reveal social bias prior to any development [19, 45, 46]. First, **setting up diverse teams** helps to mitigate measurement bias that would typically occur in the *BU-phase*, and to prevent representation and deployment bias from occurring in the *DP- and DE-phase* of the project. Organizations that embrace diverse teams are better capable of identifying potential harms by introducing different perspectives in the development process. This enables the team to better define the data mining problem with more appropriate target variables and features, specify representative populations, and anticipate different use contexts [46, 47]. Second, **exchanging with domain experts** of the specific application context addresses emerging measurement bias and prevents possible representation bias in the *DP-phase*. The interaction with domain experts helps the project team to design the model with appropriate and measurable target variables and features as well as to consider all possible affected populations [36, 38]. Third, it is necessary to discuss technical and social consequences of the use of the application in the respective real-world **context** in order to prevent bias in the deployment stage. A project team should envision the application embedded in a social system and especially consider the prevailing moral values [19, 45]. If possible, multiple contexts of use can be designed. Otherwise, constraints on other user contexts can be articulated in this stage [25, 48].

To identify and prevent possible bias in the *DU-phase*, a good prior understanding of data and its underlying relationship is advisable and can be fostered by the following three methods.

First, a statistical **estimation of appropriate proxy variables** can mitigate the occurrence of measurement bias in the *DP-phase*. Depending on the design specification, it is necessary to choose proxies for variables of interest in case they are not directly observable. Examining the underlying correlations of the proxies and the true variables of interest supports feature selection [25, 38]. Second, **data plotting** can reveal possible spikes (i.e., one-time phenomena) that can be carefully removed in order to prevent representation bias [36]. Third, **exchanging with domain experts** can be effective in the DU phase to ensure a thorough understanding of the features and data in question. Domain experts might better determine affected populations in the application context and can recommend features that should be included for model training to mitigate representation and measurement bias. Also, data scientists often face data labeling challenges in the following *DP-phase*. Gaining insights from experts can help to reduce ambiguity in this decision and consequently prevent label bias [9, 36, 38].

In the *DP-phase*, five mitigation methods can eliminate underlying biases or mitigate discrimination by modifying data prior to modeling activities.

First, **data massaging** can mitigate social bias by strategically relabeling data points near the classification margin according to a ranking of the class probabilities. By relabeling individuals from an unprivileged group to favorable outcomes and simultaneously individuals from privileged groups to unfavorable outcomes, social bias can be reduced while maintaining the overall class distribution. With the class probability ranking, individuals closest to the classification margin can be identified for relabeling to minimally affect the model's accuracy [49, 50]. Second, with **reweighing** it is possible to address representation bias and social bias already present in data. Unrepresentative datasets are balanced out by upweighing underrepresented subgroups with different weights for each combination of group and label. With this approach, discrimination can be significantly reduced while maintaining overall positive class probability [49–53]. Third, **targeted data augmentation** reduces representation bias occurring in the *DP-phase*. It improves the sampling function by populating parts of the underrepresented group in the dataset [51]. Fourth, **rapid prototyping** is an effective approach for identifying different types of unintended bias. By creating a prototype and testing it in the field, practitioners can uncover overlooked populations and prevent representation bias. Furthermore, possible discriminative effects resulting from social bias can be revealed. Also, chosen features and target variables can be tested regarding their suitability to predict the outcome of interest and consequently address measurement bias [19].

Fifth, preprocessing algorithms that transform data can be applied to mitigate social bias or discriminative effects in data. **Disparate impact remover** edits features and labels in the data by learning a probabilistic transformation and applying rank ordering within groups. This ensures that information of the non-protected attributes are preserved and the class belonging can still be correctly predicted [18]. **Learning fair representation** formulates an optimization problem of finding an intermediate representation of the data that encodes it well but simultaneously removes information about membership of a protected group. The new representation space captures true underlying features that differ across groups and can then be used to learn a new

classifier in the *MO-phase* that does not use group belonging information [22]. **Optimized preprocessing** formulates a (quasi-) convex problem for the transformation and edits features and labels while complying with fairness constraints [54].

In the *MO-phase*, six model-based methods were identified which conduct modifications of learning algorithms to mitigate bias. Two additional approaches can be applied after modeling that treats the learned model as a black box. These two methods do not modify the training data or the algorithm.

First, **prejudice remover** is an approach to introduce regularization terms or constraints that mitigate social bias during modeling. It considers differences in how the learning algorithm classifies protected non-protected groups and then penalizes the total loss of the loss function based on the amount of the difference [55, 56]. Second, **adversarial debiasing** learns a classifier which maximizes accuracy while simultaneously reduces the adversary's ability to identify the protected attribute(s). The outcome is unable to carry any group discrimination information that the adversary can use, which helps to mitigate social bias during classifier training [57]. Third, **multiple models** is a method used for Naive Bayes Classifiers. Two separate models are learned, one for the protected group and one for the non-protected group. This way, the protected attribute, as well its proxies, no longer influence the outcomes of the separate models. After combining both models, probabilities are modified so that the number of positive labels is kept as high as in the original data set [58]. Fourth, a **latent variable model** discovers the actual class labels that a data set should contain if it was discrimination-free. The parameters of the model are then set in a way such that the likelihood of the data set is maximized [50]. Fifth, the design of **interpretable models** fosters transparency and trust in algorithmic models and aids identification of biases [25, 45, 59]. Sixth, **resampling** multiple training and test set splits is an important part of building a robust classifier and consequently mitigate algorithmic bias. It prevents evaluation bias in the *EV-phase* by improving diversity in the test set [18, 41, 60].

There are two post-processing methods that are applied after the algorithmic training. First, **equalized odds** mitigates social bias by accessing only aggregated data. It can solve a linear problem that finds probabilities with which to change and equalize differences in output labels [25, 61]. Second, **multitask learning** is an efficient decoupling technique that can be added on top of black-box models to learn different classifiers for different groups, thereby mitigating algorithmic bias. It parametrizes different groups differently and learns simpler, multiple functions to account for group differences [42, 62].

In the *EV-phase*, possible evaluation bias can be addressed by two approaches. First, the **representativeness of a benchmark dataset** should be verified regarding its balanced composition of all subgroups present in the model [63]. Second, the **subgroup validity** approach assures to compare performance metrics across groups instead of accepting an aggregated metric, revealing substantial performance gaps between different subgroups. Data augmentation can balance data of underrepresented subgroups [15, 20, 48, 51].

In order to prevent deployment bias and feedback bias in the *DE-phase*, three approaches can be considered. First, a **monitoring plan** can be introduced that accounts for changes in the algorithm when the context evolves [19, 25]. Second, **human**

supervision in ML-application lifecycles mitigates possible occurrence of deployment bias and prevents feedback bias. Algorithmic recommendations cannot blindly be accepted because they cannot be expected to be bias-free. Including humans in the application loop to analyze and question the outcomes can enhance objectivity [38]. Third, **randomness** can be introduced. If the outcome of an ML-application has an impact on data generation or sampling distribution, randomness can prevent feedback bias [36].

5 Discussion

The present paper addresses the emerging interest of IS research in challenges resulting from AI implementation in organizations and sheds light on the possible negative consequences of biases in ML-projects. We examined how organizations can identify and mitigate biases. Based on the widely adopted CRISP-DM, we demonstrated a systematic process to guide practitioners when identifying biases in ML-projects and provide a common ground for further theory development. We also presented a brief compilation of methods to consider what suits the specific application and the company.

This paper contributes to theory and novel management challenges in ML-projects twofold: First, the paper summarizes the current state of knowledge about bias in ML-applications in a synthesized way. Unique mitigation methods are allocated both to the bias(es) they address and the project phase they should be applied. The outline supports future researchers to clearly state the addressed problems with shared terminologies helps to solve problems in the analysis and design of ML-applications by highlighting temporal dependencies between bias occurrence and mitigation. Second, failing to recognize interdependencies in ML-models that are embedded (e.g., as modules) in broader organizational information systems can have a significant, detrimental impact on the acceptance and use of such systems as well. In this sense, the management of bias in ML-projects can become a critical aspect in IS development processes and differs fundamentally from, for example, software development.

Our research also has practical implications that could help project teams to address bias in ML-projects. First, it serves as a communication instrument for ML-project teams. It appears most fruitful to create a shared understanding across industries and to equip teams with methods that are applicable across domains. Second, besides enhancing understanding of the variety of bias types, our work also provides a comprehensive perspective on when bias can occur and how it could be addressed. This allows a better planning and assessment of risks for ML-project managers. Lastly, with our mapping of several applicable methods to particular biases, mitigation methods can be selected. A bias mitigation method should stem from application-specific discussions about what it means to be fair in the particular application context, which determines the individual mix of technical and non-technical methods.

Future research could address the limitation of our work: First, future empirical research could study the practices of how managers and their teams deal with bias in ML-projects. Such research could further substantiate and extend the work presented here. For instance, the interdependencies between different biases are not investigated

in this paper. The occurrence of a certain bias may affect the probability of a different bias to occur. Additionally, a certain mitigation method may impact other mitigation methods or biases it is not designed to address. That is, executing a mitigation method to address a certain bias could have an impact on the effectiveness of other mitigation methods or on the probability of other biases to occur. Second, while many of the articles included in this literature review stem from computer science outlets, our research could be extended by further scrutinizing this body of knowledge. Finally, existing frameworks, including the one presented here, still do not capture the full scope of fairness in all situations. Existing frameworks may not deliver clear solutions to ethical challenges in the business and data science community.

Managing Bias in ML-projects is closely related to Explainable Artificial Intelligence (XAI). That is, XAI greatly supports the detection of a ML-model's biases by disclosing the inherent mechanisms that lead to a certain outcome [7, 64]. In this paper, XAI is included as a mitigation method ("interpretable models"). However, XAI is not sufficient to eliminate the risk of bias: Even if the outcomes are interpretable and explainable, some biases can still be introduced through activities in e.g. the DE-phase.

The consolidation of IS research with social sciences and other fields such as law and ethics could provide more guidance on what it means to be fair. How can differences in moral values be handled? How to draw the line between an actual bias and a rationally based differentiation that is justifiable? Algorithms cannot judge or determine what fairness means. While we take serious attention to bias in system development, it should also be recognized that there are limits to what can be accomplished. Some concerns arising from biases go beyond designing and programming algorithms to larger societal problems. IS research could address this issue by encouraging the discussion about the articulation of normative goals that can be computationally resolved in business projects and ensure fair decision making in society.

6 Conclusion

ML-applications can incorporate inadequate properties that lead to both technically incorrect and socially unacceptable results. Besides performance criteria such as reliability, efficiency, and accuracy, freedom from bias is an integral part of the professional (risk) management of AI-systems. Therefore, we have proposed a framework based on the CRISP-DM process model that supports the identification of possible biases and methods for taking countermeasures in the management of ML-projects.

We encourage future research to address some of the limitations of our work. It could be insightful to illustrate the managing of biases and the application of mitigation methods to specific ML-projects with real data. By doing so, the contextual conditions under which each of the identified biases can occur are better captured. Because there is no one-size-fits-all solution to the diverse implementation of ML-applications, technical and social aspects of ML should be combined to bring context-awareness to research and practice.

This paper provides a possible approach by suggesting a framework to which a large part of IS research, project managers, and developers can relate. It fosters the understanding of the occurrence of different types of biases and their possible interactions, which have been so far scattered in the literature and named with different terminologies. Furthermore, we clearly address each type of bias with possible mitigation methods. We demonstrate the necessity to incorporate social and technical aspects in bias mitigation methods that can be tailored to the individual application.

References

1. von Krogh, G.: Artificial Intelligence in Organizations: New Opportunities for Phenomenon-Based Theorizing. AMD. 4, 404–409 (2018).
2. Samuel, A.L.: Some Studies in Machine Learning Using the Game of Checkers. IBM J. Res. Dev. 3, 210–229 (1959).
3. Sauter, V.L.: Decision Support Systems for Business Intelligence. John Wiley & Sons, Inc., Hoboken, NJ, USA (2011).
4. Berente, N., Gu, B., Santhanam, R., Recker, J.: Call for Papers MISQ Special Issue on Managing AI. MISQ. (2019).
5. Mikalef, P., Popovic, A., Eriksson Lundström, J., Conboy, K.: Special Issue Call for Papers: Dark Side of Analytics and AI. The European Journal of Information Systems. (2020).
6. O'Neil, C.: Weapons of Math destruction: how big data increases inequality and threatens democracy. Allen Lane, London (2016).
7. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Computing Surveys. 51, 1–42 (2018).
8. Benbya, H., Pachidi, S., Davenport, T., Jarvenpaa, S.: Call for papers: Artificial Intelligence in Organizations: Opportunities for Management and Implications for IS Research. Journal of the Association for Information Systems (JAIS) and MISQ Executive (MISQE). (2019).
9. Barocas, S., Selbst, A.D.: Big Data's Disparate Impact. Calif. Law Rev. 104, 671–732 (2016).
10. Bailey, D., Faraj, S., Hinds, P., von Krogh, G., Leonardi, P., Hall, P.: Call for Papers Special Issue of Organization Science: Emerging Technologies and Organizing. Organization Science. (2019).
11. Moreira Nascimento, A., V. Cortez da Cunha, M.A., de Souza Meirelles, F., Scornavacca, E., V. de Melo, V.: A Literature Analysis of Research on Artificial Intelligence in Management Information System (MIS). In: AMCIS Proceedings. pp. 1–10 (2018).
12. Mariscal, G., Marbán, Ó., Fernández, C.: A survey of data mining and knowledge discovery process models and methodologies. Knowl. Eng. Rev. 25, 137–166 (2010).
13. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández Orallo, J., Kull, M., Lachiche, N., Ramírez Quintana, M.J., Flach, P.A.: CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. IEEE Trans. Knowl. Data Eng. 1–1 (2019).
14. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., Others: CRISP-DM 1.0: Step-by-step Data Mining Guide. SPSS inc. 9, 13 (2000).
15. Suresh, H., Guttag, J.V.: A Framework for Understanding Unintended Consequences of Machine Learning, http://arxiv.org/abs/1901.10002, (2019).
16. Baeza-Yates, R.: Bias on the web. Commun. ACM. 61, 54–61 (2018).
17. Feuerriegel, S., Dolata, M., Schwabe, G.: Fair AI: Challenges and Opportunities. Business & Information Systems Engineering. 62, 379–384 (2020).

18. Friedler, S.A., Choudhary, S., Scheidegger, C., Hamilton, E.P., Venkatasubramanian, S., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency. pp. 329–338. ACM, New York, USA (2019).
19. Friedman, B., Nissenbaum, H.: Bias in Computer Systems. ACM Transactions on Information Systems. 14, 330–347 (1996).
20. Mitchell, S., Potash, E., Barocas, S., D'Amour, A., Lum, K.: Prediction-Based Decisions and Fairness: A Catalogue of Choices, Assumptions, and Definitions, http://arxiv.org/abs/1811.07867, (2018).
21. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. pp. 214–226. Association for Computing Machinery, New York, USA (2011).
22. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning Fair Representations. In: International Conference on Machine Learning. pp. 325–333. JMLR, Atlanta, GA, USA (2013).
23. Narayanan, A.: Translation tutorial: 21 fairness definitions and their politics. In: Proc. Conf. Fairness Accountability Transparency. pp. 1–1. , New York, USA (2018).
24. Green, B., Hu, L.: The myth in the methodology: Towards a recontextualization of fairness in machine learning. In: Proceedings of the Machine Learning: The Debates Workshop. , Stockholm, Sweden (2018).
25. Corbett-Davies, S., Goel, S.: The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning, http://arxiv.org/abs/1808.00023, (2018).
26. Silva, S., Kenney, M.: Algorithms, platforms, and ethnic bias. Commun. ACM. 62, 37–39 (2019).
27. Bellamy, R.K.E., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K.N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K.R., Zhang, Y.: AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. IBM J. Res. Dev. 63, (2018).
28. vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., Cleven, A.: Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. Communications of the Association for Information Systems. 37, 205–224 (2015).
29. Rowe, F.: What literature review is not: diversity, boundaries and recommendations. European Journal of Information Systems. 23, 241–255 (2014).
30. Schryen, G.: Revisiting IS business value research: what we already know, what we still need to know, and how we can get there. European Journal of Information Systems. 22, 139–169 (2013).
31. Levy, Y., Ellis, T.J.: A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. Informing Science Journal. 9, 181–211 (2006).
32. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MISQ. 26, xiii–xxiii (2002).
33. Randolph, J.: A Guide to Writing the Dissertation Literature Review. Practical Assessment, Research, and Evaluation. 14, 13 (2009).
34. Fink, A.: Conducting Research Literature Reviews: From the Internet to Paper. SAGE Publications (2019).
35. Viera, A.J., Garrett, J.M.: Understanding interobserver agreement: the kappa statistic. Fam. Med. 37, 360–363 (2005).

36. Baer, T.: Understand, Manage, and Prevent Algorithmic Bias. A Guide for Business Users and Data Scientists. Apress, Berkeley, CA (2019).
37. Zarya, V.: The share of female CEOs in the fortune 500 dropped by 25% in 2018. Fortune.com. (2018).
38. d'Alessandro, B., O'Neil, C., LaGatta, T.: Conscientious Classification: A Data Scientist's Guide to Discrimination-Aware Classification. Big Data. 5, 120–134 (2017).
39. Lum, K., Isaac, W.: To predict and serve? Significance. 13, 14–19 (2016).
40. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine Bias, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.
41. Lan, J., Hu, M.Y., Patuwo, E., Zhang, G.P.: An investigation of neural network classifiers with unequal misclassification costs and group sizes. Decis. Support Syst. 48, 582–591 (2010).
42. Suresh, H., Gong, J.J., Guttag, J.V.: Learning Tasks for Multitask Learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 802–810. ACM, New York, USA (2018).
43. Chouldechova, A.: Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. Big Data. 5, 153–163 (2017).
44. Collins, E.: Punishing Risk. Georgetown Law J. 107, 57 (2018).
45. Martin, K.: Designing Ethical Algorithms. MIS Quarterly Executive. 18, 129–142 (2019).
46. Jones, M.: What we talk about when we talk about (big) data. The Journal of Strategic Information Systems. 28, 3–16 (2019).
47. Barocas, S., Boyd, D.: Engaging the ethics of data science in practice. Commun. ACM. 60, 23–25 (2017).
48. Buolamwini, J., Gebru, T.: Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In: Proceedings of the 1st Conference on Fairness, Accountability and Transparency. pp. 77–91. PMLR, New York, USA (2018).
49. Kamiran, F., Calders, T.: Data preprocessing techniques for classification without discrimination. Knowl. Inf. Syst. 33, 1–33 (2012).
50. Kamiran, F., Calders, T.: Classifying without discriminating. In: 2nd International Conference on Computer, Control and Communication. pp. 1–6. IEEE (2009).
51. Chen, I.Y., Johansson, F.D., Sontag, D.: Why Is My Classifier Discriminatory? In: Advances in Neural Information Processing Systems 31 (NIPS 2018). pp. 3539–3550 (2018).
52. Hajian, S., Domingo-Ferrer, J.: A Methodology for Direct and Indirect Discrimination Prevention in Data Mining. IEEE Trans. Knowl. Data Eng. 25, 1445–1459 (2013).
53. Kamiran, F., Žliobaite, I., Calders, T.: Quantifying explainable discrimination and removing illegal discrimination in automated decision making. Knowl. Inf. Syst. 35, 613–644 (2013).
54. Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., Varshney, K.R.: Optimized Pre-Processing for Discrimination Prevention. In: Advances in Neural Information Processing Systems 30. pp. 3992–4001. Curran Associates, Inc. (2017).
55. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Fairness-Aware Classifier with Prejudice Remover Regularizer. In: Machine Learning and Knowledge Discovery in Databases. pp. 35–50. Springer, Berlin, Heidelberg (2012).
56. Zafar, M.B., Valera, I., Rodriguez, M.G., Gummadi, K.P.: Fairness Constraints: Mechanisms for Fair Classification. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). PMLR, Fort Lauderdale, Florida, USA (2015).
57. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating Unwanted Biases with Adversarial Learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340. ACM, New York, USA (2018).

58. Calders, T., Verwer, S.: Three naive Bayes approaches for discrimination-free classification. Data Min. Knowl. Discov. 21, 277–292 (2010).
59. Binder, A., Bach, S., Montavon, G., Müller, K.-R., Samek, W.: Layer-Wise Relevance Propagation for Deep Neural Network Architectures. In: Information Science and Applications (ICISA) 2016. pp. 913–922. Springer, Singapore (2016).
60. Berardi, V.L., Patuwo, B.E., Hu, M.Y.: A principled approach for building and evaluating neural network classification models. Decis. Support Syst. 38, 233–246 (2004).
61. Hardt, M., Price, E., Srebro, N.: Equality of Opportunity in Supervised Learning. In: Advances in neural information processing systems. pp. 3315–3323. Neural Information Processing Systems (NIPS) (2016).
62. Dwork, C., Immorlica, N., Kalai, A.T., Leiserson, M.: Decoupled classifiers for fair and efficient machine learning, http://arxiv.org/abs/1707.06613, (2017).
63. Ryu, H.J., Adam, H., Mitchell, M.: InclusiveFaceNet: Improving Face Attribute Detection with Race and Gender Diversity, http://arxiv.org/abs/1712.00193, (2017).
64. Samek, W., Wiegand, T., Müller, K.-R.: Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models, http://arxiv.org/abs/1708.08296, (2017).