Conceptual Foundations on Debiasing for Machine Learning-Based Software

Anuschka Schmitt
University of St. Gallen, anuschka.schmitt@unisg.ch

Maximilian Walser
University of St.Gallen, maximilian.walser@student.unisg.ch

Tobias Benjamin Fahse
University of St.Gallen, tobias.fahse@unisg.ch

Conceptual Foundations on Debiasing for Machine Learning-Based Software

Completed Research Paper

Anuschka Schmitt
University of St.Gallen,
St.Gallen, Switzerland
anuschka.schmitt@unisg.ch

Maximilian Walser
University of St.Gallen,
St.Gallen, Switzerland
maximilian.walser@student.unisg.ch

Tobias Fahse
University of St.Gallen
St.Gallen, Switzerland
tobias.fahse@unisg.ch

Abstract

Machine learning (ML)-based software's deployment has raised serious concerns about its pervasive and harmful consequences for users, business, and society inflicted through bias. While approaches to address bias are increasingly recognized and developed, our understanding of debiasing remains nascent. Research has yet to provide a comprehensive coverage of this vast growing field, much of which is not embedded in theoretical understanding. Conceptualizing and structuring the nature, effect, and implementation of debiasing instruments could provide necessary guidance for practitioners investing in debiasing efforts. We develop a taxonomy that classifies debiasing instrument characteristics into seven key dimensions. We evaluate and refine our taxonomy through nine experts and apply our taxonomy to three actual debiasing instruments, drawing lessons for the design and choice of appropriate instruments. Bridging the gaps between our conceptual understanding of debiasing for ML-based software and its organizational implementation, we discuss contributions and future research.

Keywords: Debiasing, machine learning, software, cognitive bias, bias, algorithmic bias

Introduction

Despite the opportunities of Artificial Intelligence (AI), organizations struggle to recognize, comprehend, and respond to AI's negative implications (O'Neil, 2016). Because AI increasingly relies on probabilistic machine learning (ML) models which learn patterns without explicit human programming command (Samuel, 1959), algorithmic outcomes also exhibit erroneous and unjust judgement (Mikalef, 2020). Distorted algorithmic outcomes can thereby lead to unfair and disadvantageous outcomes for certain users (De-Arteaga et al., 2022; Teodorescu et al., 2021). Such unfair outcomes caused by (a series of arbitrary) choices and practices can be defined as bias (Barocas & Selbst, 2016; Suresh & Guttag, 2019). Recent examples of bias in software include image search association of "black girls" with pornography (Noble, 2018), recommendation algorithm's suppression of content created by LGBTQ+ users (Simpson & Semaan, 2021), or unfair credit limit distribution between men and women (Vigdor, 2019).

Researchers have urged to balance the dichotomy of economic benefits and potential harms introduced through AI, last through acknowledging and addressing bias (Floridi, 2018; Gebru et al., 2021). In addition, organizations face a loss of faith if they do not grasp the underlying mechanisms and potential implications of their software (Guidotti et al, 2018), which can lead to questions about their accountability and reliability, and, in the long run, have an influence on AI investments (Benbya et al., 2019; Mikalef, 2020). For these reasons, software companies, third parties and researchers have invested in developing tools, strategies, approaches, and software tools to prevent, detect, mitigate, and deal with the consequences of bias (Amini et al., 2019; Benjamin et al., 2019). However, our understanding of the fundamental notions of debiasing for ML-based software remain in its infancy, with the notions of bias and debiasing originally stemming from psychology. Developers' unawareness of biased outcomes, difficulty of tracing back a model's causal relationships, unrepresentative data samples, as well as an over-emphasis on model prediction accuracy have all been named as detrimental factors in driving bias (Barocas et al., 2018; Cowgill & Stevenson, 2020; Mitchell et al., 2018). Even if practitioners might be aware of the general issue of bias, volume, velocity and fragmentation of debiasing tools make it difficult to fully comprehend issues that exist in a specific context and how to best address these. ML-based software, including its probabilistic nature, data dependency, and complexity, exacerbates issues of biased software and calls into question extant theoretical notions of bias and debiasing. We argue that debiasing and the identification of suitable debiasing instruments can be as opaque as software itself, especially from an organizational perspective. Ultimately, there is no standardized and practice-oriented classification to organize debiasing instruments. Such a classification would enable researchers and developers to more effectively understand, evaluate, compare, and implement debiasing instruments. Hence, this paper poses the following research question: *How does debiasing matter for software development and how can related approaches be conceptualized and structured?*

To address this question, we developed a taxonomy for debiasing in the software product development process. Along five iterations, we identified meta-categories relevant to the classification of debiasing approaches and organize relevant characteristics based on a systematic literature review on cognitive and ML biases, and debiasing approaches. To revise and evaluate our taxonomy, we conducted nine semi-structured expert interviews with researchers and practitioners from the software industry. Our research contributes to the systematic conceptualization and practical comprehension of debiasing. We provide a more nuanced understanding of the types of debiasing instruments, as well as intended effects and potential implementers along the software development phases. We aim to demonstrate the heterogeneity of debiasing and its appropriateness for ML-based software, specifically.

Conceptual Background

Bias in the Software Product Development Process

Bias in software accounts for many issues, with its symptoms often hitting the user. Bias can lead to unfair outcomes, including direct and indirect discrimination by race (Schlesinger et al., 2018), gender (Adams & Loideáin, 2019) or disability (Trewin, 2019). As stated by Patel (2021), "[r]emoving bias from AI is not easy because there's no one cause for it". Considering increasing technological sophistication of software, bias can be introduced through the underlying model and related data, a concept we refer to as ML bias. ML's probabilistic output is associated with higher uncertainty and is thus more prone to erroneous and biased outcomes, while its non-causal complexity induces existing biases to stay unnoticed. Second, as ML software is sensitive to the data it is used on, changes in data can easily introduce bias after deployment. Third, bias can find its origin in human thinking and decision-making as ML models are developed, trained, and used by humans. As part of the following, we consider both cognitive and ML biases.

Cognitive Biases

By nature, software development is strongly influenced by human decisions (Fleischmann et al., 2014). As a result, one category of biases relevant to software development are human cognitive biases. Cognitive bias is a term in cognitive psychology for systematic inaccurate proneness in perceiving, remembering, thinking, and judging under uncertainty (Tversky & Kahneman, 1974). Cognitive biases often manifest themselves in gut feelings, beliefs or thoughts that lead to a deviation from objective reasoning and that remain unconscious (Tversky & Kahneman, 1974). Kahneman's (2011) two-system distinction of how the brain reasons helps explain the occurrence of bias: The first system is fast, constantly engaged, emotional, stereotyping. The second system is slow, rational, objective, and conscious. When we need to make decisions fast, we usually operate in System 1, while complex calculations and reflections take place in System 2. While System 1 thinking is critical to survive in a complex world with masses of stimuli, it induces cognitive bias.

We decided to provide a distinguished conceptualization of cognitive biases as “for many, if not most, of […] ML biases, human biases can be the cause. […] [P]rior works could serve as a bridge to translate and locate how and when human bias is salient and how it might take effect within the typical logic of ML applications.” (van Giffen et al., 2022, p. 105). In the last decade, an increasing number of articles have been published on cognitive biases in software engineering (Fleischmann et al., 2014; Mohanani et al., 2020). In general, there exist over 200 cognitive biases and at least 120 cognitive biases have been researched in the context of IS research (Fleischmann et al., 2014). Table 1 provides an overview of eight overarching cognitive bias categories and respective examples in software development.

Category	**Explanation**	**Exemplary Cognitive Bias in Software**
Action-Oriented Bias	Biases that cause individuals to jump to decisions without properly considering all relevant information or possible alternatives (Keren, 1997)	Software developers are *overconfident* that a project will have a certain duration or that all data are considered, leading to important aspects remaining unconsidered.
Decision Bias	Biases that affect specific decision making (Park & Lessig, 1981)	*Hyperbolic discounting* is the tendency to prefer instant rewards to later ones, even if those are smaller. For a software developer it might be easier to only consider a particular set of data or group of people.
Interest Bias	Biases that skew thinking based on individual preferences (Mohanani et al., 2020)	The *confirmation bias* is the tendency to better process information that confirms own beliefs and knowledge. This leads to poor judgement when it comes to testing the user utility of software products, which were created by the person who attempts to test them.
Memory Bias	Biases that disturb correct memory processing (Liftiah et al., 2021)	*Hindsight bias* leads software developers to believe that a certain outcome of a data analysis has already been predicted accurately in the past which is exacerbated by a lack of archived records.
Pattern recognition Bias	Biases that cause individuals to notice information more with which they are already familiar (Fleischmann et al., 2014)	*Availability bias* leads software developers to make their decisions based on information that is easier accessible and not necessarily based on data that is important for certain decisions.
Perception Bias	Preconceptions that interfere with the processing of all new information (Fleischmann et al., 2014)	*Fixation* is the tendency to focus disproportionately on one aspect of a situation, object or event. These self-imposed or imaginary barriers can be reinforced through strict requirements.
Social Bias	Biases affecting judgment because of individuals' attitudes and social interaction (Fleischmann et al., 2014)	The *bandwagon effect* leads developers to adapt their software to existing other products as opposed to creating products that bring the maximum utility to their users.
Stability Bias	Prejudices that affect the processing of new information, even if that information is objectively superior (Fleischmann et al., 2014)	The *anchoring or adjustment bias* contributes to poor understanding of problematic situations and poor estimation. For example, when customer data does not correctly represent the customer group, false assumptions are raised.

Table 1. Cognitive Bias Categories based on the Taxonomy of Fleischmann et al. (2021)

The explicit consideration of cognitive biases illustrates the importance of of how biases find their way in not only a software but also the processes and organizational decisions related to the software development process such as project management (Jorgensen & Grimstad, 2012).

Machine-Learning Biases

Like humans, AI systems become a catalysator for biases to be adopted and exacerbated (Fahse et al., 2021). AI presents the human-like ability of machines to perform autonomous cognitive tasks (Benbya et al., 2020). AI has been gaining more and more attention in recent years in the IS research landscape, promising to solve complex problems and enhance human capabilities (Benbya & Leidner, 2018; Collins et al., 2021; Dwivedi et al., 2021). Oftentimes, conscious and unconscious human decisions in the data and labeling steps result in biased data (labelling bias). Through the training process, these cognitive biases find their way into the resulting software, representing a *User to Data* bias (Mehrabi et al., 2021). Second, biases can be introduced into ML-based software during model design decisions in the development phase. For example, a model can be optimized to perform best for certain groups by testing the model on an imbalanced test set, leading to *Data to Algorithm biases* and thus biased algorithmic results before training or deploying a model. Third, after the software is deployed, cognitive biases can influence the way humans use the software: Users may perceive the software's output based on their own internalized preconceptions. This way, *Algorithm to User* biases such as deployment bias are introduced (Suresh & Guttag, 2019). These can arise due to unexpected output in unsupervised learning models or deploying a model in an unsuitable context (Mehrabi et al., 2021). Based on a systematic review, we structured ML bias along a total of 19 biases and three main points for biases to manifest in ML-based software (Mehrabi et al., 2021).

Category	Explanation	Exemplary ML Bias in Software
User to Data	Biases that are introduced through the unconscious biases of the user into the data (Mehrabi et al., 2021).	*Population bias* occurs when the chosen population does not represent the target population (Olteanu et al., 2019).
		Self-selection bias occurs when a subgroup of a population selects itself, thereby only including a subsection of data in the sample (Mehrabi et al., 2021).
Data to Algorithm	Biases that exist in data and can lead to biased algorithmic results (Mehrabi et al., 2021).	*Measurement bias* occurs when chosen features and labels are imperfect proxies for the real variables of interest. For instance, "credit worthiness" is an abstract concept that is often implemented with a measurable proxy such as a credit score (Suresh & Guttag, 2021).
		Aggregation bias occurs when a single model is used for data where there are some underlying groups or categories of examples that should be considered differently (Suresh & Guttag, 2021).
Algorithm to User	Biases that emerge and can lead to user biases (Mehrabi et al., 2021).	According to the *popularity bias*, things that are more popular are presented more and are therefore more visible (Nematzadeh et al., 2018, p.1).
		If a model is used and interpreted in a different context than it was built for, *deployment bias* can occur. For instance, an algorithm predicts when laptops of employees should be updated to not disturb during working times but is used to monitor working hours and adjust bonuses accordingly.

Table 2. ML Bias Categories Based on the Classification of Mehrabi et al. (2021)

It is important to note that there is a certain overlap of cognitive and ML biases. In that sense, cognitive biases and human intervention can take place in any of the three ML bias categories. Eventually, bias in ML-based software is a human problem with cognitive biases being (indirectly) adopted and amplified in the training and deployment of ML-based software. In the context of our study and from an organizational perspective, it is crucial to make the distinction between cognitive biases and ML biases. Focalizing the ML biases helps understand *how* the nature of such systems introduces novel issues of bias. While ML bias is necessarily linked to a software-implemented model, cognitive biases can concern team and organizational issues and span multiple software development steps (e.g., the diversity within a software product team).

Debiasing and Its Approaches in Software Development

Since biases introduced at different stages of the software development process can lead to error, misuse, and unfair outcomes for society, companies across industries rate "reducing biases" as an important issue to tackle (McKinsey, 2014). The term debiasing is oftentimes understood as preventing biases or mitigating their deleterious effects (Stac & MacMillan, 1995). Originating from the field of cognitive psychology, debiasing can be viewed as a form of "destructive testing" for human judgment by challenging the viability of a judgement (Fischoff, 1981). Larrick (2004) described debiasing as closing the gap between descriptive behavior and normative ideals. As part of all definitions, debiasing involves some sort of intervention. Fischoff (1981) provided one of the first frameworks on debiasing cognitive biases. He introduced the distinction between debiasing the person itself, the task, or an assortment between the two. A person can, for example, be debiased by raising awareness through training interventions (Shepperd et al., 2018) or by active open mindedness (Riggs, 2010). Tasks can be debiased by gaining multiple perspectives through diverse teams (Kaufmann et al., 2009) or by software tools which facilitate the decision-making process (Ralph, 2010). While nascent psychology literature was concerned with cognitive strategies and human reasoning (Stanovich, 1999), these approaches were soon criticized as humans struggle to recognize and correct their own biases. Later on, the use of "tools" such as decision aids and statistical decision analysis were considered as helpful approaches to reduce the descriptive-normative gap of bias (Kahnemann, 2003).

With the rise of contemporary ML-based systems and their biases having moved into the limelight of research discussions, debiasing has regained attention beyond the field of psychology. In the field of software engineering, Mohanani et al. (2020) and Fleischmann et al. (2014) present selected debiasing instruments including diverse teams (Nelson, 2014), workshops for software developers (Shepperd et al., 2018), mindfulness meditation (Bishop et al., 2004) or shared documentation on different software tools (Ralph, 2010). Conducting a workshop, for instance, can create awareness and warning about unconscious bias and structural inequalities in the software development process. While technological artefacts such as decision support systems have been introduced as useful debiasing tools, ML-based software paradoxically introduce biases on their own. Extant conceptualizations of and findings on debiasing might hold to a certain extent and can be applied for software development. However, the question arises of how we can conceptualize debiasing for ML-based software development. In the context of our study, debiasing instruments are technical and non-technical methods, tools, strategies, approaches, and software tools whose core purpose are detecting, preventing, or mitigating cognitive or ML biases and their harmful consequences for ML-based software.

Within the last decade, good practice approaches for data debiasing have been provided by researchers and practitioners alike. For instance, the Berkman Klein Center and MIT Media Lab developed workshop slides and a card deck for (technical) teams to spot biases in their own processes and systems.[1] Next to debiasing individuals and tasks, data and algorithms can also be the subject of debiasing instruments, (Fahse et al., 2021; Mehrabi et al., 2021; Suresh & Guttag, 2021). These include rapid prototyping for an early detection of biases (Kliegr et al., 2021), as well as data massaging to mitigate social bias such as discrimination by relabeling data points near the classification margin (Kamiran & Calders, 2012). Benjamin et al. (2019) present a framework for data licensing similar to the licensing of open source software, aimed at ensuring more transparency in the data market. Holland et al. (2018) created a "dataset nutrition label" that aims to facilitate standardized data analysis. Turning towards more comprehensive debiasing instruments for algorithms, commercial toolkits such as IBM's AI Fairness 360 consist of multiple algorithms such as reweighing (modifying weights of different training data examples) and adversarial techniques for prediction tasks (Bellamy et al., 2018). Metrics to measure individual and group fairness such as Euclidean and Manhattan Distance, have been tested (Calmon et al., 2017). Naturally, debiasing approaches for algorithms are more technical in nature. Amini et al. (2019), for instance, developed an algorithm for detecting and tackling potentially unknown racial and gender biases within training data for facial detection software. Other domains also help inform how biases in ML software development can be considered and addressed, by, for instance, learning from audits from other domains such as finance or electronic hardware. Inspired by the latter, failure mode and effects analysis (FMEA) has been proposed as an approach to integrate ethical considerations in the software design process by calculating the likelihood and severity of fairness-related failures of a ML-based system (Li & Chignell, 2022).

[1] https://aiblindspot.media.mit.edu/index.html

Classifications of Dimensions and Characteristics of Debiasing Instruments

Extant reviews on debiasing, however, lack to provide a model- and system-agnostic overview of debiasing instruments which could potentially guide practitioners along specific types of ML-based software projects. Several researchers have developed overviews of debiasing instruments, including Keren's (1990) classification of cognitive aids to reduce and eliminate cognitive biases. Sell et al. (2014) and Larrick (2004) also focused on debiasing approaches for cognitive biases. The latter introduces modifications in the decision environment as specific debiasing instruments for individuals to improve their retirement savings, for instance. In a similar vein, Muntwiler (2021) provides nine clusters of debiasing techniques (e.g., checklists, what if, group debiasing, analogical thinking) to be used at different stages of a decision-making process. While extant overviews allow to better understand the current landscape on debiasing classifications, none of these overviews are focused on ML-based software. Towards that end, van Giffen et al.'s (2022) overview of 24 bias mitigation instruments for data mining helps understand a variety of contemporary ML bias mitigation approaches. We aim to bring together these and other approaches to identify key elements defining a debiasing instrument to understand and distinguish debiasing of ML-based software from debiasing in non-ML-based scenarios (Bailey, 1994). No comprehensive classification of debiasing instruments considering both cognitive and ML biases for software development exists yet, which could be helpful for researchers and practitioners.

Research Methodology

The systematic classification of a unit of analysis, also coined taxonomy, allows to structure complex domains such as the domain of debiasing instruments (Nickerson et al., 2013). We aim to embed our taxonomy within the software product development process which enables us to mark out relevant elements to consider when studying debiasing instruments. In addition, this allows us to capture the complexity of an organizational context beyond an individual ML-based model, and thus consider both cognitive and ML bias. We followed four key steps aimed at informing our conceptualization of debiasing approaches in the software development process (Table 3).

Steps	Outcomes
Database Creation	Assumptions about the definition, key characteristics, and impact of debiasing instruments. Consolidated list of 63 debiasing instruments found in literature and commercial instruments following a systematic literature review (Webster & Watson, 2002) and qualitative content analysis (Wolfswinkel et al., 2013)
Taxonomy Development	Initial conceptualization of debiasing taxonomy, categorized into two key dimensions (Nickerson et al., 2013)
Taxonomy Evaluation	Refined and enhanced debiasing characteristics and dimensions, evaluated taxonomy through semi-structured expert interviews (Kundisch et al., 2021)
Taxonomy Application	Review of three commercial debiasing instruments (Ragin and Becker, 1992)

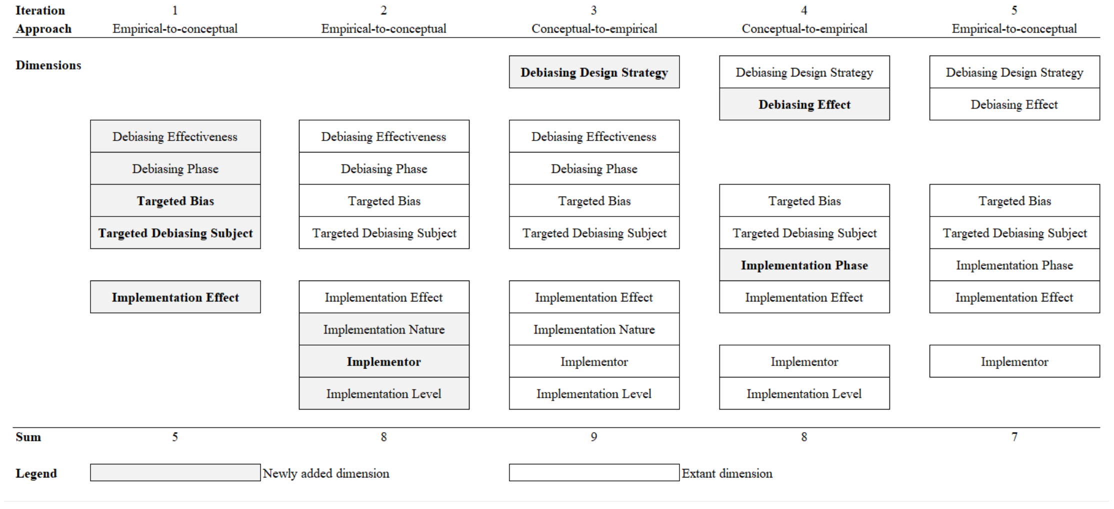
Table 3. Taxonomy Development Process for Debiasing Conceptualization

Step 1: Database Creation

Given the nascency of the research field and the scattered findings regarding the presentation of debiasing, we realized that individual instruments are commonly introduced as a direct response to specific biases (Mohanani et al., 2020). Search strings focusing on debiasing only rendered little and inappropriate results regarding our research endeavor. We therefore conducted two systematic literature reviews on cognitive biases and ML biases according to Okoli (2015) and vom Brocke et al. (2015). We thereby considered the following seven databases: ACM Digital Library, Scopus, Web of Science, EBSCO, Science Direct, IEEE Xplore and GALE Ebooks. To screen the research about cognitive biases in software engineering we build the following search string: ("bias" OR "cognitive bias" OR "decision making" OR" debiasing") AND (“information systems research” OR “Software development” OR “Software product design process” OR “software engineering”). In the second literature stream, we used the following search string: (“machine bias” OR “machine learning bias” OR “algorithmic bias” OR “bias”) AND (“Project management” OR “artificial intelligence” OR “deep learning” OR “representation learning”). We collected additional papers using backward snowballing, backward reference search, and forward reference search. We found a total of 72 papers relevant for biases and debiasing instruments for cognitive biases. We analyzed a total of 31 papers about debiasing instruments for ML biases. Many debiasing instruments are not necessarily presented as part of published research. We thus reviewed tech-focused research institutions and news to extend our database. Prominent examples include the World Economic Forum’s AI Fairness Global Library which provides over 30 debiasing tools sorted by tool type, geographical origin, and available language.[2] Based on an open coding system, we added descriptive codes while reviewing the individual debiasing instruments, following qualitative content analysis methods proposed by Mayring (2000).

Step 2: Taxonomy Development

To develop a comprehensive taxonomy of debiasing instruments, we relied on Nickerson et al.’s (2013) taxonomy development method. Grounded on the defined main purpose and users of the taxonomy, we formulated the following meta-characteristic: *The systematic classification of the defining characteristics of all methods, techniques and (software-) tools to prevent, mitigate and debias cognitive and machine learning biases in ML-based software development*. We hereby considered the context of software platform and product development processes to inform the practical implementation of debiasing instruments and construct a comprehensive contribution to the extant knowledge of debiasing approaches. Based on the proposed conditions of Nickerson et al. (2013) and Sowa and Zachmann (1992), we defined nine objective (e.g., “At least one object is classified under every characteristic of every dimension.”) and six subjective ending conditions (e.g., Does the number of dimensions allow the taxonomy to be meaningful without being unwieldy or overwhelming?”) to be met to end the iterative taxonomy development. For the first two iterations, we followed an empirical-to-conceptual approach. According to this approach, a smaller collection of already classified objects are used to identify characteristics, which are then structured into dimensions. In every iteration, we included between ten and twenty new debiasing instruments extracted from research overviews (e.g., Mehrabi et al., 2021; Mohanani et al., 2020; van Giffen et al., 2022). In the third iteration, common characteristics of debiasing instruments were identified by studying fundamental works in the research field of debiasing cognitive biases such as the work of Fischoff (1981) and Keren (1990). As part of the last iteration, we ensured to have considered commercially available debiasing tools we encountered at later research stages.



Iteration	1	2	3	4	5
Approach	Empirical-to-conceptual	Empirical-to-conceptual	Conceptual-to-empirical	Conceptual-to-empirical	Empirical-to-conceptual
Dimensions			**Debiasing Design Strategy**	Debiasing Design Strategy	Debiasing Design Strategy
				Debiasing Effect	Debiasing Effect
	Debiasing Effectiveness	Debiasing Effectiveness	Debiasing Effectiveness		
	Debiasing Phase	Debiasing Phase	Debiasing Phase		
	Targeted Bias	Targeted Bias	Targeted Bias	Targeted Bias	Targeted Bias
	Targeted Debiasing Subject	Targeted Debiasing Subject	Targeted Debiasing Subject	Targeted Debiasing Subject	Targeted Debiasing Subject
				Implementation Phase	Implementation Phase
	Implementation Effect	Implementation Effect	Implementation Effect	Implementation Effect	Implementation Effect
		Implementation Nature	Implementation Nature		
		Implementor	Implementor	Implementor	Implementor
		Implementation Level	Implementation Level	Implementation Level	
Sum	5	8	9	8	7

Legend ▨ Newly added dimension □ Extant dimension

Figure 1. Taxonomy Development Based on Nickerson et al. (2013)

[2] https://www.aifairnesslibrary.com/resources

Step 3: Taxonomy Evaluation

We evaluated the quality of the taxonomy based on consistency, robustness, comprehensiveness, extendibility and explainability (Szopinski et al., 2019). We conducted nine semi-structured interviews with experts involved in software development (e.g., software product managers, software developers) to ensure practical usefulness and researchers stemming from IS and ethics to evaluate the theoretical foundation of the taxonomy. The interview guide consisted of 23 questions, with the first cluster of questions focusing on the quality of the taxonomy and the second part concerning the practical implementation of the taxonomy. Interviewees received the final version of our taxonomy, meta-characteristic and exemplary debiasing instruments upfront. Interviews lasted between 27 and 78 minutes.

Regarding the consistency of our taxonomy, the number of dimensions (seven) was perceived as adequate by all the interviewees. One of the interviewees mentioned that the most important distinction when using debiasing instruments is the distinction between unstructured and structured data. Therefore, we split the characteristic in *Targeted Debiasing Subject* into these two characteristics. Concerning the robustness of our taxonomy, interviewees agreed that there are no major overlaps between the dimensions. However, correlation between certain characteristics were pointed out, including the tackling of cognitive biases, which necessarily affects a task or an individual. Feedback regarding the comprehensiveness of our taxonomy differed between the two interview clusters with the four experts from IS research having little to no difficulties understanding the purpose of the taxonomy and the wording of individual dimensions and characteristics. We changed *Debiasing Design Strategy* to *Debiasing Meta-Strategy* and *Debiasing Effect* to *Debiasing Target*, which was positively confirmed by the software industry interviewees concerning understandability. Interview feedback on the extendibility and explanatory power of the taxonomy were positive.

Step 4: Taxonomy Application

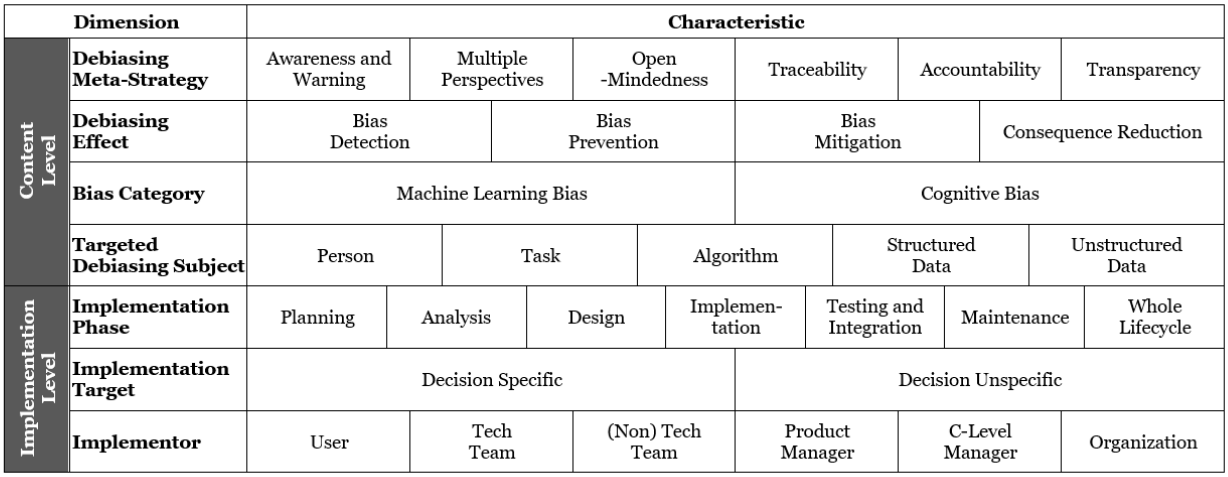
To examine the applicability of our taxonomy, as well as to support the depth and breadth of our analysis, we chose three debiasing instrument to examine in greater detail, namely team diversity, dataset nutrition labels, and human supervision. We set out to choose instruments that 1) span multiple debiasing strategies, tackled bias, stakeholders, and development phases, and 2) were accessible via various data sources (e.g., empirical studies, articles, open source, user discussions and posts) to enable a rich and comprehensive analysis.

Taxonomy on Debiasing for the Software Development Process

In the following section, we present our consolidated and revised version of the debiasing taxonomy after conducting five iterations and integrating the feedback from our expert interviews. Our unit of analysis is a single debiasing instrument. The overall taxonomy and its dimensions and characteristics are presented in Table 4 below. According to the nature of a taxonomy (Gupta & Bostrom, 2009), a specific debiasing instrument would fall into one characteristic for each dimension, i.e., we expect an instrument to focus on a particular effect or to focus on one debiasing subject. Debiasing instruments can be structured along two dimensions: the content and the implementation level of the instrument.

Content Level

The first cluster of dimensions relates to the content of a debiasing instrument. The *Debiasing Meta-Strategy* describes the main strategy the debiasing instrument is deployed for. The primary cause of cognitive biases can be traced back to the individual who makes the decisions (Bazerman & Moore, 2009; Shepperd et al., 2018). In that sense, creating *awareness and warning* about biases and the general impact biases is one of the most common strategies for improving rationality and objectivity in decision making. Several steps are suggested to facilitate learning, including psychological explanations and workshops introducing the reasons and implications of biases. Several researchers also mention the bias-reducing effect of assessing a decision problem from *multiple perspectives*, which can be enabled by including different stakeholders, domain experts and areas of expertise in the software process (Kaufmann et al., 2009). In a similar vein, active *open mindedness* challenges present views and knowledge (Riggs, 2010).



	Dimension	Characteristic						
Content Level	Debiasing Meta-Strategy	Awareness and Warning	Multiple Perspectives	Open -Mindedness	Traceability	Accountability	Transparency	
	Debiasing Effect	Bias Detection	Bias Prevention	Bias Mitigation	Consequence Reduction			
	Bias Category	Machine Learning Bias	Cognitive Bias					
	Targeted Debiasing Subject	Person	Task	Algorithm	Structured Data	Unstructured Data		
Implementation Level	Implementation Phase	Planning	Analysis	Design	Implemen-tation	Testing and Integration	Maintenance	Whole Lifecycle
	Implementation Target	Decision Specific	Decision Unspecific					
	Implementor	User	Tech Team	(Non) Tech Team	Product Manager	C-Level Manager	Organization	

Table 4. Taxonomy for Debiasing Instruments for Software Development

One-time debiasing training interventions, such as educational video games and training videos, can have a significant effect on practicing active open mindedness (Morewedge et al., 2015). In addition, slowing down the decision process, e.g., through cognitive forcing functions, allows individuals to reflect on their decision as the brain is induced to shift from fast, emotional thinking to more controlled, and rule-based thinking (Kahneman, 2003; Lilienfeld et al., 2009). In that sense, *open mindedness* targets the individual and can have long-term effects, whereas the *multiple perspectives* strategy requires the involvement and consideration of multiple individuals, and can also be used as an ad hoc strategy, e.g., when attempting to understand data for a particular ML software project. *Traceability* can be a key aim of a debiasing instrument to identify and measure the impact of modifications to a model, as well as to understand discrepancies between identified requirements and their implementation (Mohan & Jain, 2008). In practice, traceability is often performed in a retrospective approach, and its advantages are consequently not always exploited to the maximum extent (Cleland-Huang et al., 2014). With *accountability,* beliefs and actions are justified, and individuals and organizations held responsible for their decisions (Correia, 2017). Accountability can be achieved by external oversight boards and formal authority. Debiasing instruments can also aim to make data and decision processes more transparent (Fischoff, 1981). *Transparency* can be achieved by demanding human explanations and thus creating incentive to make decisions in a more objective manner, as well as by using technical tools to make datasets more transparent (Steffel et al., 2016; Tomalin et al., 2021).

We recognized that the *effect* of debiasing instruments can be further classified by either *detecting a bias, preventing a bias, mitigating a bias, or by reducing the consequences of a bias*. In that sense, different debiasing instruments can act at different temporal points around when a bias is emerging (Fischhoff & Baruch, 1982). For instance, in situations where data is already collected, labeled, and deployed and an organization has come to realize that the deployed data is biased, dealing with the consequences of the implemented bias is paramount. In the best case, organizations think of where biases could occur and detect these before having more significant consequences in the software development process. The *Bias Category* dimension distinguishes between whether a debiasing approach is aimed at tackling a purely cognitive bias or machine learning biases (Mehrabi et al., 2021). Last, the question arises who or what the *targeted debiasing subject* is. This dimension was already considered by Fischoff (1981) who introduced the distinction between debiasing a person or a task. Due to the increasing importance of ML-based models in software, we further differentiate task as a debiasing subject and include algorithm, structured data, and unstructured data as three additional characteristics. Based on the expert interviews, we distinguish between structured data, that is data with an imposed composition and thus machine-understood dependencies, and unstructured data, that is data in an unstructured format and without data type declaration (Weglarz, 2004).

Implementation Level

Beyond the nature and aim of a debiasing approach, debiasing approaches can differ on who, how, and when they are implemented. The next set of dimensions therefore concerns the implementation. We hereby view the *implementation phase* as a crucial dimension. We followed the six stages of the Information Systems Development Lifecycle (ISDL) used to organize software projects as referred to by various IS researchers (Beynon-Davies et al., 2000; Huang, 2008; Sing, 2016). Accordingly, we distinguish among the planning, analysis, design, implementation, testing and integration, and maintenance phases. According to the interview feedback, most practitioners can easily apply their own product development process to this lifecycle. While the planning phase is central for software managers and decisions about what information will be used, composition of code is defined as part of the design phase. Later stages of the product development are just as crucial, with actual coding not taking place until the implementation phase. As part of the testing and integration phase, it can be checked whether biases may have been incorporated (Huang, 2008; Nunamaker et al., 1990; Singh, 2016). A debiasing instrument can also be deployed throughout the whole ISDL, such as instruments targeting an individual (Beynon-Davies et al., 2000; Huang, 2008).

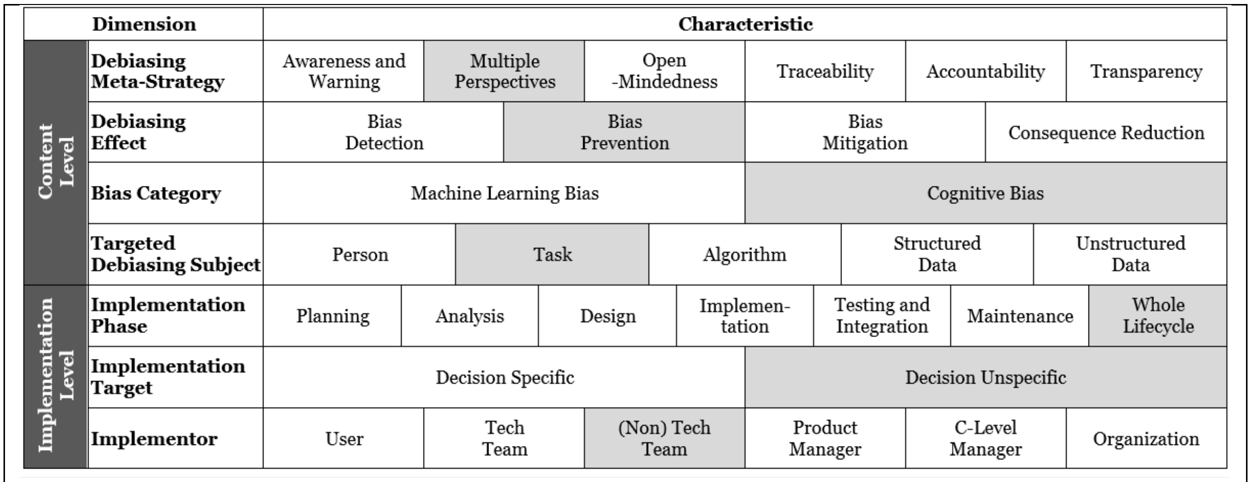
Linked to the cycle of software development, Kaufmann et al. (2009) introduced another dimension relevant to the implementation of debiasing instruments, namely that of the *implementation target*. A debiasing instrument can be aimed at one decision within the software development process, whereas a *decision unspecific* debiasing instrument is aimed at several decisions and can thereby span multiple implementation phases. Some instruments like awareness training or team diversity cannot be aimed at one decision only, making the impact of decision unspecific instruments difficult to measure. Last, the *implementor* dimension, based on Noor's (2020) classification, is concerned with the stakeholder responsible for the successful implementation of a respective debiasing instrument. The user being the responsible implementor becomes particularly important regarding the *algorithm to user* ML-bias category. Debiasing instruments focused on data or algorithms are usually implemented by the tech team, whereas debiasing instruments can be specifically targeted to a manager responsible for designing, managing, and monitoring software products throughout the product lifecycle. These instruments become particular important in the context of accountability-targeted debiasing instruments. Some debiasing instruments can only be implemented by the whole organization.

Applying the Taxonomy Along Three Debiasing Instruments

Based on our evaluated and refined taxonomy, we aim to illustrate the taxonomy as a descriptive, explorative tool to identify debiasing instruments for a specific ML algorithm and its application, as well as to verify whether a chosen debiasing approach meets implementors' expectations. We therefore chose three exemplary debiasing instruments which differ in respective dimensions as well as empirical understanding.

Team Diversity

Team diversity is not a novel approach to debiasing yet has become even more important in the context of (ML-based) software given the pervasiveness and far-reaching implications discrimination in decision-making software can have (Kaufmann et al., 2009). Creating well-balanced teams with individuals from different economical, racial, and academic backgrounds allows to introduce multiple perspectives into the software development process, thereby challenging predominant narratives and assumptions imposed by individual team members (Montibeller & von Winterfeldt, 2015; Nelson, 2014). Demographic group diversity has been shown to reduce prediction errors and improve decision making (Cowgill et al., 2020). With females being more likely to make ethical business decisions (Dawson, 1997; Peterson et al., 1991) and software jobs commonly targeting more male than female talent (Lambrecht & Tucker, 2019), a common debiasing approach is to invest in more female or minority developers (Bessen et al., 2022). In that sense, team diversity is fundamental to debiasing the software development process and allows to challenge implicit thought patterns and thus prevent cognitive biases sustainability and preemptively.

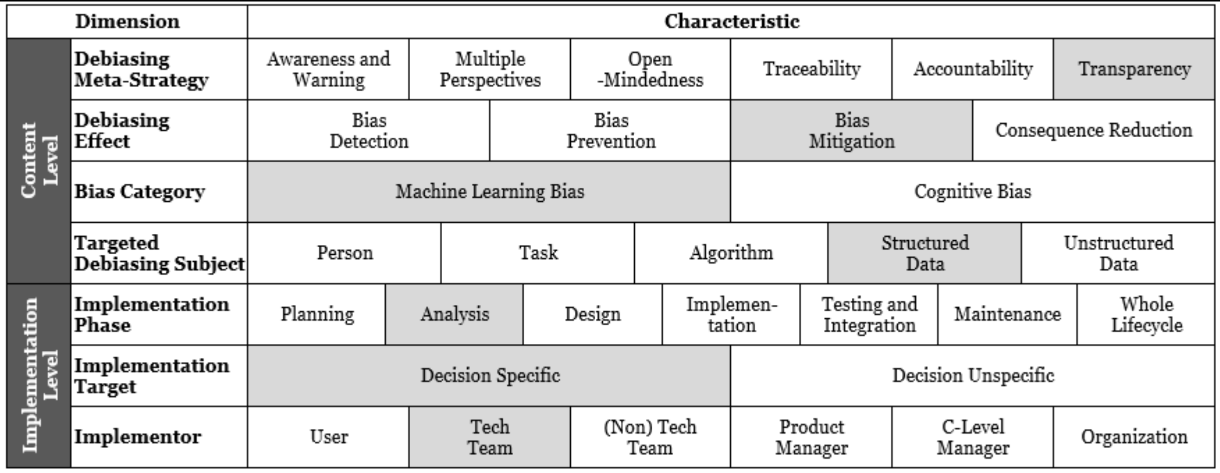


	Dimension	Characteristic						
Content Level	Debiasing Meta-Strategy	Awareness and Warning	[✓ Multiple Perspectives]	Open -Mindedness	Traceability	Accountability	Transparency	
	Debiasing Effect	Bias Detection	[✓ Bias Prevention]	Bias Mitigation	Consequence Reduction			
	Bias Category	Machine Learning Bias	[✓ Cognitive Bias]					
	Targeted Debiasing Subject	Person	[✓ Task]	Algorithm	Structured Data	Unstructured Data		
Implementation Level	Implementation Phase	Planning	Analysis	Design	Implemen-tation	Testing and Integration	Maintenance	[✓ Whole Lifecycle]
	Implementation Target	Decision Specific	[✓ Decision Unspecific]					
	Implementor	User	Tech Team	[✓ (Non) Tech Team]	Product Manager	C-Level Manager	Organization	

Table 5. Taxonomy Application for Team Diversity

Data Nutrition Labels

On a more granular level, debiasing instruments can target specific datasets. With institutions urging the documentation of ML datasets, debiasing instruments such as data nutrition labels are less prominent yet crucial to prevent the reproduction or amplification of bias reflected in datasets early in the software development process (Gebru et al., 2021). Inspired by food nutrition labels, labels for datasets can point towards anomalies in distributions or missing data. Making transparent the quality of a dataset can help inform the decision on whether to use a dataset for a particular use case. Several published design solutions such as data statements for natural language processing (Bender & Friedman, 2018), nutritional labels developed for ranking algorithms (Yang et al., 2018), and informative, supplementary datasheets for datasets (Gebru et al., 2021) have informed this instrument primarily used by data analysts. Beyond increasing transparency around data used for a particular model, increasingly appropriate data is used for a particular software product. Another key promise of data nutrition labels is, that compared to post accountability measures such as audits or reverse engineering, data engineers are forced to explain the choice of data (Yang et al., 2018). In a similar vein, Mitchell et al. (2019) put forward model cards for Google's face and object detection projects to document the performance characteristics of ML models and increase transparency of model reporting. These are, however, now publicly accessible in hindsight of the software deployment.

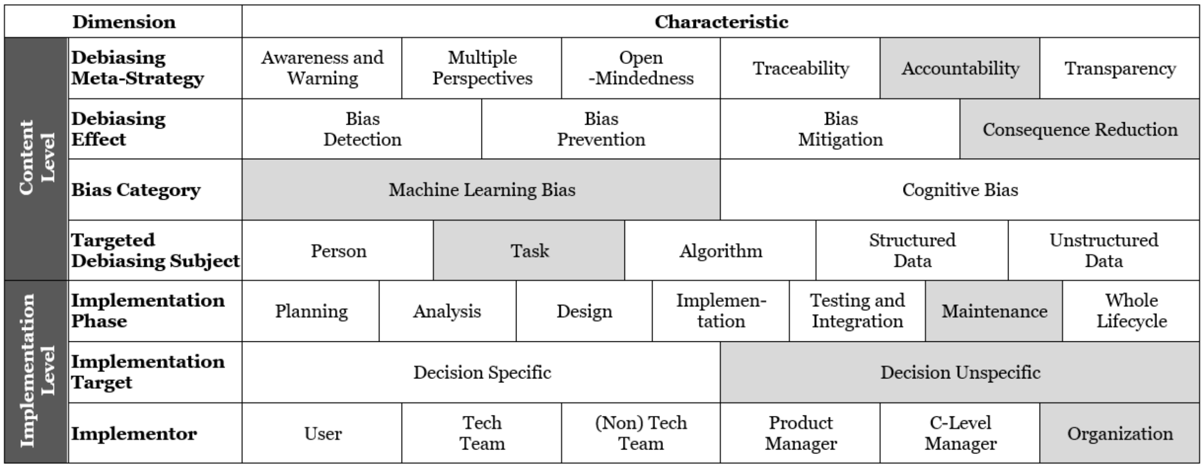


	Dimension	Characteristic						
Content Level	Debiasing Meta-Strategy	Awareness and Warning	Multiple Perspectives	Open -Mindedness	Traceability	Accountability	[✓ Transparency]	
	Debiasing Effect	Bias Detection	Bias Prevention	[✓ Bias Mitigation]	Consequence Reduction			
	Bias Category	[✓ Machine Learning Bias]	Cognitive Bias					
	Targeted Debiasing Subject	Person	Task	Algorithm	[✓ Structured Data]	Unstructured Data		
Implementation Level	Implementation Phase	Planning	[✓ Analysis]	Design	Implemen-tation	Testing and Integration	Maintenance	Whole Lifecycle
	Implementation Target	[✓ Decision Specific]	Decision Unspecific					
	Implementor	User	[✓ Tech Team]	(Non) Tech Team	Product Manager	C-Level Manager	Organization	

Table 6. Taxonomy Application for Data Nutrition Labels

Human Supervision

Human supervision, e.g., through independent oversight boards, helps create human accountability and enforce continuous analysis of software systems (Baer, 2019). Subject matter experts, target groups, and civil society organizations such as Algorithm Watch[3] or Algorithmic Justice League[4] can be included here. It is implemented in the maintenance phase by the organization to detect detrimental, decision unspecific behavior of the system. Human supervision can help detect feedback bias and feedback loops by integrating outputs from initial algorithms as inputs in future iterations and thereby reinforcing initial biases (Bellamy et al., 2018). Both internal (e.g., employees or contractors of the implementor) and external audits (e.g., civil organizations) can act as a "third line of defense" by analyzing and inspecting a software, with implementation or in hindsight, for bias or functionality, safety, and privacy issues (Raji et al., 2020). Mozilla's Open Source Audit Tooling Project aims to provide an overview of resources and tools for algorithmic auditing to hold ML system operators accountable.[5] Beyond formal auditing approaches, Shen et al. (2021) proposed the concept of "everyday algorithm auditing" as an informal auditing approach of everyday users of ML-based software to detect and raise awareness about bias they experienced when interacting with the system. For instance, Google Translate users noticed and reacted upon how the software associated certain genders with stereotypical activities and professions (Olson, 2018). By giving formal authority to supervision instruments, unfair systems can be acted upon and regulatory changes considered continuously (Reisman et al., 2018). However, external audits can suffer from access to the target, as well as public pressure and hostile corporate reaction.



	Dimension	Characteristic						
Content Level	Debiasing Meta-Strategy	Awareness and Warning	Multiple Perspectives	Open -Mindedness	Traceability	Accountability	Transparency	
	Debiasing Effect	Bias Detection	Bias Prevention	Bias Mitigation	Consequence Reduction			
	Bias Category	Machine Learning Bias	Cognitive Bias					
	Targeted Debiasing Subject	Person	Task	Algorithm	Structured Data	Unstructured Data		
Implementation Level	Implementation Phase	Planning	Analysis	Design	Implementation	Testing and Integration	Maintenance	Whole Lifecycle
	Implementation Target	Decision Specific	Decision Unspecific					
	Implementor	User	Tech Team	(Non) Tech Team	Product Manager	C-Level Manager	Organization	

Table 7. Taxonomy Application for Human Supervision

Discussion

Debiasing approaches for ML-based software has become a prominent topic at the forefront of research and is gaining increasing attention within organizational practices. Within the psychological and technical domains, scattered overviews and approaches to debiasing have been put forth. We have taken initial steps to conceptualize and classify debiasing instruments for ML-based software development. We argue that debiasing is concerned with both cognitive and ML bias in this context. Our developed taxonomy and applications shed light on relevant contributions on how to improve biased decision-making and ML-based software in business, while simultaneously hinting towards important research gaps.

Machine Learning (ML)-based software introduces novel uncertainties which motivate to revisit extant notions of bias and debiasing. Our study and developed taxonomy provide a theoretical contribution to the

[3] https://algorithmwatch.org/en/

[4] https://www.ajl.org/

[5] https://foundation.mozilla.org/en/what-we-fund/fellowships/oat/

literature of cognitive and ML biases, as well as the domain of software development. We do so by developing a debiasing taxonomy that extends extant classifications of debiasing and debiasing instruments from a ML-based software development perspective (Keren, 1990). We shed light on the relevance of technical and cognitive challenges in organizational and individual decision making (Mohanani et al., 2020). While extant theoretical notions on debiasing also seem to hold in the context of ML-based software, novel types of bias (i.e., ML bias) and novel instantiations of cognitive bias (i.e., algorithm to user) arise in such a context. More so, existing conceptualizations of debiasing such as Fischoff's (1981) distinction between individuals and tasks as a debiasing subject must be extended. As with ML-based software, data and algorithms themselves can become the subject of a debiasing instrument.

Given the potential of debiasing approaches to overcome organizational challenges and potentially hazardous repercussions, our taxonomy offers an initial proposition of how to understand, structure, and decide on debiasing for ML-based software. By considering the dimensions and characteristics of our debiasing taxonomy, we aim to guide researchers and practitioners alike in understanding the implications of specific debiasing instruments. On a basic level, our taxonomy can enable and determine high-level decision criteria for the identification of a suitable debiasing instrument in an organizational context.

Several avenues for further research arise in the context of the organizational implementation of debiasing instruments. First, it is unclear how well companies are knowledgeable and equipped in terms of debiasing. Particular debiasing instruments, such as oversight boards, seem neither to be common nor their effect to be measured. Qualitative methods such as case studies and ethnographic research could describe critical factors for organizational decision-making and deployment of debiasing instruments. Effectiveness, acceptance, and feasibility of debiasing instruments for organizational practices should be examined. Second, given the complexity of real-word organizational decision-making, IT managers should be guided in when and how to implement debiasing instruments. Bessen et al.'s (2022) study on the cost of ethical AI sheds light on how debiasing is balanced against AI innovation and competition, data access, as well as costs and resources associated with the adherence to policies or ethic guidelines. One might argue that our study takes a deontological, Kantian perspective in that debiasing is deemed as the point of departure in the case of any biased software. Inspired by the risk management process in information management (Bannerman, 2008), adopting basic strategies for risk management could help inform a management of bias and deployment of debiasing tools that are both ethically and economically desirable. Li and Chignell's (2002) proposition of a FMEA for AI systems provides a relevant approach by considering three aspects of a system's (fairness) failure that are used to calculate the risk level: (1) the severity of a failure if it occurs, (2) the probability of occurrence, and (3) the probability of detection. In that sense, the FMEA approach enables to incorporate debiasing considerations into the software design process by weighing the costs and benefits of debiasing for a certain ML application.

Conclusion

With this work, we have proposed an initial classification of debiasing instruments for the software development process. The omnipresence of published research on and detrimental consequences of bias in ML-based software points towards the relevance and importance of debiasing instruments. By structuring the taxonomy along key dimensions of debiasing content and implementation, we have proposed a software development lifecycle view on the choice and implementation of debiasing instruments for organizational use. Having drawn on three actual debiasing instruments proves the applicability of our developed taxonomy and illustrates the heterogeneity of debiasing instruments. Given the lacking theoretical embedding of the nascent research field of debiasing instruments for ML-based software, the taxonomy is intended to guide the identification of a suitable debiasing instrument for a particular use case, as well as to offer a conceptual understanding of debiasing for ML-based software specifically. It is our hope that this study will help inform future research on debiasing instruments, as well as the implementation of debiasing approaches in organization practices related to software development.

Acknowledgements

We thank the Swiss National Science Foundation for funding parts of this research (grant 192718).

References

Adams, R., & Loideáin, N. N. 2019. "Addressing indirect discrimination and gender stereotypes in AI virtual personal assistants: The role of international human rights law," Cambridge International Law Journal (8:2), pp.241-257.

Amini, A., Soleimany, A. P., Schwarting, W., Bhatia, S. N., and Rus, D. 2019. "Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society,* pp. 289–295.

Baer, T. 2019. Understand, Manage, and Prevent Algorithmic Bias. A Guide for Business Users and Data Scientists, Berkeley, CA: Apress, pp. 1–245.

Bailey, K.D. 1994. Typologies and Taxonomies: An Introduction to Classification Techniques, CA: Sage.

Bannerman, P. L. 2008. "Risk and risk management in software projects: A reassessment," *Journal of systems and software* (81:12), pp. 2118-2133.

Barocas, S., and Boyd, D. 2017. "Engaging the Ethics of Data Science in Practice," Communications of the ACM (60:11), pp. 23–25.

Barocas, S., Hardt, M., and Narayanan, A. 2018. "Fairness and Machine Learning" retrieved from http://www. fairmlbook.org.

Benbya, H., Davenport, T., and Pachidi, S. 2020. "Artificial Intelligence in Organizations: Current State and Future Opportunitiesm," *MIS Quarterly Executive* (19), pp.9–21.

Benbya, H., Pachidi, S., Davenport, T., and Jarvenpaa, S. 2019. "Call for Papers on Artificial Intelligence in Organizations: Opportunities for Management and Implications for IS Research," *Journal of the Association for Information Systems (JAIS) - MISQ Executive (MISQE),* Joint Special Issue.

Bender, E. M., and Friedman, B. 2018. "Data statements for natural language processing: Toward mitigating system bias and enabling better science," in *Transactions of the Association for Computational Linguistics* (6), pp. 587-604.

Benjamin, M., Gagnon, P., Rostamzadeh, N., Pal, C., Bengio, Y., and Shee, A. 2019. "Towards Standardization of Data Licenses: The Montreal Data License," arXiv:1903.12262 [cs, stat].

Bessen, J., Impink, S. M., and Seamans, R. 2022. "The Cost of Ethical AI Development for AI Startups," in Proceedings of 2022 ACM conference on Artificial Intelligence, Ethics, and Society (AIES'22), Oxford, United Kingdom.

Beynon-Davies, P., Owens, I., and Lloyd-Williams, M. 2000. "Melding Information Systems Evaluation with the Information Systems Development Life-Cycle," in Proceedings of the 8th European Conference on Information Systems.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. 2017. "Optimized pre-processing for discrimination prevention," *Advances in neural information processing systems* (30).

Cogwill, B., and Stevenson, M. 2020. "Algorithmic Social Engineering," in *AEA Papers and Proceedings,* 110, pp. 96-100.

Cowgill, B., Dell'Acqua, F., Deng, S., Hsu, D., Verma, N., and Chaintreau, A. "Biased Programmers? Or biased data? A field experiment in operationalizing ai ethics," in Proceedings of the 21st ACM Conference on Economics and Computation, pp. 679-681.

Collins, C., Dennehy, D., Conboy, K., and Mikalef, P. 2021. "Artificial intelligence in information systems research: A systematic literature review and research agenda," *International Journal of Information Management* (60:102383).

Correia, V. 2017. "Accountability Breeds Response-Ability: Contextual Debiasing and Accountability in Argumentation,". in P. Brézillon, R. Turner, & C. Penco (eds.), Modeling and Using Context, Springer International Publishing, pp. 127–136.

Dawson, L. M. 1997. "Ethical Differences Between Men and Women in the Sales Profession," in *Journal of Business Ethics,* 16(11), pp. 1143–1152.

De-Arteaga, M., Feuerriegel, S., and Saar-Tsechansky, M. 2022. "Algorithmic Fairness in Business Analytics: Directions for Research and Practice", in Production and Operations Management, 10.48550/arXiv.2207.10991.

Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., Williams, M. D. 2021. "Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management,* (57:101994).

Fahse, T., Huber, V., and van Giffen, B. 2021. "Managing Bias in Machine Learning Projects," in *Proceedings of 16th International Conference on Wirtschaftsinformatik (WI),* Duisburg-Essen, Germany.

Fischoff, B. 1981. "Debiasing," *Decision Research,* Eugene, Oregon. Retrieved from https://apps.dtic.mil/sti/citations/ADA099435

Fleischmann, M., Amirpur, M., Benlian, A., & Hess, T. 2014. "Cognitive Biases in Information Systems. Research: A Scientometric Analysis," in *ECIS 2014 Proceedings—22nd European Conference on Information Systems.*

Floridi, L. 2018. "Soft ethics, the governance of the digital and the General Data Protection Regulation," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (376:2133).

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. 2021. "Datasheets for datasets," in *Communications of the ACM*, (64:12), pp. 86-92.

Glaeser, J., and Laude, G. 2010. *Experteninterviews und qualitative Inhaltsanalyse: Als Instrumente rekontruierender Untersuchungen*, VS Verlag für Sozialwissenschaften.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. 2018. "A Survey of Methods for Explaining Black Box Models," ACM Computing Surveys (51:5), pp. 1–42.

Horlacher, A., & Hess, T. 2016. "*What Does a Chief Digital Officer Do? Managerial Tasks and Roles of a New C-Level Position in the Context of Digital Transformation,"* in 49th Hawaii International Conference on System Sciences (HICSS), pp. 5126–5135.

Huang, H. 2008. "A sustainable systems development lifecycle," in *Proceedings of PACIS 2008.*

Jorgensen, M, and Grimstad, S. 2012. "Software development estimation biases: The role of interdependence," *IEEE Trans. Softw. Eng.* (38:3), pp. 677-693.

Kahneman, D. 2011. *Thinking, fast and slow.* Farrar, Straus and Giroux.

Kamiran, F., and Calders, T. 2012. "Classifying without discriminating," in *2nd International Conference on Computer, Control and Communication*, pp. 1-6.

Kaufmann, L., Michel, A., and Carter, C. R. 2009. "Debiasing Strategies in Supply Management Decision Making," *Journal of Business Logistics* (30:1), pp. 85–106.

Keren, G. 1990. *Cognitive aids and debiasing methods: can cognitive pills cure cognitive ills?"* in *Advances in Psychology* (68), pp. 523-552.

Keren, G. 1997. "On The Calibration of Probability Judgments: Some Critical Comments and Alternative Perspectives," *Journal of Behavioral Decision Making* (10:3) pp. 269–278.

Kliegr, T., Bahník, Š., and Fürnkranz, J. 2021. "A review of possible effects of cognitive biases on interpretation of rule-based machine learning models," *Artificial Intelligence* (295).

Li, J., and Chignell, M. 2022. "FMEA-AI: AI fairness impact assessment using failure mode and effects analysis", in *AI Ethics*. https://doi.org/10.1007/s43681-022-00145-9

Liftiah, L., Amawidyati, S. A. G., Anto, A. H. F., Sugiarianti, S., and Zen, Y. Z. 2021. "The Implicit Memory Bias During Pandemic Covid-19 in University Students," in *11th Annual International Conference on Industrial Engineering and Operations Management, IEOM 2021,* pp. 3708-3715.

Mayring, P. 2000. *Qualitative Inhaltsanalyse*, VS Verlag für Sozialwissenschaften.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. 2021. "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys* (54:6), pp. 1–35.

Mikalef, P., Popovic, A., Eriksson Lundström, J., and Conboy, K. 2020. "Special Issue Call for Papers: Dark Side of Analytics and AI," *European Journal of Information Systems.*

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., Gebru, T. 2019. "Model Cards for Model Reporting," in FAT* '19: Conference on Fairness, Accountability, and Transparency, January 29–31, 2019, Atlanta, GAUSA. ACM, New York, NY, USA. https://doi.org/10.1145/3287560.3287596

Mitchell, S., Potash, E., Barocas, S., D'Amour, A., and Lum, K. 2018. "Prediction-based decisions and fairness: A catalogue of choices, assumptions, and definitions," arXiv preprint arXiv:1811.07867. 2018.

Mohanani, R., Salman, I., Turhan, B., Rodriguez, P., and Ralph, P. 2020. "Cognitive Biases in Software Engineering: A Systematic Mapping Study," *IEEE Transactions on Software Engineering* (46:12), pp. 1318–1339.

Nematzadeh, A., Ciampaglia, G. L., Menczer, F., and Flammini, A. 2018. "How algorithmic popularity bias hinders or promotes quality," *Scientific Reports* (8:1) pp. 1-10.

Nickerson, R. C., Varshney, U., and Muntermann, J. 2013. "A method for taxonomy development and its application in information systems," *European Journal of Information Systems* (22:3), pp. 336-359.

Nunamaker, J. F., Chen, M., and Purdin, T. D. M. 1990. "Systems Development in Information Systems Research," *Journal of Management Information Systems* (7:3), pp. 89–106.

Noor, E. 2020. "How we can ensure AI develops as a force for good rather than harm," *World Economic Forum*, retrieved from https://www.weforum.org/ agenda/2020/01/how-we-can-ensure-ai-develops-as-aforce-for-good-rather-than-harm**.**

Olson, P. 2018. "The algorithm that helped google translate become sexist", *Forbes*, retrieved from https://www.forbes .com/sites/parmyolson/2018/02/15/the-algorithm-that-helped-google-translate become-sexist/?sh=6c1d0807daa2

Olteanu, A., Castillo, C., Diaz, F., and Kıcıman, E. 2019. "Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries," *Frontiers in Big Data* (2:13).

O'Neil, C. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy,* London: Allen Lane.

Park, C. W., & Lessig, V. P. 1981. "Familiarity and Its Impact on Consumer Decision Biases and Heuristics," *Journal of Consumer Research* (8:2), pp. 223–231.

Patel, P. 2021, April 20. "Engineering Bias Out of AI > Machines that learn the worst human impulses can still relearn," in *IEEE Spectrum,* retrieved from https://spectrum.ieee.org/engineering-bias-out-of-ai.

Peterson, A. R., Beltramini, R. F., and Kozmetsky, G. 1991. "Concerns of college students regarding business ethics: A replication," in *Journal of Business Ethics*, 10(10), pp 733-738

Ragin, C. C., and Becker, H. S. (Eds.) 1992. *What is a case?: Exploring the foundations of social inquiry*. Cambridge University Press.

Raji, I.D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., and Barnes, P. 2020. "Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing," in *Conference on Fairness, Accountability, and Transparency (FAT* '20),* Barcelona, Spain. ACM, New York, NY, USA. https://doi.org/10.1145/3351095. 3372873

Reisman, D., Schultz, J., Crawford, K., and Whittaker, M. 2018. "Algorithmic Impact Assessments: A Practical Framework for Public Agency Accountability," retrieved from https://ainowinstitute.org/aiareport2018.pdf

Renner, K.-H., and Jacob, N.-C. 2020. *Was ist ein Interview?* In *K.-H. Renner & N.-C. Jacob, Das Interview* pp. 1–17. Springer Berlin Heidelberg.

Rikkers, L. F. 2002. "The bandwagon effect," *Gastrointestinal Surgery* (6:6), pp. 8.

Schlesinger, A., O'Hara, K. P., and Taylor, A. S. 2018. "Let's Talk About Race: Identity, Chatbots, and AI," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.

Shen, H., DeVos, A., Eslami, M., and Holstein, K. 2021. "Everyday Algorithm Auditing: Understanding the Power of Everyday Users in Surfacing Harmful Algorithmic Behaviors." *Proc. ACM Hum.- Comput. Interact. 5, CSCW2,* Article 433 (October 2021). https://doi.org/10.1145/347957

Shepperd, M., Mair, C., and Jørgensen, M. 2018. "An experimental evaluation of a de-biasing intervention for professional software developers," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing,* pp. 1510–1517.

Simpson, E., and Semaan, B. 2021. "For you, or For "you"? Everyday LGBTQ+ encounters with TikTok," in *Proceedings of the ACM on Human-Computer Interaction 4, CSCW3*, pp. 1–34.

Singh, R. 2016. "The Easy Way To Understand SDLC," Dignitas Digital. Retrieved form https://www.dignitasdigital.com/blog/easy-way-to-understand-sdlc/

Steffel, M., Williams, E. F., and Pogacar, R. 2016. "Ethically Deployed Defaults: Transparency and Consumer Protection through Disclosure and Preference Articulation," *Journal of Marketing Research* (53:5), pp. 865–880.

Suresh, H., and Guttag, J. V. 2021. "A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle," in *Equity and Access in Algorithms, Mechanisms, and Optimization, EAAMO*, New York, NY, USA.

Szopinski, D., Schoormann, T., and Kundisch, D. 2019. 7»Because Your Taxonomy is Worth IT: towards a Framework for Taxonomy Evaluation," in *European Conference of Information Systems*.

Teodorescu, M. H. M., Morse, L. Awwad, Y., and Kane, G. C. 2021. "Failures of Fairness in Automation Require a Deeper Understanding of Human-ML Augmentation," *MIS Quarterly* (45:

3) pp.1483-1500.

Tomalin, M., Byrne, B., Concannon, S., Saunders, D., and Ullmann, S. 2021. "The practical ethics of bias reduction in machine translation: Why domain adaptation is better than data debiasing," *Ethics and Information Technology* (23:3), pp. 419–433.

Trewin, S. 2018. "AI fairness for people with disabilities: Point of view," retrieved from arXiv:1811.10670.

Tversky, A., & Kahneman, D. 1974. "Judgment under Uncertainty: Heuristics and Biases," Science (185:4157), pp. 1124–1131.

Tversky, A., & Kahneman, D. 1983. "Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment," *Psychological Review* (90:4), pp. 293–315.

Tversky, A., and Kahneman, D. 1992. "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and Uncertainty* (5:4), pp. 297-323.

Wambsganss, T., Schmitt, A., Mahning, T., Ott, A., Soellner, S., Ngo, N.A., Geyer-Klingeberg, J., Nakladal, J., and Leimeister, J.M. "The Potential of Technology-Mediated Learning Processes: A Taxonomy and Research Agenda for Educational Process Mining," in *International Conference on Information Systems (ICIS),* Austin, Texas.

Wirth, R., and Hipp, J. 2000. "CRISP-DM: Towards a standard process model for data ming," in *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, (11), London, UK: Springer Verlag.

Wolfswinkel, J. F., Furtmueller, E., and Wilderom, C. P. 2013. "Using grounded theory as a method for rigorously reviewing literature," *European journal of information systems* (22:1), pp. 45-55.

World Economic Forum Global Future Council on Human Rights 2016–2018. 2018. "How to Prevent Discriminatory Outcomes in Machine Learning," retrieved from https://www.weforum.org/whitepapers/how-to-prevent-discriminatory-outcomes-in-machine-learning.

Van Giffen, B., Herhausen, D., and Fahse, T. 2022. "Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods," *Journal of Business Research* (144), pp. 93-106.

Vigdor, N. 2019. "Apple card investigated after gender discrimination complaints," *The New York Times* , retrieved from https://www.nytimes.com/2019/11/10/business/Apple-credit-card-investigation.html

Yang, K., Stoyanovich, J., Asudeh, A., Howe, B., Jagadish, H. V., and Miklau, G. 2018. "A nutritional label for rankings," in *Proceedings of the 2018 international conference on management of data*, pp. 1773-1776.