On developers' practices for hazard diagnosis in machine learning systems

Balayn, A.M.A.

DOI
10.4233/uuid:ea94239f-5e95-4705-9deb-32196d74daaa

Publication date
2023

Document Version
Final published version

Citation (APA)
Balayn, A. M. A. (2023). *On developers' practices for hazard diagnosis in machine learning systems*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:ea94239f-5e95-4705-9deb-32196d74daaa

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

AGATHE BALAYN

ON DEVELOPERS' PRACTICES FOR HAZARD DIAGNOSIS IN MACHINE LEARNING SYSTEMS

On Developers' Practices for Hazard Diagnosis in Machine Learning Systems

On Developers' Practices for Hazard Diagnosis in Machine Learning Systems

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on Wednesday, 4th of October 2023 at 15:00 o'clock.

by

Agathe Balayn

Master of Science in Computer Science,
Delft University of Technology, the Netherlands,
and Ingénieur,
ENSTA ParisTech, Institut Polytechnique de Paris, France
born in Paris, France

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	Chairperson
Prof. dr. ir. G.J.P.M. Houben,	Delft University of Technology, promotor
Prof. dr. ir. A. Bozzon,	Delft University of Technology, promotor
Independent members:	
Prof. dr. I. Shklovski	University of Copenhagen, Denmark
Prof. dr. P. Cudré-Mauroux	University of Fribourg, Switzerland
Prof. dr. S. Ben Allouch	University of Amsterdam, the Netherlands
Prof. dr. ir. M.S. Kleinsmann	Delft University of Technology, the Netherlands,
Prof. dr. G.W. Kortuem	Delft University of Technology, the Netherlands, reserve member

Dr. ir. J. Yang has greatly contributed to the feedback process on certain chapters of this thesis.
This research was partially supported by the HyperEdge Sensing project of Cognizant.

SIKS Dissertation Series No. 2023-24. The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Published and distributed by: Agathe Balayn

Keywords: Machine learning, Machine learning practitioners, Algorithmic harms, Algorithmic fairness, Algorithmic robustness, Algorithmic explainability, Mixed-method, Qualitative studies

Printed by: Gildeprint

Cover design by: Agathe Balayn, with the support of Dr. David Maxwell (typeset in *MADE Evolve Sans*)

Front & Back: Icarus and his wings, as a metaphor for us, our use of artificial intelligence (in pink accents), and its underlying infrastructures.

ISBN 978-94-6419-926-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Un exemple n'est pas forcément un exemple à suivre.

Albert Camus

CONTENTS

1

INTRODUCTION

Machine learning (ML) is increasingly recognized as a technology with a great potential for task automation, task acceleration, or task effectiveness improvement. In practice, ML has already shown promises in various industries and public organisations. It is now used in advertising, banking and finance, document management, security, predictive maintenance, healthcare, retail, law, agriculture, manufacturing, transportation, etc.[1] For the public sector, it is envisioned to be especially impactful for cyberdefense, traffic management, administrative tasks, real-time translation,[2] road infrastructure inspection, tax-evasion detection, etc.[3]

Next to the potential utility of ML, what explains its rapid adoption is one unique characteristics of its application process: ML is seen as one of the artificial intelligence technologies that enables a (relatively) easy development of automated software, as it primarily consists in automatically learning relevant patterns from data, instead of thoroughly identifying such patterns and manually expressing them in a formal language [414]. The growth of ML has been fostered by a variety of seminal research papers proposing new algorithms to train ML models, such as ridge regression [360], classification trees [130], generalized additive models [340], support vector machines [345], random forests [129], etc., and later on new architectures for deep learning models. These models work with a variety of input data, e.g., tabular data, images, text, video, etc., to perform a variety of tasks, e.g., classification, regression, detection, segmentation, etc., corresponding to the needs of various domains. One can for instance think of Word2Vec embeddings [550] or the BERT model [430] for natural language processing, the AlexNet model [453] for computer vision, or the deep Long Short-term Memory recurrent neural networks [306] for speech-to-text applications, etc.

[1] https://www.grandviewresearch.com/industry-analysis/machine-learning-market

[2] https://wp.nyu.edu/dispatch/5-examples-of-using-ai-deep-learning-for-the-government-and-public-sector/

[3] https://www.mckinsey.com/industries/public-and-social-sector/our-insights/when-governments-turn-to-ai-algorithms-trade-offs-and-trust

1

1.1. MOTIVATION: ML HAZARDS, FROM ML THEORY TO ML PRACTICES

Despite the potential of ML, ML also suffers from hazards that can cause or reinforce harms. These ML hazards are at the core of the motivation for this thesis, as we describe in the remaining of this section.

1.1.1. DANGERS OF ML MODELS: FAILURES, HAZARDS & HARMS

One of the primary arguments for the adoption of ML to perform certain tasks in a specific application context is the advantages it brings, in comparison to humans working for this application. Oftentimes, ML is argued to be more accurate, and less biased than human decision makers, making the decision-making task at hand safer and fairer — "If you want the bias out, get the algorithms in" (MIT research scientist Andrew McAfee [752]). The proposed arguments are respectively that humans are not able to process as much data (and in a same amount of time) as a machine in order to identify and take the most appropriate action, and that humans are all biased and make decisions based on biased judgements, whereas a machine would be more objective and bring consistency to the decisions.

Unfortunately, these arguments have revealed flawed in recent years. A plethora of incidents and accidents caused by ML-powered systems have led to identify many hazards and harms of ML. To name a few, we can cite discrimination in allocation task [546], offensive representations in classification tasks[4], denying the principal of individual justice [254], insecurity (e.g., in relation to adversarial attacks) [610], safety issues due to false positives or false negatives (e.g., accidents in autonomous driving or healthcare), privacy infringement (e.g., datasets collected without informed consent), unnecessary cost, environmental impact due to data storage and computational power required, etc.[5] [490, 240, 120] And of course, not all harms are known.

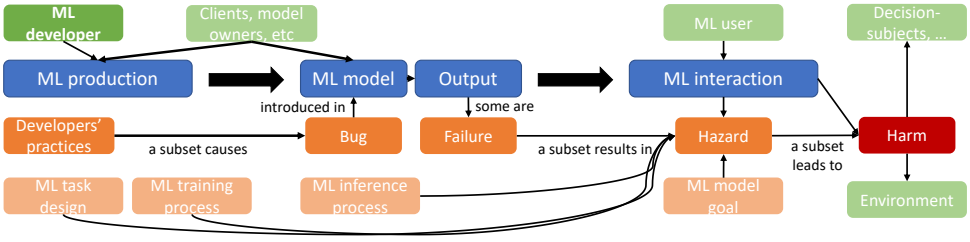


Figure 1.1: Clarification of the terminology used in this thesis around the concepts of *harm*, *failure*, and *hazard*. In dark orange, the type of hazards (and their causes) we primarily focus on (including in Part III), and in light orange those hazards for which we leave to future work the development of diagnosis methods. In blue, we show how hazards and their causes relate to the different components of a machine learning system. In green, we illustrate which stakeholders typically intervene on these components, we primarily focus on the machine learning developers (dark green), as they are those stakeholders whose technical practices might lead to harms.

[4] https://www.theguardian.com/technology/2023/feb/08/biased-ai-algorithms-racy-women-bodies

[5] https://www.w3.org/TR/webmachinelearning-ethics/

In Figure 1.1, we clarify the terminology that we will use along the thesis. The examples above constitute examples of *algorithmic harms*, i.e., harms caused by the development or use of an ML system in the real world. Before a harm arises, we talk about *hazard*, i.e., the risk that a harm could arise in the world. In the thesis, we investigate two types of hazards: a) hazards that find their cause in a dangerous *output failure* of the ML system (i.e., a hazardous output failure, such as those that lead to discrimination), the failure itself being caused by a bug in the system that results from problematic ML developers' practices (e.g., a problematic design choice about the training dataset or the architecture of the ML model); or b) hazards that are not due to the system output but to its inherently problematic goal, or due to its development and deployment processes (e.g., environmental impact). Hazardous output failures are the issues for which we develop diagnostic solutions in Part III. In this thesis, we focus on the technical elements (failures, bugs, and design choices), that when unaccounted for, might transform a hazard of type a) or type b) into a harm. In Figure 1.2, we provide a more concrete example of the way in which a problematic practice of an ML developer might result into a harm.

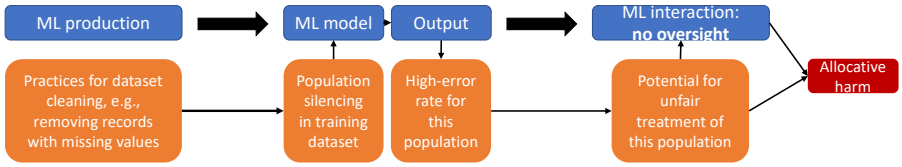


Figure 1.2: Example of the way in which one flawed practice of an ML developer might result into a harm. Imagine a use-case where an ML developer builds an ML system to allocate welfare benefits across individuals.

EXAMPLES OF HARMS COMING FROM HAZARDOUS MODEL OUTPUT FAILURES

Across application domains, the use of ML systems has caused many harms, among which many (more than 2400 at the time of the writing of this thesis) are referenced in the AI Incident Database.[6] We give here a few examples of these harms, that are specifically related to failures in the outputs of ML-powered systems.

Discrimination. In relation to tabular data, ProPublica[7] has shown that the COMPAS system used in the United States of America to predict recidivism risk discriminates against African Americans in comparison to white defendants. In the Netherlands, the child care benefits scandal ("toeslagenaffaire"[8]) revealed that the ML system used by the Dutch government to spot benefits fraud inaccurately flagged families from ethnic minorities and lower-income families as suspicious of frauds, leading to exorbitant debts. Around natural language processing, the Amazon automatic screening tool[9] for hiring

[6] https://incidentdatabase.ai/
[7] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
[8] https://www.politico.eu/article/dutch-scandal-serves-as-a-warning-for-europe-over-risks-of-using-algorithms/
[9] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

1

has been discovered to penalize women candidates for a job in comparison to men candidates. As for computer vision applications, investigations have shown highly unequal errors rates between intersectional groups within facial attribute classification systems (especially between darker-skin color women, and lighter-skin color men) [138], offensive labels outputted to characterize darker-skin color individuals, such as with the infamous Google Vision API that incorrectly described certain individuals as "gorillas"[10] or a dark-skin hand holding a thermometer as holding a gun[11], etc.

Physical harms. The above examples reflect ML system outputs that cause discrimination and have strong harmful social impact on individuals or communities especially when related to resource allocation or erroneous representations. System outputs have also led to safety risks and physical accidents (without association to discrimination). Countless examples show misclassifications of ML algorithms that led to errors in medical diagnoses [527], sometimes with strongly imbalanced error rates across categories of population [593]; misrecognitions that could have led to car accidents, or actually led to injuring pedestrians[12][13], etc.

CAUSES OF THESE HAZARDS & HARMS

The hazards potentially resulting in these harms find their causes in a multitude of intricate design choices made to develop and exploit ML-based systems [546, 28].

- **Dataset design.** One of the first issues resides in the design of the training datasets for the ML models underlying the systems, that bear many limitations. Among these limitations, the most frequently cited are the following. It is now well-understood that the data on which a model is trained contains historical human biases, and hence a model trained on this data reproduces and oftentimes amplifies these biases, causing unfairness in its outputs or output errors that might lead to safety risks [687]. Besides, it is also now clear that models pick up on spurious correlations contained in these datasets [785], leading to incorrect and over-simplified inference mechanisms, that again lead to safety risks. Distribution shifts [448] between the training data and the data the model sees in deployment are also one of the main causes of these outputs failures, as the model has not been able to learn to make correct predictions on types of data it has never seen before.

- **Model design.** The construction of the model itself can also cause errors in the outputs, e.g., due to shortcut learning [284] or in-adapted objective functions [28].

- **Model interaction design.** The way one interacts and uses the outputs of a model is also a source of errors, e.g., when one might under-rely, or over-trust the outputs of the model [452].

[10] https://www.bbc.com/news/technology-33347866

[11] https://algorithmwatch.org/en/google-vision-racism/#:~:text=In%20an%20experiment%20that%20became,was%20labeled%20%E2%80%9Celectronic%20device%E2%80%9D.

[12] Accidents related to Tesla cars: https://www.theguardian.com/technology/2022/oct/26/tesla-criminal-investigation-self-driving-claims-sources

[13] Accidents related to Uber cars: https://www.bbc.com/news/technology-54175359

- **Task design.** Needless to say, the application for which the model might be used can also be considered problematic and harmful in itself by certain populations, such as in Iran where the authorities envision the use of facial recognition technologies in order to enforce the hijab law on women[14].

In this thesis, we especially investigate dataset and model design practices, and develop solutions for developers to better diagnose hazardous failures of ML-based systems resulting from those dataset and model design choices (Part III). These problematic design choices are themselves due to a variety of reasons that we also explore further, especially in Part II, and that had been rarely explored at the time of conducting the research for our thesis. Of course, research has yet to identify, understand, and develop mitigation methods for all these hazards [369], and it is not known whether ML developers are able to handle them without the inputs from research. Besides, even when hazards are fairly well-understood by the research community, there might be other obstacles for the developers to handle them, that we study further. For instance, in the literature and in our studies, we identify organisational and business pressures [666], a lack of education about these problems [743, 662], and a risky attitude from the developers [220], e.g., not feeling responsible for the problems, or not considering their design choices potentially hazardous.

1.1.2. FROM GUIDELINES & THEORY TO PRACTICES AGAINST HAZARDS: THE MISSING LINK

We identify several efforts aimed at mitigating potential hazards and harms. A plethora of documents have emerged, all aiming at tackling the problem by design, especially numerous ML ethics guidelines [412] from companies and public institutions, and new regulations such as the General Data Protection Regulation[15] (GDPR), the Artificial Intelligence Act[16] (AI Act), or the Digital Services Act[17] (DSA) in the European Union. In the realm of trustworthy ML/AI ethics research, countless research papers that aim at developing methods for developers to identify and mitigate these issues have been published [153], alongside a number of papers that propose documentation and checklists [283, 556] for ML developers to further reflect about the potential harms of the systems they build (we draw an overview of these research publications in the next section). Yet, "AI ethics is failing in many cases" [326]. Despite the lack of precise estimates of the number of accidents caused by ML every year, it does not seem that the number of these accidents is decreasing, especially when looking at the recently-released systems, that continue perpetrating various harms, such as ChatGPT[18] for text generation, or DALL-E and Stable Diffusion for image generation[19].

We argue further in this thesis that one of the primary reasons for the number of harms not to decrease despite all the policy and research efforts deployed is the lack of

[14] https://www.theguardian.com/global-development/2022/sep/05/iran-government-facial-recognition-technology-hijab-law-crackdown
[15] https://gdpr-info.eu/
[16] https://artificialintelligenceact.eu/
[17] https://digital-strategy.ec.europa.eu/en/policies/digital-services-act-package
[18] https://www.insider.com/chatgpt-is-like-many-other-ai-models-rife-with-bias-2023-1
[19] https://techpolicy.press/researchers-find-stable-diffusion-amplifies-stereotypes/

understanding about practices of the developers of ML models. These practices might bear limitations as developers are not necessarily *ethical unicorns* [662] and might face a diversity of challenges (as hinted above). Yet, these limitations remain overlooked. Overlooking these limitations in turn results in mis-aligning current policy and research efforts with the real needs of ML developers. Hence, the understanding and analysis of practices of these ML developers appears as the missing link between harms and the development of appropriate directions to diagnose and prevent the system failures causing harms.

1.1.3. GOAL OF THE THESIS: MITIGATING HAZARDS

In this thesis, we aim at contributing to the growing body of knowledge and methods related to the development of trustworthy ML technologies that would be less hazardous and harmful. Especially, we give ourselves two objectives:

- Characterizing the misalignment between ML research tackling questions of hazardous output failures and harms, and practical efforts ML developers make towards tackling the hazards of ML.
- Developing and thoroughly evaluating model explainability-based, technical, and methodological support for ML developers to tackle hazards of ML, specifically to diagnose model output failures, based on the most urgent types of misalignment identified.

In the remaining of this chapter, before explaining our mixed-method approach towards achieving our objectives, we present an overview of related works, that should give the reader a better understanding of the state of the discipline, its main research directions, the position of our work within this body of literature, the relevant research areas we get inspiration from, and the scoping of our work. Finally, we list our research questions and contributions.

1.2. POSITIONING OF THE THESIS IN THE ML LANDSCAPE

Facing the breadth of ML research and the breadth of our goal, in this section, we position our work within existing literature, and we explain how we scope down our goal in relation to insights from existing literature.

1.2.1. ML, A FIELD ACROSS THEORY & PRACTICE

Machine learning (ML) is a technology that has been studied from various perspectives. We now outline these perspectives as a background for our own research work, and especially discuss the areas the closest to our work (as summarized in Figure 1.3). We differentiate between research on ML theory and research on ML practices. In terms of theory, technical research aims at developing theories and algorithmic tools for developers to tackle system failures and harms during the ML lifecycle, and interdisciplinary research aims at characterizing the harms that ML might cause and at analyzing proposed technical theories especially in terms of conceptual limitations. Socio-technical research on ML practices typically investigates how various stakeholders conceive and handle the harms that are theorized and formalised in the technical research, and the limitations of these theories. We are especially interested in the research that revolves around developers' practices. While we acknowledge the ambiguous and pervasive frontiers between the different research fields and research communities, for the sake of simplicity, we present only a brief overview of the relevant research areas.

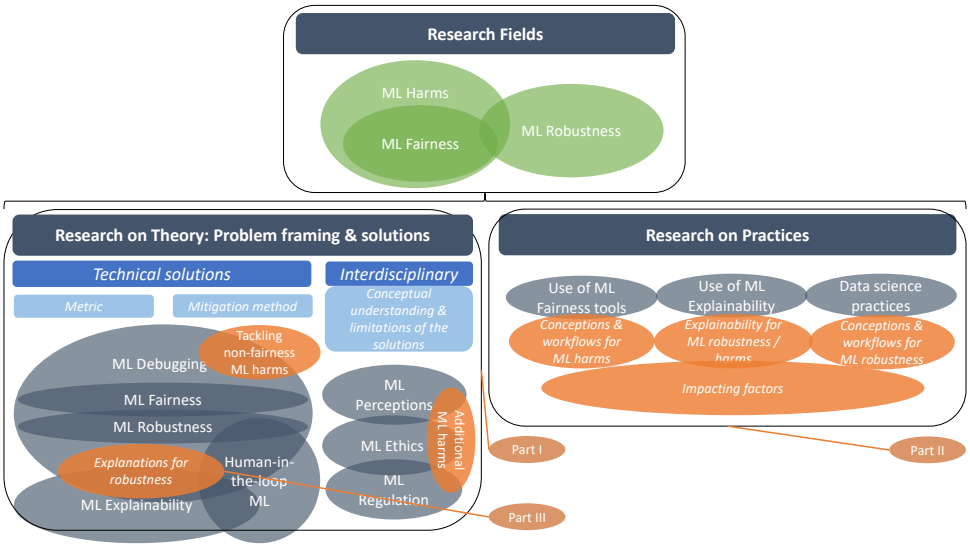


Figure 1.3: Summary of the related works in the area of machine learning failures and harms. We operate a clear distinction between the theoretical works and the works investigating practices. We also distinguish between the technical works (metric and mitigation methods), the socio-technical ones aiming at understanding the problem and potential limitations of the technical work, and the other socio-technical works that study the practices with regard to these prior works. We emphasize in grey the areas where work has been conducted, and in orange the areas where clear research gaps reveal. Those gaps in italic are the ones we tackle, while the others are left for future work.

1

MACHINE LEARNING THEORY, AS A TECHNICAL OBJECT OF ENQUIRY

Many works in ML are stemming from a technical perspective. Their goal is typically to improve the technology in terms of output performance (e.g., improving the correctness of the outputs) [407, 378] and process performance (e.g., improving the inference speed or training time of the technology, reducing the amount of data needed to obtain a given output accuracy, etc.) [540] or transparency [573, 44], as well as to develop new capabilities (e.g., the expansion from classification tasks to regression tasks, recommendations, etc.) [212]. Such works are performed either in an application-agnostic manner, or for specific, novel, domains of application (e.g., agriculture [496] or medicine [664]). Strongly connected to these algorithmic works is progress in terms of hardware that is necessary to power these algorithms, such as for the computational power necessary to train the algorithms and build models (e.g., development of CPU, GPU, and NPU for fast and large computational power), and for the memory storage required for saving and loading the high amount of data samples on which the algorithms are trained (e.g., development of larger memory storage, etc.) [96, 341]. Facing the fast expansion of the field, researchers have specialized in different application areas (cf. Figure 1.4), reflected by the different conferences in which scientific papers are published, such as computer vision based on image and video data (e.g., CVPR, ICCV), natural language processing based on text data (e.g., EMNLP), general ML primarily relying on tabular data (e.g., ICLR, NeurIPS), recommender systems (e.g., RecSys), etc. Below, we describe some of the most relevant research areas for the goals of our thesis.

The Thirty-Sixth Annual Conference on Neural Information Processing Systems (NeurIPS 2022) is an interdisciplinary conference that brings together researchers in machine learning, neuroscience, statistics, optimization, computer vision, natural language processing, life sciences, natural sciences, social sciences, and other adjacent fields. We invite submissions presenting new and original research on topics including but not limited to the following:

- General Machine Learning
- Deep Learning (e.g., architectures, generative models, optimization for deep networks)
- Reinforcement Learning (e.g., decision and control, planning, hierarchical RL, robotics)
- Applications (e.g., speech processing, computer vision, NLP)
- Machine Learning for Sciences (e.g. biology, physics, health sciences, social sciences)
- Probabilistic Methods (e.g., variational inference, causal inference, Gaussian processes)
- Optimization (e.g., convex and non-convex optimization)
- Neuroscience and Cognitive Science (e.g., neural coding, brain-computer interfaces)
- Theory (e.g., control theory, learning theory, algorithmic game theory)
- Infrastructure (e.g., datasets, competitions, implementations, libraries)
- Social Aspects of Machine Learning (e.g., AI safety, fairness, privacy, interpretability, human-AI interaction, ethics)

Machine learning is a rapidly evolving field, and so we welcome interdisciplinary submissions that do not fit neatly into existing categories.

Figure 1.4: Call for papers for the 2022 NeurIPS conference.[20]

[20] https://nips.cc/Conferences/2022/CallForPapers

Machine Learning Diagnosing & Debugging. Diagnosing and debugging ML [150] is the idea of developing methods to identify and characterize failures of ML models and their causes, and methods to mitigate these failures —depending on the failures, they might be considered hazards of ML. A failure of an ML model corresponds to the condition when the behavior of a model is not aligned with the expectations one has for this behavior. A model's behavior can relate to a large diversity of properties, be it properties related to the type of outputs that are expected (e.g., accurate outputs, fair outputs), or properties of the process related to obtaining the model (e.g., short training time, privacy-preserving training datasets) or its outputs (e.g., inference speed, robustness to adversarial attacks). Typically, an unexpected behavior is due to various kinds of bugs in a model, be it code implementation bugs (the script does not execute properly), bugs in the design of the ML pipeline (e.g., too small training dataset for the size of the model architecture, in-adapted loss functions or architecture hyperparameters), or bugs due to wrong translation between the intended design and the code implementation. While traditional software debugging has primarily focused on the first type of bugs, technical ML research primarily focuses on the second type, and human-computer interaction has recently investigated the last type of bug (cf. Part II Chapter 5 for more details).

In terms of technical research, ML debugging encompasses research on multiple, interdependent topics. ML testing refers to "any activity aimed at detecting differences between existing and required behaviours of ML systems" [924], and the research in this area proposes various methods to identify failures of the models according to various output or process objectives. Among those objectives, ML robustness [670] has become one sub-field in itself where metrics and mitigation methods are proposed for measuring and avoiding brittleness of the models to various adversarial and natural perturbations that might cause output errors and harms. Model certification is the idea of formally ensuring certain properties of a model outputs or inference process [218], where research spans the development of formal model specifications and formal proofs to ensure that the model adheres to the desired properties. Model verification of certain model (safety) properties [628] in turn proposes formal specifications of these properties and develops method to test them efficiently. These research areas are now also joined with research from the data management field, typically to identify dataset bugs [303], or problematic design choices in the data engineering pipeline [514].

In this thesis, we focus primarily on ML robustness (the ML developers we interview in Part II are not concerned at all with model certification and verification), that is a first step towards developing less harmful models (before actually certifying these models). We especially investigate state-of-the-art approaches proposed in research, and the challenges developers face when tackling potential issues with natural perturbations (we leave for future work adversarial perturbations).

Machine Learning Explainability. Closely related to ML debugging are the two research fields of ML explainability and ML fairness.

As ML is a technology whose development and use involve a great diversity of stakeholders, it is argued —and even required by regulations (e.g., the GDPR)— that ML models should be explainable [101]. For the practitioners developing these systems, having explanations about the model can allow for more effective debugging [100]. For a user

of these systems, an explainable model can allow for better trust calibration and trustworthy human-ML collaborations [827]. For a decision subject of a model, explainability can serve to trust and accept decisions, or to ask for recourse [423]. For a model owner or external auditor, explainability is necessary in order to judge on the readiness for deployment of the model [511]. Explainability research currently focuses on developing methods to extract post-hoc explanations on the functioning of a model, or to build models that are inherently explainable [140] (cf. Part III Chapter 10 for a more detailed account of existing explanability methods). This is challenging due to the diversity of models that exist, and because the faithfulness of an explanation is both hard to evaluate (no ground truth exists) and difficult to ensure [943, 232].

In this thesis, we investigate further to what extent developers make use of explainability to avoid hazardous failures and harms, and develop a new explainability method and user-interface to support developers further in their activities.

Machine Learning Fairness. It is now well understood that ML models might cause discrimination and unfairness because of biased outputs (one type of output failure) [546]. The field of ML fairness has hence developed in order to propose fairness metrics to evaluate the potential social impact of the ML models [848], and unfairness mitigation methods to develop models without harmful social impact [268]. There exists to date more than 30 fairness metrics that aim at measuring different kinds of social impact and build on top of each other to address the limitations in the modeling of the problem. A large number of mitigation methods also exist, that address transformations of the training dataset, of the model architecture or training procedure, or of the post-processing functions for the model outputs (cf. Part I Chapter 4 for a more detailed account of the research on this topic). These methods still remain limited in that they cannot yet be applied to all types of ML models that have been developed to date, and they do not allow for building entirely fair models [98, 369].

In this thesis, we investigate how developers perceive and use these methods in an effort to avoid algorithmic unfairness and other harms caused by ML models. Both the fields of explainability and fairness present a predominance of technical works, however, as we explain in the next subsections, these works are often argued to be necessary but insufficient towards preventing all hazards of ML. This explains the need for the interdisciplinary and socio-technical research we will discuss.

Human-in-the-Loop Machine Learning. Human-in-the-Loop (HIL) ML is a research area that leverages the latest advancements in crowd computing knowledge and infrastructures in an effort to solve some of the problems that ML suffers from [746]. Initially, Human-in-the-Loop ML was especially relevant for decreasing the number of annotations needed to train a ML model (active learning) [900] or to increase the quality of the labels used to train the models [237], as these were the two obvious sources of failures and components of the ML pipeline where human capabilities could be leveraged. It is now also increasingly used for debugging ML models. One can think especially of the identification of unknown unknowns of ML models [741, 334], where human capabilities are especially relevant to identify meaningful human patterns of data samples where the model might be wrongly confident. Even more recently, human-in-the-loop tech-

niques are combined to explainability methods, as the natural next component of the ML pipeline where humans are useful [824]. They can interpret explanations to identify potential model bugs, leading to an easier formulation of solutions to these bugs and improvement of the models.

In this thesis, we rely on this research area to develop a more interpretable explainability method for developers to diagnose their ML models and avoid harms.

MACHINE LEARNING THEORY, AS AN INTERDISCIPLINARY OBJECT OF ENQUIRY

ML is a technology that is increasingly introduced into society. Similarly to any other technology, it can have not only a positive, but also negative impact on society. The harms caused by ML cannot be understood with a sole technical lens on the technology. It is also necessary to adopt a socio-technical lens to scrutinize this technology, understand its potential impact on populations and the environment, and propose well-informed and appropriate solutions to this impact. This has recently given rise to new interdisciplinary conferences, such as the ACM Conference on Fairness, Accountability, and Transparency (FAccT), and the AAAI / ACM Conference on Artificial Intelligene, Ethics, and Society (AIES), whose research focus is often broadly referred to as responsible AI, trustworthy AI, or AI ethics research directions. There, philosophy [104], ethics [907], and science and technology studies (STS) [312, 109] scholars have especially aimed at characterizing the harms caused by ML models and ML research trends, and surfacing the various philosophy theories that underlie proposed fairness metrics or mitigation methods. They have also investigated what a good explanation is for various stakeholders. Legal scholars [89] and policy makers have investigated the ins and outs of regulations to understand their fit to the hazards of this new technology, and proposed new regulations for a more responsible use of the technology, referring to technical, algorithmic fairness and explainability, solutions.

Especially relevant to our thesis are the works that draw upon prior research in other research fields, to surface and characterize the harms and other hazards caused by the outputs of ML models, but also by their production. Scholars have taken inspiration from the way hazards and risks are modeled and avoided in other fields such as aviation and related this to ML to envision development guidelines, regulations, or audits [661]. They have also looked into research on human values and drawn parallels for ML in order to identify which values might be infringed [184, 915]. Many of these works have insisted on the conceptual limitations of ML fairness frameworks to represent, measure, and mitigate the harms caused by ML models, by looking at fields like political philosophy as comparison points. For instance, it has been argued that existing fairness metrics focus solely on statistical properties of the outputs of the models, leaving out of consideration other justice criteria such as an individuals' entitlements to a fair procedure [310]. Certain works have also taken a broader lens on the systems, for instance pointing out to the poor labor conditions of crowd workers annotating training datasets [393], or denouncing the environmental impact of large models [93, 464]. Meta-research works have also reflexively investigated how ML research is conducted to re-orient it further, e.g., in terms of limitations and poisonous trends current model benchmarks foster [659].

In this thesis, we survey the understanding of harms and limitations of existing technical solutions, and later on investigate the practices of ML developers surrounding these harms and the current limitations of technical solutions.

MACHINE LEARNING PRACTICES, A SOCIO-TECHNICAL OBJECT OF ENQUIRY

Next to these conceptual works on ML hazards, another, recent, socio-technical research direction stemming from human-computer interaction and science and technology studies researchers consists in investigating the practices of those developing the systems [569, 620, 346], and in the development of non-technical solutions to support further these practices [243, 283, 31]. These works have investigated the challenges for developers to achieve ML fairness [220, 481, 369], or explainable models [497, 244], discussed a number of design opportunities to remedy to these challenges, and developed and tested a number of solutions. Oftentimes, these solutions do not require high technical complexity, but are geared towards socio-technical reflections that are rarely usual for these developers. For instance, works have discussed documentation of the models, but also of the training and test datasets [283], guidance frameworks to build less harmful models [663], or reflection frameworks to inspire the identification of failures or design model interactions that allow a more appropriate usage of these models [243]. Other works, thanks to their critical perspective, have also identified a number of hazards and challenges (organisational, historical, business incentives) that do not necessarily call for design changes, but imply broader structural changes in terms of regulations on the use of ML, and in cases prompting not to deploy models [666, 82].

In this thesis, we continue on this line of work by studying practices of ML developers and proposing new solutions for them. We fill in the gap around diagnosing and debugging practices (most works focus on the use of explainability methods by developers without looking at the broader context of debugging). Besides, differently from these works, we acknowledge the conceptual limitations of existing technical solutions, and investigate how developers work with those limitations.

1.2.2. SCOPING THE RESEARCH

As the hazards of machine learning (ML) are numerous, it is not possible in a single thesis to fully address how to make ML models less harmful. Hence, we explain below how we scoped down our work, in terms of problem targeted and its breadth, in comparison to existing ML research directions outlined earlier. Table 1.1 summarizes the scoping of our research.

SCOPING THE PROBLEM TARGETED

Types of hazards, harms, failures, and bugs for which we study practices. In terms of harms, we only investigate the ones that are not caused voluntarily (e.g., not adversarial attacks), and that the technical research discusses as being observable primarily from within the outputs of a model (e.g., not from ML processes, such as security and privacy). That is, the failures we investigate revolve around undesired outputs, in terms of the unfairness they might cause [546], and in terms of physical harms that might result from other output inaccuracies, especially from brittleness to natural perturbations [448]. Natural perturbations correspond to any shift in the data distribution between the training-test data, and the production data on which the model makes inferences. Such perturbations are typical of ML applications as it is oftentimes impossible for developers to collect training datasets in the same conditions as the production data are captured to input to the models. As they can be dangerous —they typically lead to drops

Table 1.1: Summary of the scope of our thesis in terms of goals, proposed solutions, and future relevant research directions.

Dimension	Our focus	Other directions
Types of harms	Primarily harms that are caused by problematic outputs from a ML model.	Other harms do not come from the outputs of a model, but from, e.g., the way the model is produced, the way individuals are represented in the data, and by the application of ML itself to the intended task.
Bugs	Wrong configuration of the ML pipeline.	Faulty implementation, faulty translation between intended configuration and code implementation.
Borders of the system investigated	Data collection pipeline, data engineering pipeline, and model engineering pipeline; from model inputs to model outputs.	Infrastructures sustaining the pipelines and models; design of the model requirements before building the pipelines, usage of AutoML systems for system development; interaction between the model outputs and their usage by, e.g., decision makers.
Stakeholders	ML developers, and their need for interaction with domain experts and model owners.	Domain experts, model owners, model users, decision/data subjects, and broader impacted stakeholders (e.g., families of the decision subjects).
ML lifecyle	Development stage.	Deployment, production, monitoring, updating.
Domain of application	No particular domain.	Repeating similar analysis within specific domains would lead to additional reasons for the research/practice gaps, and additional needs.
Types of data, model, and application	Deep-learning based image classification tasks; tabular-data based classification and regression tasks.	Other types of data (e.g., text, video); other types of models (e.g., newest deep learning models such as ChatGPT); other types of tasks (e.g., recommendation, generation, translation).
HCI research goal	Surfacing practices around fairness and robustness; developing and evaluating usable explainability tools for model debugging.	Investigating methods for eliciting harm-related requirements.
Goal of our technical solution	Harm discovery and characterization.	No investigation of appropriate metrics for harms; no investigation of the link between output errors and the harmful impact they might have; no development of fully automated methods to identify model harms and their causes; no development of method for mitigating harms.
Technical & design solutions	Development of explainability method and supporting tool.	Besides explainability methods, other methods are proposed for model debugging and fairness, that could all be benchmarked and made more usable.
Others	Policy implications [62].	Educational, cultural, regulation, factors.

in accuracy— but rarely researched, they are urgent to tackle. They are highly connected to unfairness issues in the outputs of models, as these often result from a distribution shift between the training data and the ideal, “fair” or “unbiased” data distribution (if this exists) [796].

In practice, the conditions in which hazards become harmful cannot be understood solely within the outputs of the model, but only come to be in relation to the way the model is used in its environment. For instance, certain failures in the outputs of the models might not be harmful depending on the mode of human-ML collaboration [747], such as when a decision-maker oversees the outputs and can detect certain types of failures; or when some failures –false positives or false negatives– are less dangerous than

others for the application [418, 725], e.g., if someone without disease receives extra-care because of being wrongly identified as sick, this might be less impactful than for someone actually sick who does not receive this care). Besides, the bugs causing these harms might manifest early in the ML lifecycle, e.g., unfairness might be observed from the outputs of a model, but it can already arise from the features selected to train the model and to make inferences on new data samples [315]. Because of this intricate nature of hazards, harms, and their causes, we study together hazards that are typically mentioned around model outputs, but we account for their complexity in terms of sources and real impact. For instance, while explainability is often argued to provide a useful set of methods to identify model output errors without considerations around unfairness in mind, we argue that it is useful also for unfairness as it allows to identify model features that are potentially causing the outputs unfairness. That is to say, in ML, there is no clear distinction between the failure, its harmful impact, and the bugs causing it, and hence we cannot study them in isolation.

Types of causes of hazards for which we develop a solution. We cannot develop a solution for all types of output issues, nor for all the obstacles that are faced by developers and that we identify when studying practices. For instance, developers face challenging organisational factors (typically related to the model owners and model clients) when building ML systems, but we do not investigate deeply solutions to these factors. In this thesis, we focus on a fundamental, technical problem, when developing solutions: model diagnosis during model development, i.e., identifying and characterizing hazardous failures and their sources. We do not propose solutions to mitigate these failures, this would merit another thesis. Specifically, we tackle hazardous failures coming from model outputs and from problematic features.

Besides, as we detail in the list of research questions tackled, in terms of possible solution, we especially focus on explainability algorithms as a potential helper for model debugging against certain types of output issues. We acknowledge the existence of other technical methods towards debugging (e.g., the automatic machine learning testing methods mentioned earlier), but cannot tackle all of them in a single thesis. We choose to focus on explainability because of the amount of research it currently receives, and its arguable advantages, e.g., in terms of transparency it brings, and in terms of potential for identifying harms related to various output failures and their underlying bugs.

ACKNOWLEDGING THE BREADTH OF THIS PROBLEM

Stakeholders & domains of application. In terms of stakeholders, we primarily focus on the practitioners developing the ML models and data science pipelines (e.g., data engineering pipelines). We discuss the primary interactions reported by these ML developers with other stakeholders, however, we do not extensively discuss challenges for each of these stakeholders, except in Part III where we delve deeper into supporting the interaction with the domain experts, as this appeared to be one of the main challenges for developers to debug their models.

Stages of the machine learning lifecycle. The ML lifecycle is extremely broad, and diverse across teams and organisations [24]. Its representations also vary across research

papers and research communities. For instance, while some focus on the core technical activities of the lifecycle, others look more broadly also at the aspects related to socio-technical aspects [920]. In our work, we primarily focus on the pre-deployment phase. We consider this phase the first crucial phase to avoid harms from happening, as it is after this phase that the model is put into production and harms can happen. We acknowledge still that many additional phases happen after the development of the initial model, but have not received as much attention as the development.

Within the pre-deployment phase, despite the data-centric approach currently advocated [42], we do not focus on data solely, —or model specifically as often done in the purely technical works—, but look at both. Indeed, we note that in practice, different practitioners might adopt a strong focus on one or the other, or consider them together, especially due to the highly iterative process of the ML pipeline development (cf. Part II). We also discover later on (cf. Part III Chapter 10) that developers can use explainability methods both as a mean to investigate data bugs, but also as a tool to investigate model training bugs, or bugs due to a wrong combination of dataset and model design (e.g., overfitting). Hence, it is not possible to study them independently.

Data and model type. Finally, as mentioned above, ML is a vast domain, with many different applications areas and specificities of the technology across data types. While we do not operate a strong distinction between these types of model, we primarily focus on models for computer vision applications. We make this choice because, while a large breadth of technical works apply to these models, socio-technical works have conducted much fewer investigations on them. Hence, we want to address this gap in the literature. Yet, we note many parallels between research on computer vision applications specifically and tabular data works, hence we survey the literature on both as it can bring sources of inspiration. Finally, as developers typically envision issues of physical safety relatively easily for computer vision models, but issues of social harms more easily for tabular data applications, we also interview developers working on tabular data in relation to social harms, to still get a good understanding of their challenges and needs towards social harms.

1.3. RESEARCH METHODOLOGY: DEVELOPER-CENTERED ML RESEARCH

There appears to be a misalignment between research and practice, as the former increases but the number of accidents in practice does not decrease [40]. Better understanding this problematic phenomenon naturally calls for a better characterization of practice on one side, and of current research lines on the other side, in order to characterize the (mis)alignment, and the subsequent hazards and harms. Understanding practices in turn calls for research questions, contributions, prior information, and methodologies that we should borrow from other well-established fields, such as philosophy, sociology, law, and design.

The problem of systems creating new harms or reinforcing existing harms, and especially identifying, characterizing, and mitigating these harms, is not purely technical, but *socio-technical*. While physical harms might be considered less debatable and easily identifiable, what is a non-physical harm is a subjective question. Non-physical harms can only be understood by investigating the complex societal or economical impacts of the systems. Hence, one should critically investigate current research lines and practices, bringing in insights from other fields, while acknowledging the socio-technical nature of the problem, instead of assuming a clear, well-understood, and fixed idea of harm, that could be easily, once and for all, translated into a mathematical formulation on which one can simply propose optimization methods or evaluation procedures. This complexity and subjectivity of the problem calls for *interdisciplinary work*, for instance borrowing existing philosophical concepts and theories to envision the potential harms caused by the systems, and to understand the moral philosophy theories behind developers' conceptions of these harms, and bringing the few existing regulations to identify additional harms.

In conjunction with conceptually understanding the problem of harms and the ways in which developers envision them, one has to first, actually surface the developer's practices and conceptions of harms. This asks for adopting qualitative methods from fields such as sociology.

Once the problem is understood conceptually and in terms of developers' practices, in order to propose solutions, one should adopt a rigorous approach with a *sound design of the solution and a rigorous evaluation*. This brings the need for additional, qualitative methods, borrowed from other fields, such as design, to both design and evaluate the solutions. Evaluation could also call for quantitative work. Yet, this is not always feasible as the population sampling of ML developers working on specific types of models and that would accept to participate to our studies is not necessarily large.

All in all, as suggested by our stated goals and the above explanations, for this thesis, we need to adopt a *developer-centered, mixed-method, interdisciplinary approach, that borrows knowledge and methodologies from other disciplines*. We describe further this approach and these disciplines in the next subsection.

1.3.1. MAIN STAGES OF THE APPROACH: NEEDS FIRST, TECHNOLOGY LAST

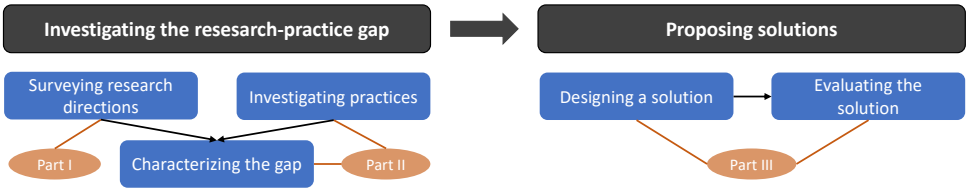


Figure 1.5: Overview of the two-stage approach adopted in this thesis.

OUR TWO-STAGED ITERATIVE APPROACH

We take an approach (cf. Figure 1.5) that relies on the regular scientific method in order to develop solutions for making ML systems less harmful. Instead of assuming a specific problem to solve (based on technical limitations of existing technical research) and working towards proposing a solution and evaluating it, we first apply the scientific method to the investigation of the problem itself, and only later on apply this scientific method to propose solutions to the identified problem. This way, we aspire to tackle problems that are urgent for developers and that would better support them in developing less hazardous ML models. To do so, we depart from an algorithm-focused approach, to go to a human-centered approach, where we do not necessarily identify areas with a lack of technical research, but remain open to identify challenges in terms of usage of the technical research outputs by developers. Naturally, one should later on iterate over any proposed solution, and over the two-stages of the approach once practices or research directions change.

This two-stage approach, where we first investigate the problem and then the solution, follows the double diamond model of the design process, proposed by the British design council[21] following investigation of a myriad of design projects. In the thesis, we first investigate the current problem broadly, then delve deeper into understanding one of the sub-problems identified, and later on develop solutions to this sub-problem.

AN APPROACH ADOPTED IN ADJACENT DISCIPLINES

The most relevant adjacent fields. Besides design research, this iterative, two-staged, approach has already been discussed in other disciplines that also face a misalignment between research and practice, making it particularly adapted to our problem. Especially, in human computer interaction (HCI), Norman [586] argues that the differences in skills required by a researcher or a practitioner and in the interests they have are some of the primary contributors reinforcing the research/practice gap. According to Norman, the widely-spread research trend that follows the precept "Technology First, Needs Last", indeed reinforces the gap. There, one typically performs research either to improve an existing product, or for the “fun” of research and developing fundamental knowledge without a specific problem in mind. Yet, this approach does not allow for the creation of

[21] https://www.designcouncil.org.uk/our-work/news-opinion/double-diamond-universally-accepted-depiction-design-process/

innovative insights that answer a specific problem. Hence, Norman argues that some researchers should also research problems and needs first, and later on solutions to these problems —of course, developing solutions might in turn require fundamental research. This "Technology First, Needs Last" trend is also the primary research trend we identify for ML, and that we do not follow, preferring the approach "Needs first, Technology Last". Norman concludes for the need of "translational developers" that would go from research to support of its use in practice, and from practical needs to ensuring their research. This is one of the roles we take in this thesis. We see the ML developer as the designer of systems, that we, as translational developers, should investigate. Other HCI researchers [190, 307, 847] have advocated for a similar approach, employing the idea of emphasizing the need for efforts in *trickling down* research inputs to practitioners, and *bubbling up* practitioners' needs to researchers.

Besides, a former artificial intelligence researcher, Phil Agre [13] theorized in 1997 the lens of Critical Technical Practice after working on the technical aspects of artificial intelligence, and noticing the limitations of current research trends. "A critical technical practice will, at least for the foreseeable future, require a split identity – one foot planted in the craft work of design and the other foot planted in the reflexive work of critique." He proposed to constantly iterate between two activities: 1) critically analyzing current assumptions, methods, and research directions to generate new questions, and 2) designing for these new questions. As we share similar concerns with the limits of current ML research, our approach also reflects the critical technical practice advocated by Agre. The first activity mentioned is what we do when we search for explanations for the misalignment between research and practice, and more generally look for hidden assumptions from research. The second activity is what we do next when we propose potential solutions, evaluate them thoroughly and discuss them further. Such lens has also been adopted previously to investigate other harms related to ML, such as privacy [322] and explainability [244].

While each of these disciplines has a slightly different objective (e.g., solving a problem by designing appropriate solutions, critically examining a problem and its solutions to redirect the research about the problem or its solutions towards more relevant ones), the approach they take is similar. We also follow a similar approach, as it is adapted to our research goal. Especially, as the research/practice misalignment is still not well understood, taking this approach allows to investigate it broadly, keeping the field of possibilities open, in case one or multiple objectives of these disciplines reveal more relevant than others for our problem.

How about ML research? Prior research in ML has rarely adopted such a reflexive, developer-centric approach, e.g., the evaluation of explainability methods is typically, solely quantitative, lacking user-studies focusing on the use of the methods by the intended developers [871, 413, 931]. Yet, a few works have already been conducted along those lines. These works are typically proposed in conferences such as FAccT, AIES, CHI, or CSCW, and often summarized as the field of HCI+ML or STS and ML (cf. Introduction Chapter 1.2.1). Empirically, some researchers have studied practices and needs revolving around ML, using various qualitative methods, such as semi-structured interviews on the participants' own use-cases [369] or given use-cases [220], or ethnographies

[620]. Others have proposed to shift from algorithm-centered research to data-centric work with data excellence (the underlying idea being that data quality is the main driver of ML harms) [42, 714, 110]. Conceptually, others have critically analyzed the way research is currently organized, i.e., the research directions taken [46], the assumptions made [557, 457, 603, 450], the methods adopted for evaluating solutions [659, 738, 500] or even for writing a research paper with reflexivity [771] —we will especially focus on the assumptions. Finally, other researchers [457, 244] have designed new initial solutions and paradigms to be further evaluated and iterated upon, such as Ehsan et al. [244] who adopted Phil Agre's Critical Technical Practice lens to propose new explainable ML research directions. This thesis complements the efforts of these works.

1.3.2. MIXED METHODS ACROSS STAGES, GROUNDED IN THE PRACTICE

We now delve deeper into the methods we adopt for each stage of our approach. Naturally, surfacing practices, identifying major research directions, understanding the research/practice gap, and proposing and evaluating solutions, do not all call for the same methods. Hence, we adopt a mixed-method approach [202] in this thesis. We conduct structured literature surveys [37] to understand research directions, and qualitative studies to understand practices, and critically reflect on the research/practice gap [501, 586]. Later on, we take a research through design approach [789], with system co-design, system technical implementation, quantitative system evaluation in terms of technical capabilities, and qualitative system evaluation in terms of non-functional requirements and usefulness to the developers. In each stage of the approach, we make sure to ground our work with developers by questioning current practices, surfacing technical literature that might relate to developers or that assumes certain practices, and evaluating the solution not only technically but also with the developers.

CHARACTERIZING THE PROBLEM

A number of computer science research areas have already discussed the gap between research and practice, notably software engineering [501, 398], and human-computer interaction [586, 189].

In software engineering, Ivanov et al. [398] for instance investigated the broad research/practice gap of software engineering by interviewing practitioners and asking them directly what they focus on in their practice and what are their needs, and comparing this to current research directions, identifying a misalignment in research lines but also a lack of knowledge from practitioners about potential useful resources from practice. They argued that researchers, having much less resources than industry, should be more strategic in choosing their research directions, and (partially) follow what practitioners care about, in order to produce impactful work. Lichter et al. [501] observed specifically that software engineering literature was extensively describing benefits of prototyping before building a full industry software system, yet it was unclear whether practice was recognizing the same advantages and more broadly whether practice did understand the activity of prototyping (e.g., categories of prototyping, goals of and methods for prototyping) in a similar way to the literature. To investigate these questions, they interviewed practitioners across various projects about the ways they perform prototyping, and extracted insights by performing comparisons with the literature. Others

have extended their considerations to the gap between the needs of software engineering practice and software engineering education —that remains connected to software engineering research as one of the vectors between research and practice— [594, 280], e.g., through surveys of students, industry requirements, etc. While this is not a topic in which we delve deeper, it could be relevant in the future in terms of ML education and practice, as education is one of the factors we identify as limited towards developing less hazardous models (cf. Part II Chapter 7).

In human computer interaction, conceptually, several researchers [190, 307, 847] have theorized further the gap (a multi-directional gap between theoretical and applied research, and with practice) and proposed solutions. They especially argued for exploring design practices via qualitative methods such as ethnographies or research through design activities [282, 789], and for bridging the gap via co-producing boundary objects via, e.g., via workshops [847]. Colusso et al. [189] for instance further characterized the gap by interviewing HCI practitioners directly questioning their reasons for not using academic research, and identified a number of translational resources (e.g., actionable design guidelines, design patterns, keyword mapping, etc.) needed to bridge the gap.

Our method. Acknowledging these adjacent works, the amount of ML research, and the diversity of ML practices, we divide the first stage into two intertwined activities. We propose to thoroughly understand the state-of-the-art via rigorous surveys of the literature [37], and we identify ML practices via qualitative methods. While performing both of these activities, we extract insights towards our goal by critically reflecting on the misalignments between the identified research trends and the identified challenges and needs of developers, and identifying limitations in existing practices by critically comparing them to existing research outputs.

In terms of qualitative methods, we use grounded theory methods, i.e., "systematic, yet flexible guidelines for collecting and analyzing qualitative data to construct theories 'grounded' in the data themselves." [166] Especially, we conduct semi-structured interviews of a breadth of developers, that we analyze through both inductive and deductive coding approaches. We synthesize our codes and resulting comparisons with the literature into diverse conceptual frameworks. Other methods such as ethnographies could also be used in the future, but were not feasible in the time imparted for the thesis.

CO-DESIGNING AND EVALUATING HUMAN-CENTERED SOLUTIONS

The next stage after characterizing the problem and selecting one specific sub-problem is the iterative development and evaluation of solutions. For this, we adopt a human-centered approach to the solutions. We do not necessarily create entirely novel, complex, technical methods, but evaluate thoroughly existing methods via user-studies, and adapt and expand (e.g., combining various prior research areas such as explainability, data mining, and crowdsourcing) the relevant methods (e.g., explainability) to fit the developers, the problems they encounter, and their needs in context (e.g., cost). For that, we conduct qualitative and quantitative user-studies of existing and proposed solutions to investigate the usability of these solutions. We also evaluate the performance of our solutions in terms of correctness and richness of their outputs via ad-hoc quantitative evaluation procedures and qualitative thematic analysis of the outputs. To develop our

solutions, we adopt a research through design method [282, 789]. Our goal is simultaneously to build a solution useful to the developers, and to use our prototype solution to further understand these developers, their needs, background knowledge, challenges, etc. There, we especially proceed to formative research steps with desk research and developer interviews, and co-design sessions [792] to arrive to high-fidelity prototypes of solutions that we later implement and evaluate. Both the formative studies that lead to design the solutions and the evaluation of the implemented solutions contribute new research insights in the realm of the intersection between ML practices and needs, and ML hazards.

1

1.4. RESEARCH QUESTIONS & ORIGINAL CONTRIBUTIONS

Based on the stages of our approach and the methods envisioned to conduct the research for each of the stages, we organize our research work and the present thesis in three parts (cf. Figure 1.6). Part I serves to identify current research directions and research insights in terms of machine learning (ML) harms and their mitigation via robustness and fairness technical approaches. Part II first serves to understand ML developers' practices with regard to these harms and proposed technical approaches, and then to characterizing the research / practice gap by comparing the insights of Part I to those of this Part. Finally, Part III serves to propose and evaluate initial solutions to some of the manifestations of the research / practice gap. Below, we detail further the research questions and contributions of each part.

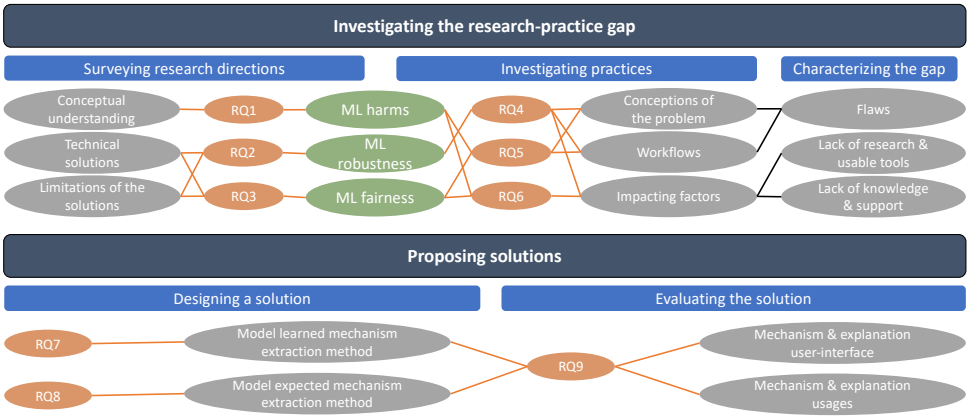


Figure 1.6: Summary of the research questions addressed in this thesis (orange), in relation to the main stages of our approach (black and blue), main objects of focus (green), and main insights (grey).

1.4.1. PART I: UNDERSTANDING STATE-OF-THE-ART

In the first stage of our approach, we investigate the causes for the research / practice gap, which requires us to study both existing scientific literature and practices. In Part I, we proceed to the study of the literature. In particular, we ask the following research questions in each of the corresponding chapter:

- **RQ1**: How does the literature conceptualize harms that machine learning might cause, and their technical sources within the machine learning lifecycle? (Chapter 2)
- **RQ2**: How does the literature on machine learning robustness propose to tackle hazardous output failures? What are the known limitations of these propositions? What does this literature tell us about the (mis)alignment between research and practice in terms of using robustness-related solutions? (Chapter 3)
- **RQ3**: How does the literature on machine learning fairness propose to tackle hazardous output failures? What are the known limitations of these propositions? What

does this literature teach us about the (mis)alignment between research and practice in terms of using fairness-related solutions? (Chapter 4)

The original contributions of Part I are the following:

- An overview of the current state of technical knowledge on two main output hazards of machine learning, namely fairness (discrimination) and robustness (brittleness), in terms of proposed formalisations of the problem, causes of the problem, and mitigation methods.
- An overview of the conceptual and practical limitations of the above technical research envisioned by the interdisciplinary community, and the outline of future research directions in terms of future technical and design contributions.
- An overview of the (limited) knowledge about practices of machine learning developers in relation to fairness and robustness issues.

Part I is based on extracts[22] of the following publications:

- A full paper published at the Transactions on Social Computing 2021 [70] (***Agathe Balayn***, *Jie Yang, Zoltan Szlavik, Alessandro Bozzon*).
- A full paper currently under evaluation at the ACM Computing Surveys 2023 [825] (*Andrea Tocchetti, Lorenzo Corti,* ***Agathe Balayn***[23]*, Mireia Yurrita, Philip Lippmann, Marco Brambilla, Jie Yang*).
- A full paper published at the VLDB Journal 2021 [66] (***Agathe Balayn***, *Christoph Lofi, Geert-Jan Houben*).
- A technical report written for the European Digital Rights organisation (EDRi) in 2021 [62] (***Agathe Balayn***, *Seda Gürses*).

1.4.2. PART II: SURFACING PRACTICES & ANALYSING THE RESEARCH / PRACTICE GAP

In Part II, we proceed to the study of practices of ML developers and to contrasting these practices with the state-of-the-art research identified in Part I, in order to characterize the nature of the research / practice gap and its causes. We ask the following research questions:

- **RQ4**: How do machine learning developers debug their models for robustness issues in development? How does the research/practice gap manifest in this step of the machine learning lifecycle? What are the main challenges and limitations in these practices? (Chapter 5)

[22] We did not modify the text of the publications, except when reconciling terminology across chapters of the thesis. We also did not modify the order of the sections in the publications. We removed sections or paragraphs from the original publications that were bringing an unnecessary deep level of details, without further contributing to the storyline of our thesis.

[23] The first three authors are co-first authors.

- **RQ5**: How do machine learning developers envision and tackle unfairness issues and other harms that might arise from the models they develop? How does the research/practice gap manifest in this step of the machine learning lifecycle? What are the main limitations of their practices? (Chapter 6)
- **RQ6**: What are the main underlying factors that impact the attitudes and practices of machine learning developers, and that might represent challenges leading to the persistence of harms? (Chapter 7)

The original contributions of this part are the following:

- A detailed account of current attitudes and practices for developing a machine learning model, both in terms of goals the developers have, workflows they execute towards these goals, and methods they exploit to do so, in terms of robustness broadly, and in terms of fairness and other social harms.
- An account of the (mis)alignment between research and practice, with specifically a list of the limitations of current practices identified by a comparison with existing literature on the topic, and a list of limitations of current technical work in supporting ML practices.
- A reflection on the nature and reasons for the research/practice gap that lead to these limitations, with the identification of various human and contextual factors impacting the gap, and a discussion of future research directions necessary to tackle the gap.

Part II is based on the following publications:

- A full paper published at the CHI 2023 conference [68] (***Agathe Balayn***, *Natasa Rikalo, Jie Yang, Alessandro Bozzon*).
- A full paper currently under evaluation at the CHI 2024 conference [61] (***Agathe Balayn***, *Ujwal Gadiraju, Jie Yang*).
- A full paper published at the AIES 2023 conference [71] (***Agathe Balayn***, *Mireia Yurrita, Jie Yang, Ujwal Gadiraju*).

1.4.3. PART III: BRIDGING THE GAP: A MODEL MECHANISM DIAGNOSIS TOOL

After our broad inspection of the research/practice gap, we delve deeper into one issue identified. We tackle the challenge developers face in developing non-hazardous models and evaluating the harmfulness of their model during development time without having access to production data. We propose for them to shift from using accuracy metrics on test datasets to estimate the harmfulness of their model —limited because the test accuracy might not reflect production accuracy due to distribution shifts between test and production data—, to exploiting information about the mechanisms the model uses to make predictions on test data. We define a model mechanism as a rule a model uses to associate a data sample to its predicted class label, this rule taking the shape of features of the data sample the model identifies and correlates to the predicted label. Mechanism accuracy is more informative than test (prediction) accuracy in terms of potential harms a model might cause in production as a model might make correct test predictions using

a wrong mechanisms, that would lead to wrong predictions in production, and test accuracy does not allow to identify these signs of wrong production predictions. In order to develop mechanism-based harm estimation, we ask and answer the following research questions:

- **RQ7:** How can one collect easily-interpretable mechanisms learned by a model for making predictions? (Chapter 8)
- **RQ8:** How can one collect the mechanisms the model is expected to learn according to human reasoning? (Chapter 9)
- **RQ9:** How can a developer use our mechanism information to diagnose a model's output failures? How useful is this information in comparison to the one provided by existing explanability methods? (Chapter 10)

The original contributions of this part are the following:

- A method, its implementation, and its evaluation for collecting the mechanisms learned by a model.
- A game with a purpose for collecting tacit knowledge of crowd workers efficiently, that could be translated into expected mechanisms for a model.
- A co-created user-interface displaying our mechanism-based information and other types of explanations extracted from a model, that can be used by practitioners to diagnose their models for output harms.
- A user-study that investigates the utility and usability of the user-interface, and research insights for the design of new supportive tools for practitioners.

Part III is based on the following publications:

- A full paper published at the Web Conference 2021 [69] (***Agathe Balayn***, *Panagiotis Soilis, Christoph Lofi, Jie Yang, Alessandro Bozzon*).
- A full paper published at the Web Conference 2022 [64] and a demo paper at the HCOMP 2021 conference [63] (***Agathe Balayn***, *Gaole He, Andrea Hu, Jie Yang, Ujwal Gadiraju*).
- A full paper published at the CHI 2022 conference [67] (***Agathe Balayn***, *Natasa Rikalo, Christoph Lofi, Jie Yang, Alessandro Bozzon*).

I

State-of-The-Art on Hazardous Failure Diagnosis

In the first stage of our approach, we aimed at understanding the causes for the research / practice gap in machine learning (ML), which requires us to investigate both existing scientific literature and practices, and contrast them. In Part I, we proceed to the study of the literature, via rigorous literature surveys. Later on, in Part II, we will tackle the study of practices and contrast those with the literature, and in Part III, we will start proposing solutions to certain causes for the gap. As our goal is to understand the research about machine learning harms comprehensively in order to analyse practices later on, we have to survey two sets of research publications. We survey the literature that describes harms and surfaces their causes —this is typically socio-technical research as harms can only be understood by looking at the technology, its functioning and outputs, and its impact in the real-world—. We also survey the literature that proposes solutions to identify and mitigate these harms —primarily technical research—, and that discusses potential limitations of these solutions.

In Chapter 2, we specifically ask:

> ***RQ1:*** *How does the literature conceptualize harms that machine learning might cause, and their technical sources within the machine learning lifecycle?*

To answer this question, we first take the concrete example of one technology, automatic offensive language detection systems, in order to identify a number of harms that have been discussed for this technology, and the sources of these harms. As other research papers discuss additional harms without focusing on specific machine learning applications, we then expand the scope of machine learning harms by surveying additional, interdisciplinary, publications. This chapter is based on an extract from a publication accepted in the TSC journal [70], and an extract of one section of a report written for the European Digital Rights organisation (EDRi) [62]. Specifically, we only keep from these publications the most relevant information about harms, without going into too extensive depth in the descriptions of these harms, and without providing an unnecessary large number of concrete examples, as this is not necessary to answer our research questions.

From the literature about offensive language detection system, we find that most harms consist in the negative social impact, e.g., in terms of unfairness and discrimination, that errors in the outputs of a model can have. These harms are typically caused by problematic dataset and model design, and find deeper sources into the limited, ambiguous, requirements outlined for the systems. From the rest of the literature, we find the same harms and their sources, as well as additional harms that are not directly related to the outputs of the systems: harms directly caused by the design of the machine learning task and dataset (e.g., offensive prediction labels), as well as harms caused by the production of the systems and especially their infrastructure (e.g., environmental impact).

When looking for research publications that propose solutions to harms stemming from errors in the outputs of the machine learning models, we identify two lines of technical solutions: solutions to ensure the robustness of machine learning models, and others to ensure the fairness of the models' outputs. We do not explore solutions proposed towards the other types of harms, as they are very rarely published into the literature,

and typically do not consist in technical contributions (the solutions typically consist in rethinking the entire system design or not deploying the system).

In Chapter 3, we investigate the first line of research. We specifically ask:

> ***RQ2:*** *How does the literature on machine learning robustness propose to tackle harms? What are the known limitations of these propositions? What does this literature tell us about the (mis)alignment between research and practice in terms of using robustness-related solutions?*

This chapter is based on an extract from a publication under revision for the ACM Computing Surveys [825], that rigorously surveys the literature on machine learning robustness. Considering the length of the contribution in the survey and the broader scope, we decided to only retain content directly related to the rest of the content in the thesis, i.e., sections that provide a broad overview of the field with its main technical research directions, and that discuss perspectives for future work, with a human-centered lens. Despite the potential usefulness of these technical research directions, the developers interviewed in Part II never referred to any of them and only rarely referred to robustness failures in any case, rendering their detailed, technical understanding unnecessary for answering our research questions.

In Chapter 4, we investigate the second line of research. We specifically ask:

> ***RQ3:*** *How does the literature on machine learning fairness propose to tackle harms? What are the known limitations of these propositions? What does this literature teach us about the (mis)alignment between research and practice in terms of using fairness-related solutions?*

Here, we survey the machine learning literature that deal with this topic, but also the related data management literature, as Chapter 2 showed that many harms are caused by the way training and test datasets are created. Besides, we also account for the growing set of interdisciplinary, critical, science and technology studies papers, that identify conceptual but also practical limitations of the proposed machine learning fairness solutions. This chapter is based on an extract from a publication accepted in the VLDB Journal [66], and an extract of one section of a report written for the European Digital Rights organisation (EDRi) [62]. Specifically, we kept from these publications only the minimum level of detail that allows to answer our research questions.

From these two lines of research, we identify a large set of metrics and mitigation methods that are proposed to tackle unfairness and brittleness issues of the machine learning models. These two types of issues are sometimes tackled together due to their interdependence, as for instance distribution shifts that cause brittleness might also increase the unfairness of a model that seemed fair on its training dataset, or because a method that increases a model's robustness might also decrease its fairness. We also identify a number of conceptual and practical limitations to these metrics and methods. For instance, the practical applicability to real-life scenarios and usability are questioned, e.g., the proposed methods assume access to datasets or model parameters that

machine learning developers cannot reach in practice, and the conceptual frame of the methods does not account for the complexity of the harms they aim at targeting, e.g., although unfairness mitigation methods allow to reach a certain level of parity in treatment of different individuals in a population (e.g., equal rate of output errors across groups of population), they do not encompass the different ways in which a same model output might impact differently each individual. In terms of the (mis)alignement between research and practice however, we primarily identify a lack of research on the perception and use of the technical solutions by machine learning developers. We especially do not know to what extent they are considered by these issues and the harms they might cause, whether they use the technical solutions, and if so, how they are able to tackle their limitations. These are research gaps that we investigate in Part II, in order to identify the most pressing needs to be tackled in Part III with further research work.

To summarize, this part contributes three rigorous surveys of the literature, one about harms, their causes, and their treatment within automatic offensive language detection systems [70] and other types of machine learning based systems [62]; one about technical approaches for mitigating robustness failures of machine learning models and limitations of current research [825]; and one about technical approaches for mitigating unfairness issues [66], and the limitations of these proposed approaches [62]. From these contributions, our main realization was that despite the complexity of the problem of hazardous machine learning failures, the amount of technical research tackling both robustness and fairness failures towards hazardless machine learning is not yet equalized by research on the limitations of these approaches and on the practices of developers who might try building hazardless systems –potentially with these approaches. We especially mitigate the the lack of research on practices in Part II of this thesis.

2

On Algorithmic Hazards, Harms & their Sources

2.1. Introduction

In this chapter, we begin our investigation of harms and their sources. We start by focusing on one specific use-case, the automatic detection of conflictual languages, because it tackles an interesting intersection between machine learning and its application to one field, allowing for more precisely envisioning harms[1]. We then investigate broader literature that identifies additional harms, that might not occur in the above use-case. Harmful, aggressive, abusive and offensive languages in online communications are a growing concern [251, 291, 873]. They constitute a threat to Freedom of Speech [829], damage the dignity of the targeted individuals [857], and prevent healthy and fruitful conversations [538]. The recent hearings [470] of the biggest social network's platform (Facebook) CEO also testify of the growing public attention on the issue.

Manual moderation is still the most reliable method for content filtering [491, 632, 469, 443], but it suffers from several issues. Content moderators cannot handle the deluge of user generated content fast enough not to endanger anyone. Moreover, they are continuously exposed to hurtful content, which induces mental issues and can lead to self-harm acts [775]. Under the societal and political pressure [387, 263], online platforms are urged to find computational solutions to detect conflictual languages [272]. Machine learning approaches are considered the best solutions [263], due to their promise to achieve reasonable detection performance at scale. In practice, error rates still demand for extensive manual moderation. For instance, Arango et al. [35] show the frequent drop of performance for machine learning models evaluated on deployment data (e.g., a model which achieves 70 F1-score on its test dataset, can only achieve 21.1 F1-score on another dataset).

[1] We left out from the original publication [70] the subsections that provide extensive descriptions of the survey methodology, of the differences between this survey and prior work, of the disentanglement of definitions around conflictual languages that we had operated, and of additional research challenges not directly tied to algorithmic harms.

Classification errors can raise various harms, and especially concerns of discrimination [807]. For example, models might systematically misclassify certain populations more often than others, for instance more often associating tweets written in African-American English to negative classes than tweets written in Standard American English [567], or misrepresent their identities due to stereotypical associations between certain concepts and sensitive attributes [119]. The causes of these errors can be summarized under the broad term of *bias*. When the training dataset is biased towards certain (latent) characteristics, the model is implicitly taught a biased representation of the conflictual languages. While these biases are *technical* artifacts, we argue that their root causes and solutions cannot only be found in the technical realm. Issues at the *conceptual* level induce these biases and the challenges in tackling them. Through this survey, we show the existence of several *mismatches* between the typical formalisation of conflictual languages in the computer science literature, and how people perceive and experience such languages in reality. Mismatches first manifest at a *terminological* level, as publications often use an incorrect term to refer to the conflictual language they study; but they further deepen into *semantic and contextual* levels. For instance, psychology literature highlights that the perception of conflictual languages depends on various contextual factors [178], such as one's prior experiences (e.g., someone who is frequently subject to racial prejudice might perceive sentences as hate speech more strongly), or the direct context of a sentence, e.g., its author and target. Failing to acknowledge such rich characterisation has obvious implications for the correctness and effectiveness of the deployed system. Consider, for instance, the widely-used practice of keyword-based sampling in training data construction, i.e., collecting conflictual text based on certain keywords. This method implicitly teaches a model that conflictual languages contain specific words, and leaves out offensive texts with more subtle – or "coded" language that, in practice, makes the resulting system ineffective.

In this survey, we aim at surfacing and systematically characterising these mismatches and the technical biases that reinforce them, to highlight relevant research challenges. Figure 2.1 summarises the research fields and technical aspects addressed in our survey. By interrogating psychology literature, we drive an informed analysis of trends in computer science papers, and propose a consolidate taxonomy for conflictual languages. Then, we identify the biases that arise from prior conceptual mismatches. By adopting a data-centered view, we show that many issues in the outputs of the systems originate from problematic choices in the design of data engineering pipelines.

The computer science literature on the automatic detection of online conflictual langugaes focuses on a few languages: hate speech [263, 727], cyberbullying [702, 18, 392, 17], flaming [528], offensive [713], and aggressive language [468]. We compare all these languages, and focus on their most common manifestation, text. We believe that research on one language might benefit research for another language, and that a precise and organised terminology is needed to improve the quality and applicability of automatic solutions [849]. We employ the term Online Conflictual Language (*OCL*) to refer to the overarching category of online language that subsumes all these types. We use the term "language" instead of "speech" (used in computer science to refer to hate speech [263, 221]) because the latter implies the spoken nature of the sentence [723]. In contrast to terms with specific meanings (e.g., "aggression" implies the intention to harm),

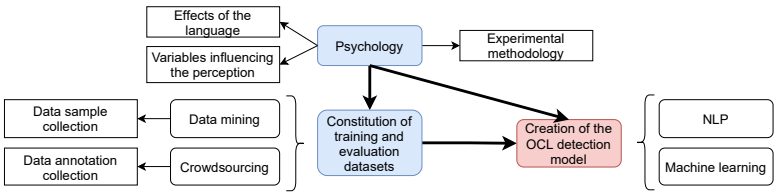


Figure 2.1: Dependencies which influence the design of online conflictual language detection systems. Technical works in NLP and machine learning, and possibly works from psychology and politics determine the inference task. Datasets are then developed (or selected) according to the task, in a way that is also informed by data mining and crowdsourcing literature.

we use the term "conflict", defined as *"the occurrence of mutually antagonistic or opposing forces, including events, behaviors, desires, attitudes, and emotions"*. We use "Online Conflictual Language" also to avoid ambiguity and confusion, as the term has not been previously used in psychology, linguistics, or computer science.

In summary, we make the following contributions:

1. A discussion of the psychological aspects related to *OCL*s (section 2.3) that uncovers conceptual mismatches with automatic detection works, and a reflection on the experimental practices that could contribute to computer science research.

2. A comprehensive review of the typical data engineering pipelines used for building datasets (section 2.4), and of their technical biases (e.g., usage of disagreement metrics for evaluating the annotation quality of subjective *OCL*) that can be harmful and participate to the low generalization abilities of the systems.

3. A quantitative review of conflictual language detection models (section 2.5), and an analysis of their limitations in terms of performance, leading to the identification of additional biases. Guided by our *OCL* taxonomy, our work offers a principled characterisation of differences, similarities, limitations and opportunities in computer science approaches. The lack of features relevant to individual *OCL* and the integration of social biases are pressing issues, for which future research could draw inspiration from literature in psychology, machine learning fairness, and explainability.

4. An extensive discussion of open, technical and structural, research challenges, with actionable suggestions for future work inspired by various psychology and computer science domains, and informed by our systematic literature analysis (section 2.6).

5. In a last section, we will provide an overview of other harms caused or reinforced by machine learning systems, but that have not been necessarily related to *OCL* detection systems (section 2.7).

2.2. METHODOLOGY

To achieve the aforementioned contributions, we take a multi-step approach including, (1) retrieving relevant terms about *OCL*, (2) literature search and analysis, (3) taxonomy of terms creation, and (4) analysis of the research challenges. More details can be found in the original publication [70].

2.3. CONCEPTUAL MISMATCHES TOWARDS TECHNICAL BIASES

In this section, we argue for the existence of profound *conceptual mismatches* pertaining to the focus of computer science literature on the development of algorithmic pipelines, mostly due to lack of consideration for the application context –*contextual mismatch*–, or for the specific properties of the targeted *OCL* –*semantic mismatch*. We provide an overview of the insights about *OCL* that can be found outside computer science research, and compare them to high-level findings from our systematic survey of computer science literature.

2.3.1. EXTERNAL INSIGHTS ON ONLINE CONFLICTUAL LANGUAGES

SEMANTIC KNOWLEDGE FROM PSYCHOLOGY

Researchers in psychology have extensively studied conflictual languages, beyond the context of Web communication platforms. We summarize here the major insights relevant for the prospect of detecting these *OCL*.

Three main types of variables influence how *OCL* is perceived by external *observers* (see Table 2.1): the *language content*, including the properties of a *person or group targeted* by *OCL*; the *language context*; and characteristics of the *observer*.

Table 2.1: The factors identified in psychology literature that influence *OCL* perception, organised in 3 categories (*internal* characteristics of the observer, characteristics of the *sentence content* and of the *sentence context*), and the approach taken to measure these variables.

Category	Variable	Measure	Paper
Observer	Gender	Question	[318, 235, 198, 199]
Observer	Ethnicity	Question	[198, 199, 877]
Observer	Education	Question	[198, 199]
Observer	Age	Question	[198, 199]
Observer	Liberalism inclination	Question (scale)	[235]
Observer	"Individuals' attributions of intent", angry and anxious dispositions	Not investigated	[318]
Observer	Sense of mastery, self-esteem	Question	[635]
Observer	Frequency to which people are subject to racial prejudice, "beliefs about the appropriateness of expressing racial prejudice"	Question (scale)	[591, 877]
Observer	Membership esteem to the offended group	Question (scales)	[114]
Context/Content	Targeted group or person	Scenario	[198, 199, 114, 353]
Content	Category of hate speech	Info in dataset	[353]
Content	Prejudice, sentence properties	In the dataset	[223, 198]
Context	Public or private sentence	Scenario	[198, 199]
Context	Received response to the language	Scenario	[198, 199, 200]
Context	Author, its characteristics, race, gender	Scenario	[203, 639]
Context	Hierarchical level of perpetrator and victim	Question	[821]
Context	Internet community	Info in dataset	[779]
Context	Social status of a group	Question	[353]

Internal characteristics of the observer. The perception of certain *OCL* depends on the internal characteristics of someone who observes the language. This hints at the subjective nature of many online conflictual languages. For instance, Guberman et al. [318] observe a difference in *aggressiveness* ratings of tweets depending on *gender* (women rate tweets more often as aggressive than men) and mention the tendency that some people have "to interpret ambiguous stimuli as being intentionally aggressive" and the dispositions of people to become angry and anxious. Downs et al. [235] identify that *gender and liberalism inclination* influence how harmful a hate speech is perceived. Similarly, Cowan et al. [198, 199] point out that the *ethnicity, gender, education, and age* of the observer influence the perceived offensiveness of hate speech. Besides, attention is called on the distinction between the perceived *offensiveness* and *harmfulness* [200], with for example ethnicity being a main factor in the perceived harmfulness. This highlights the importance of clearly and precisely defining the *OCL* to detect, in order to account for the correct variables of importance.

Works focused on racial hate speech also pinpoint *the frequency to which people are subject to racial prejudice* and *people's "beliefs about the appropriateness of expressing racial prejudice"* [591], and *ethnicity* [877] (e.g. people of color who are more often subject of racial aggression perceive Web memes as more offensive, unlike White people). This speech triggers various emotional responses (fear, anger, sadness, outrage), and people with high membership esteem react more strongly to threats to their group than low identifiers [114].

Sentence content and context. The syntactic and semantic *properties of the sentence*, e.g. length, usage of profanity, and its *context* –author [203] and how its direct target behaved and felt [200], targeted group, whether it is public or private, and whether it received a response [198, 199]– influence how offensive it is perceived [203, 353, 198]. For instance, the perception of profanity depends on the *community* [779] as different communities use profanity with different frequencies and contexts and judge the words differently. Besides, a speech toward a single individual is seen as more offensive than a speech toward a group of people [114]. Also, a speech is offensive when it presents a property of an individual ("personal characteristic, belief", etc.) in a certain way which does not need to be hateful [36], as the wrongfulness comes solely from the aim of its author: "attempt to denigrate, humiliate, diminish, dishonour, or disrespect the other". The context is particularly relevant when distinguishing between languages that are *harmful* – which damages someone's interests – from languages that are *hurtful (offensive)* – which causes mental distress.

These three types of variables implicitly include finer-grained characteristics of the language: *the focus towards certain types of population and specific targets, the type of language used, the author, its intent and the effect on the targets.*

CONTEXTUAL INFORMATION AROUND *OCL* DETECTION SYSTEMS

Context of application of the systems. The application domain of an *OCL* detection system determines its *context* of operation (e.g., a social media primarily used by children within a single country using a single language, or used by a specific political community to discuss political opinions on specific subjects). Context consists in the type

of platform (e.g., social media, conversational agent) on which *OCL* should be detected, the type of end-users and their backgrounds, the type of communities and populations that are present on the platform or interact with this agent, the topics that are frequently tackled, and the natural language typically employed (which can be different from offline language). These characteristics might impact how someone perceives *OCL* [849]. Understanding this impact would allow to scope the context in which systems can be used, and would determine how to collect datasets for training, and how to develop and test algorithms.

Laws and regulations, either governmental or from social media platforms, further constrain the type of online conflictual languages to be detected. They focus on certain properties of language, such as intent or targets (identified in the previous section) that are often more specific or nuanced [112]. For instance, the British government decided after many debates on "protections only against intentionally threatening expressions of religious hatred, not against those that were merely abusive or insulting, nor those that are reckless and likely to stir up hatred". Philosophy also studies when *OCL* should be limited and similarly defines criteria to make a decision, by analysing case-by-case past events of *OCL* on social media [319]. Especially, it should be limited when "it is reasonable and feasible to assume that an act of Internet speech will cause harm to others", and more specifically when "targeted hate speech that carries with it immediate harm (capability to carry out the violence), individualized harm (capability to assault the target), and capability to carry out the threat (actualized means of committing the violence)". As our investigation in the remaining of the paper shows, such nuances are not necessarily reflected in the ways datasets and models are developed, yet would be of importance, for instance, not to unintentionally restrict freedom of expression.

Hard technical requirements for the applications. The applications in which *OCL* detection systems are implemented also impose hard technical requirements (e.g., *OCL* posts should be removed from a platform within a certain amount of time). While these requirements do not necessarily impact the nature of the *OCL* to detect, they might impose constraints on the detection pipelines (e.g., cost of data collection, speed of machine learning inferences with or without the possibility to involve humans-in-the-loop), and trade-offs with the system accuracy (e.g., scalability vs. accuracy). However, these requirements are not often accounted for in the literature, which instead focuses on accuracy. Only 4% of surveyed publications mention the *scalability* of their system, mainly the time efficiency to detect online conflictual language (*OCL*), and only 6% tackle the creation of a full system in opposition to a detection method. These numbers are small considering the need for efficient solutions, since leaving *OCL* public for too long might have psychological consequences for the readers.

The systems are ought to perform well *continuously over time*. Yet, only few systems continuously collect datasets, whereas this would shed light on the evolution of *OCL* along time, the changes in the users of platforms, and how they impact a model's performance, etc. Efforts to develop systems such as MANDOLA [618] or the Online Hate Index[2] would greatly contribute to progresses in the field.

[2] https://www.adl.org/resources/reports/the-online-hate-index

2.3.2. COMPUTER SCIENCE STUDIES ON *OCL*

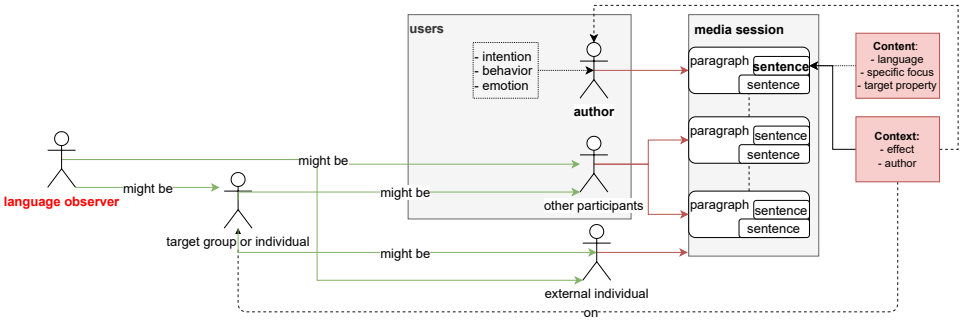


Figure 2.2: Summary of the entities of importance in the understanding of *OCL*, as identified by computer science studies.

Researchers in computer science have conducted studies on the use and spread of *OCL*s on the Web. They perform both manual analysis and statistical observations on datasets collected for the studies, and discover properties of the languages which could be used to tune the features employed by automatic detection methods. These studies serve as a source to identify the entities studied in computer science literature, their relations, and their properties (summarized in Figure 2.2). These entities are primarily the *author* of a language –its behavior, intentions and emotions–, the *language content* itself, be it a single sentence or an entire paragraph –the language used, the targeted property of a person or group, and implicitly the focus of the language since only sentences containing expressions of hate are studied– and its *context* –how it affects the *target person or group*.

Hateful behaviors are characterized with perpetrators *internal characteristics* –their account creation dates, e.g., *hateful* users might be often banned; the amount of the users' activity on the media; the position of the users in the network graph; whether the users are identified as spammers–, and the *characteristics of the sentences they write* – the lexical content and sentiment of their posts and hashtags [676, 168, 148]. ElSherief et al. [247] also identify various personality traits of both authors and targets of hate speech.

Other studies target *media sessions*, i.e., a conversation between several individuals. This is the case for cyberaggression, where both text, images and possibly users are studied –e.g., the role of the author in the cyberbullying–, sometimes with a temporal dimension [375, 92].

Certain studies [559, 197, 753, 215, 865, 444, 323, 933, 822, 779, 357] characterize the *language* itself, through the *sentence content* (i.e., the used vocabulary); the *targets*; the *context* (how the language is perceived); the relation between the type of target and the type of content employed [246]; and the effect of users' anonymity and users' geography. These properties are compared across platforms [459]. One study focuses on why and with which intensity a language is perceived as conflictual by an observer, using questionnaires: a sentence is seen as cyberbullying when it contains threats of physical violence, harassment and profanity terms [224].

2.3.3. COMPUTER SCIENCE FRAMING OF *OCL*

In the remaining of this section, we identify conceptual mismatches that translate into technical biases in the design of automatic *OCL* detection systems. To do so, we compare the formulation of detection tasks in computer science publications to the above insights. We also provide an outline of the works on biases, and contrast them with our previous insights.

FRAMING OF AUTOMATIC DETECTION TASKS

Here, we present how classification tasks are generally framed, and show the diversity of the classes and entities used across tasks.

Table 2.2: Type of entity per online conflictual language (*OCL*), accounted for in computer science classification tasks.

	Aggression	Offensive	Abusive	Harmful language
Media sessions	6	0	0	0
Sentence	83	75	12	1
User	13	1	1	0
Words	3	0	1	0

Entities We find a strong imbalance across entities targeted by classification tasks (Table 2.2). Sentences are the most studied. A few works also detect single words corresponding to a specific online conflictual language (*OCL*), or identify users, public accounts and media sessions which comport *OCL*, based on the detection of sentences and words. Retrieving data for media sessions or users is technically more challenging than for words or sentences. Media sessions are only studied for *aggression* because they allow to analyze the users' behaviors that emphasize user intention, a characteristic specific to aggression. Studying sentences allows to access certain properties of *OCL* (e.g., language type, focus and possibly intention.), but leaves out information relevant for certain types of languages, such as the effect on the reader for offensive languages, or possibly the intention of the author.

Classes. The number of classes targeted in the classification tasks is imbalanced. Most tasks use 2 classes (77.7%) (e.g., is hate, is not hate language) or 3 classes (15.6%) (e.g., is positive, is neutral, is hate language), which corresponds to the basic requirement of the systems. The tasks with more classes (4 to 13) reflect the intensity of an *OCL* language, which is more challenging to detect. As we discuss in the next subsection, binary classes do not necessarily reflect the understanding of *OCL* obtained from our previous analysis. For instance, psychology pointed out to the dependency of certain *OCL* perception on various contextual factors, left out when binary classes are predicted for bare sentences.

MAIN BIAS CONCERNS

We report here the types of biases studied explicitly in relation to automatic *OCL* detection. These mainly relate to certain inherent contextual properties of *OCL* identified by psychology literature, and to a few properties specific to the online context –in certain

cases using the term "bias" directly–, but also to the potential discriminatory impact of *OCL* detection systems. We also investigate how these bias concerns compare to the semantic and contextual information identified in the previous subsection.

Inherent contextual biases. Works on cyberbullying detection have shown how different *authors* of *OCL* –difference based on gender [206], age, profanity history [208] or intent [10]– shape differently their sentences. A few properties of the target or *observer* of the language have also been indirectly studied, mostly through the properties (especially the gender) of the employed dataset annotators (e.g., workers from crowdsourcing platforms) [721]. Yet, the actual observers (e.g., social media users) do not necessarily resemble the annotators of a crowdsourcing platform, and hence studies might not fit the perceptions of actual users. The *conversation context*, specifically *replies* to *OCL*, has also been investigated in a few works [495, 626].

Biases related to the online context of the systems. The contextual characteristics identified in the previous subsections are often not mentioned in papers developing detection methods, except for the *platforms* from which datasets are collected. The similarities and differences in the natural language written across platforms is sometimes investigated by measuring the generalizability performance of models trained on one platform and one dataset across platforms and across datasets [12, 316], as a proxy for the intensity of the differences. Besides, no work was found to study the diverse perceptions of *OCL* of users across platforms.

Similarly, only few works discuss the end-user related information that should drive the development of a system. Arango et al. [35] show that many datasets suffer from *user biases*. Few users constitute the authors of the majority of *OCL* in common datasets, thus identifying *OCL* could translate into identifying the author of a text sample. leading to overestimating models' performance. Besides, only the user social network [416] is investigated as user contextual cue, while it is shown to increase detection accuracy of models relying on it.

Discrimination-related biases. Recent papers employ the term "bias" to study system artifacts that might create discriminatory harms. Such harms are identified by comparing the performance of a system for different subpopulations of users, e.g., based on gender [613], or other sensitive information [56], e.g., sexual orientation [181]; and possibly on intersectional attributes of the users, e.g., gender and political orientation [439]; or racial biases based on dialects [721, 213]. These biases rely on properties of the end-users, and their translation into natural language in the applications (e.g., the background of the end-users imply a dialect). These harms are explained by imbalances of various nature in training datasets (e.g., more sentences written by male authors than by authors of other genders). Sun et al. [807] provide a review of the formalization of these biases in natural language processing tasks, not specifically related to *OCL* detection.

Computer science works that account for biases do not yet encompass all kinds of relevant contextual and semantic information. We take a systematic approach in the remaining of this paper to identify the technical biases that occur from the non-consideration of this information. That is what we discuss in greater extent in the next subsection.

2.3.4. TOWARDS THE TECHNICAL MISMATCHES

We identified the main properties of online conflictual languages as defined by social sciences and the applications' context, and the ones integrated into computer science works. We now synthesise these properties to surface mismatches in computer science research. These mismatches relate to the inherent properties of *OCL* and to the subjectivity of certain *OCL*, left out from both datasets and machine learning models.

MISMATCHES AND CHALLENGES IN THE EXPLOITATION OF THE CHARACTERISTICS OF *OCL*

Mismatches in the selection of variables. The three types of variables that influence the perceptions of online conflictual languages (*OCL*s) identified from social science (subsection 2.3.1), i.e., the internal characteristics of the observer, the sentence context and its content, are similar to the ones found in computer science studies (subsection 2.3.2). However, the exact characteristics investigated vary. Computer science studies focus on properties directly measurable or that can be inferred from information available on the online platforms, while psychology works rely on additional individual questionnaires.

Besides, only few detection methods use these specific characteristics of the languages. For instance, it is recommended to use a sentence context in a media session, and possibly the interactions of the sentence author with other users. It was also shown that the aggregation of hate messages from multiple sources creates stronger harms than a single message from one unique source [486]. However, only individual sentences are usually collected, without any meta-data on context. Psychology also points out to specific language uses, such as euphemism in harassment [286], or humour for hate speech [884], e.g., humour affects the perception of offensiveness for certain types of hate speech (here racism or sexism). However, these are often cited as future work in computer science, except for Magu et Luo [525] who study euphemisms within hate speech, or the recent works on sarcasm in ACL workshops.

Mismatch in the choice of target entity to detect. Psychology and computer science studies highlight the importance of looking beyond sentences, and at single user's behaviors or at entire scenarios, and of distinguishing between certain specific *OCL*. However, current setups do not focus on these factors (subsection 2.3.3), which could lead computer science researchers to target research objects that are ill-defined. Hence, we recommend to refer to the social science literature around the targeted *OCL* to identify the important elements to include in datasets or algorithms for automatic classification of each *OCL*.

Challenges in data collection. The above gaps constitute socio-technical challenges: the social science insights need to be translated into accurate quantities measurable in practice in the technical systems. For instance, considering context in computer science is challenging due to the difficulty in scoping and collecting it, e.g., links in posts are often outdated, finding characteristics of the authors or receivers might be intractable and privacy infringing. This could –ideally– be solved when building training datasets by interrogating users on their perceptions and intentions, but it would be impossible in deployment where users could not be solicited for each post. This shows again the

necessity to identify requirements of applications precisely, as they shape the constraints for training and deployment.

The relevant variables that impact the perceptions of *OCL* need to be identified more exhaustively as psychology studies do not necessarily tackle *OCL*s on the Web, but also in real life scenarios. Also, certain online conflictual languages (*OCL*s) are rarely addressed in psychology research, certainly due to their exclusive online nature (e.g., flaming).

The validity and importance of certain properties about the context of the language used only in computer science (e.g., user account creation date, amount of her activity on the media, her position in the network graph) could be further explored by adopting the methodology followed in psychology. Certain properties might be proxies for some of the psychology variables, e.g., they could help to identify the intent of the author of a post. This leaves the opportunity for computer scientists to work with psychologists to bridge the gap between these domains, and to more precisely define the concepts they study.

SPREAD OF THE MISMATCHES INTO THE CLASSIFICATION PIPELINES

The development of *OCL* detection systems follows the general development of machine learning applications [24, 812]. First, requirements are defined and specified into characteristics for the data, machine learning model and its evaluation. Then, data are collected, cleaned and labeled by annotators. Features are extracted, a machine learning algorithm is developed and trained. The resulting model is evaluated and later deployed and monitored. Certain steps might be iterated over to approach closer the initial requirements, and possibly to revise these requirements.

Shortcomings in the systems arise from these steps. Under-defined requirements (mentioned in previous subsections) propagate into the next data-oriented and algorithm-oriented steps of the pipelines. Tuning pipeline components even for well-defined requirements is challenging. For instance, a system might be asked to perform equally well for children and adult users. However, with the subjectivity of certain *OCL*, building datasets with single, binary labels for each data record, and models that predict single labels, does not fit this requirement.

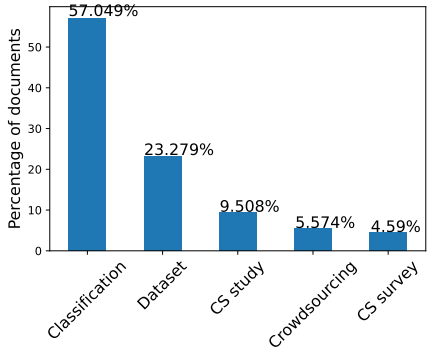


Figure 2.3: Distribution of computer science literature focusing on *OCL*

We identified 5 research directions in the computer science literature, that integrate

the different steps of the pipelines (literature surveys, statistical studies, classification methods, creation of datasets, and crowdsourcing tasks to collect labels), with a strong bias towards classification methods (Figure 2.3). There are especially few papers interested in crowdsourcing methods despite the challenge of obtaining high-quality *OCL* labels with such ambiguous and subjective *OCL* [318]. This hints at many research opportunities, especially around the biases contained in datasets, and studies to better understand *OCL*.

Next, we investigate the biases in detection pipelines. We pass current practices through the new requirements coming from the semantic and contextual mismatches to identify limitations, challenges and potential solutions. To further substantiate our critical analysis, we situate literature on machine learning biases and unfairness [812] in the present pipelines.

2.4. DATASET CONSTRUCTION FOR THE DETECTION OF *OCL*

We now analyse the datasets and data engineering pipelines used in online conflictual language (*OCL*) detection systems. While the process of creating a dataset is long and costly, out of the 194 publications for which experiments have been conducted, only 33% of them use an already-existing dataset (5 do not specify the dataset used). Such numbers motivate the need to understand the specificities of data pipelines, which do not seem standardized. We critically reflect on the pipelines, and their biases. In light of the recent research on data excellence [201, 623, 872], this surfaces new challenges to adapt the pipelines to the types of *OCL* targeted and the various applications in which the systems might be applied.

2.4.1. DATA SAMPLE COLLECTION

DATA RETRIEVAL

Data sources. Data samples are collected from various sources on the Web (Table 2.3). Twitter is used in majority due to its popularity and the easiness to get data, while other social media (Formspring, YouTube, MySpace, Wikipedia and Facebook) are used less [537]. Various sites such as the news website `Gazzetta.it` [625] usually specialized in one topic like sport or politics and discussion forums such as voat, 4chan or reddit are also investigated. Table 2.4 shows the distribution of languages in the publications, and highlights a strong unbalance between English (74.4%) and the other languages present only in 1 to 6 papers.

Yet, recent works exhibit efforts towards the diversification of the objects of study. Datasets are created for less studied languages such as Hinglish [421, 181], Bengali [458] and Arabic [183, 324], revealing new challenges pertaining to the particular language structures (e.g., in Hinglish, the grammar is not fixed, the written words use Roman script for spoken works in Hindi [421]); and for less common social media platforms (e.g., YouTube comments [458, 183]).

Following these works, we consider worth building new datasets to investigate more sources and languages, and increasing the research on cross-sources for more adaptability of the models [316]. Machine translation models in conjunction with English-based classifiers could also be investigated, especially for datasets that mix multiple languages.

Table 2.3: Dataset sources distribution.

Data source	Count
Twitter	98
Formspring	18
News site	16
YouTube	14
MySpace	14
Forum	13
Wikipedia	12
Facebook, individual or group conversations	11
Instagram	9
Yahoo	8
Other content-sharing social media	7
AskFM	7
Website (non social media, e.g., Tumblr, Whisper)	6

Table 2.4: Datasets Language Distribution.

Sample language	Count
English	157
Indonesian	6
Japanese	6
Dutch	5
Spanish	4
Portuguese	4
German	4
Arabic	3
Hindi	3
English-Hindi	3
French	2
Korean	2
Greek	2
Italian	2
Bengali	1
Russian	1
Turkish	1

Data mining methods. Most datasets are collected by retrieving samples which contain specific elements, such as abusive words [386], hashtags and keywords from controversial politics sites [118], or offensiveness dictionaries [675]. Several papers use snowball sampling [652, 373] or variations such as first retrieving tweets based on hashtags and then all the other tweets from their authors [817]. Others are retrieved by crawling entire pages selected for their likeliness to contain online conflictual language (*OCL*) (e.g., anti-Islam pages [831], offensive blog posts [221], public celebrity pages [250]), or by crawling and randomly sampling social media feeds [584, 753]. Additional filtering based on keywords or negative vocabulary is sometimes applied to maximize the number of *OCL* samples [641]. Similarly to psychology studies, the authors of [692] manually create cyberbullying scenarios from which students write an entire discussion used as dataset.

15% of the classification papers simplify the detection task by distinguishing smaller tasks of sub-topics that share similar properties. Researchers use datasets for specific *OCL* sub-type (e.g., datasets on sexism and racism for hate speech [638, 868, 933, 647, 614, 274, 867], on hateful speech towards black people, plus-sized individuals and women [703], or towards refugees and Muslims [933, 131]), or domains (e.g., news, politics, entertainment, business for insult detection [778] or disability, race and sexual orientation for hate speech [144]).

Introduction of biases. Each parameter set-up for data collection biases the dataset. The choice of data source, keyword for retrieving initial sets of samples, and languages for these queries directly impact the type of users for which the subsequent trained model will show good performance. Less obvious choices also skew the data distribution. For instance, through the selection of random samples from a forum history; or by selecting only the first posts. In both cases, the topics discussed might be more or less detailed, or the authors of posts might use more or less strong *OCL*. Skews are also intro-

duced by a crawler's (human, or automatic) browser setting, e.g., due to the geographical region, or search habits. Poletto et al. [640] discuss further certain of these biases in their survey. The period of time when the dataset is collected is also of importance. This concern is highlighted in computer vision, such as for the Pascal VOC dataset [368], reportedly collected in January, and composed of an above-average number of Christmas trees, as images in Flickr (the media they used) were ordered by recency. Machine learning models for *OCL* detection are especially sensitive to the events contained in the data [262], as these events shape the type of language and topics the models can interpret. Ptaszynskia et al. [645] recommend regularly collecting samples to update datasets with the most recent vocabulary. Sampling per keyword also introduces biases in the datasets [264]. The samples retrieved often contain words considered rude, while more subtle forms of *OCL* might not be accounted for. Founta et al. [264] instead propose to collect data by combining random sampling and tweets retrieved using keywords.

These biases become harmful when they skew the data distribution away from the expected one, or enforce discriminatory associations between attributes. According to the bias framework of Suresh et al. [812], *representation biases* manifest when the training data distributions integrate few information around underrepresented populations, leading to low model performance. This definition could be expanded to over-represented populations, for which a model might learn spurious correlations, and to "population" as either individuals or other kinds of concepts such as conversation topics.

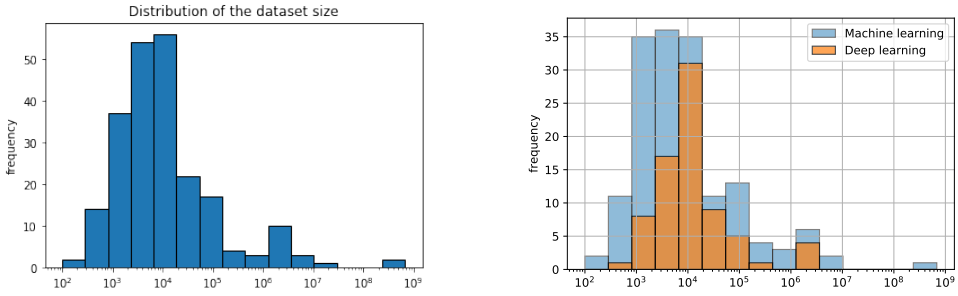
Various fields (e.g., linguistics) study the different strategies employed in order to express *OCL*; for instance, when expressing hate [47]: othering, stereotyping, conceptual metaphors, implicitness, constructive and fictive dialogues. Linguistics identifies these strategies for individual topics – e.g., "conceptual metaphors in comments related to migrants in Cyprus"; or media studies, –e.g., "in the case of racism, it was found the use of vicarious observation, racist humor, negative racial stereotyping, racist online media, and racist online hate groups. The online hate against women tends to use shaming. [..] flaming, trolling, hostility, obscenity, high incidence of insults, aggressive lexis, suspicion, demasculinization, and dehumanization can inflict harm" [163]. This information could be exploited in order to verify the diversity and representativeness of the samples collected in a dataset.

Dataset collection parameters are not always aligned with insights from psychology. While psychology puts forward context as important for classifying *OCL*, most posts are stripped down from their meta data and conversational context. Pavlopoulos [626] did not find any interaction with the title and the previous sentence of a post, yet context can be broader, e.g., the whole discussion, and merits further investigation. Multiple challenges in reference to this mismatch are discussed in subsection 2.3.4.

DATA PROCESSING

Data augmentation methods. Figure 2.4a shows the distribution of the number of training data employed in the classification tasks, with a majority of datasets around 1000 and 11000 samples. As expected, deep learning approaches make use of larger datasets (about 10000 samples) than traditional machine learning approaches (about 5000 samples) – Figure 2.4b.

Despite needing large datasets, only 14% of the classification papers mention explicitly data augmentation techniques, mainly to balance datasets. This is common as Web



(a) General distribution.

(b) Distribution per classification method.

Figure 2.4: Distribution of the number of training data employed in classification tasks.

platforms contain a majority of non-*OCL* text (e.g., abusive tweets only represent 0.1% to 3% of tweets [264]). This extreme unbalance explains why certain papers further retrieve data using *OCL* seed words, instead of performing synthetic data augmentation. Out of the 69 papers whose figures are available, 39% have a balanced dataset.

Data augmentation is performed either by over-sampling or by under-sampling certain classes or both. 9 papers randomly duplicate the minority class samples and 8 remove samples from the majority class. 6 papers employ the Synthetic Minority Oversampling Technique (SMOTE) for over-sampling by creating artificial data samples in the feature space. 2 create synthetic data with two-way sample translation and sliding windows [700], or with random sample generation with a character encoding and introduction of known online conflictual language (*OCL*) words in these sequences [733].

The different data augmentation methods do not all perform well for each classification task [171]. Thus, we not only recommend to investigate data augmentation further, but we also propose to create a list of large datasets for each type of *OCL* so that researchers have common benchmark datasets for evaluation, as suggested for abuse detection by Jurgens et al. [415]. Poletto et al. [640] propose a review of existing benchmark corpora that supports the identification of missing text corpus. Existing datasets could be merged together to augment their size. Deep generative models are also recently investigated to synthesize new data samples automatically, with promising results [886]. Further investigation of their conditions of applications, and of the choice of hyperparameters, would be beneficial.

Next to balancing a dataset, Park et al. [613] augment their dataset by substituting female entities to males ones and vice-versa, to reduce gender bias. The validity of the synthesized data samples would merit being further investigated in relation to the specific types of *OCL* of each use-case, especially when studying multiple sub-categories of *OCL*.

Introduction of biases. As a sign of representational biases, Grondahl et al. [316] show that models performing well on a dataset with the same distribution as the training dataset, perform poorly on other datasets; but perform equally well when they are retrained on a dataset with this other distribution. This suggests that the architecture of

the model is not the primary factor for the resulting performance, but that the datasets themselves all contain their own biases, hindering generalization to other datasets.

Data augmentation and processing reinforce or introduce representational biases. For instance, most data instances that are representative of a certain *OCL* might deal primarily with a certain topic. Augmenting the dataset for the *OCL* class would then reinforce the presence of this topic in association with the *OCL* label. Also, basic preprocessing activities such as stemming and lemmatization can remove useful indications, e.g., gender word endings in gendered languages, skewing the data towards one single type of representation. The curation of misspellings might skew the representation of populations that frequently use such spelling. Grondahl et al. [316] experimented with natural-looking adversarial perturbations –which could be misspellings– and showed that models are not robust to those. Besides, misspellings are not all spelling mistakes, but can be meaningful, and vary the interpretation of a sentence from the "clean sentence". Curating the data then prevents a model to learn such new types of interpretations.

In other domains such as computer vision [667, 351, 568], models are made less brittle by augmenting the datasets with natural or adversarial perturbations that could arise at deployment time. We suggest to test similar solutions in the context of *OCL*. Especially, brittleness to natural perturbations such as voluntary or unintentional misspellings might be partly due to the ways data are processed: when misspellings are resolved, the models are not trained on such diverse, possibly adversarial inputs, increasing their brittleness.

DATA SPLITTING

Dataset splitting is not standardized in the *OCL* detection pipelines. Arango et al. [35] showed it can lead to overestimation of models' performance. When it is done after feature engineering (or after data augmentation and curation), information from the test data is leaked into the training data as the feature extraction methods might rely on data distributions, resulting in obtaining high performance in laboratory settings but low performance in deployment.

This highlights general issues with the management of data in research settings. If the data are studied along time, it is important not to sample them randomly but follow this temporal sequence, in order to observe how generalizable a dataset from one time window is to another time window. These and more issues are also identified in the general data management literature for machine learning [726]. The implementation of common benchmark structures respecting these data management rules would support the propagation of good practices in the preparation of datasets for the training and evaluation of models.

2.4.2. DATA ANNOTATION COLLECTION

Here we discuss how dataset annotations are collected. Annotation refers to the labeling of data instances (e.g., a sentence or a tweet) that might contain online conflictual language (*OCL*). These annotations are usually collected by aggregating the inputs of multiple annotators into a single label, in order to ensure its quality. 95% of the 80 papers with available information go through this human annotation phase. A few papers instead

use machine learning [148], inference from data context [817, 703], or semi-supervised learning [276] to infer labels. Notably, some works mentioned by Fortuna et al. [263], build lexicons of *OCL* [242, 876] to train better classification algorithms. We do not include them here as they do not correspond to the annotation of evaluation datasets, and do not detail their crowdsourcing setup.

SET-UP OF THE ANNOTATION PROCESS

Instructions to the annotators. A binary question is typically asked to the annotators (the answer "undecided" is sometimes added), potentially with a rating [143, 170]. However it is argued in psychology literature that rating comments on a valence scale is too vague for the annotators, who prefer binary questions [778, 777]. Closer to psychology which asks annotators to rate several propositions, Guberman et al. [318] investigate perceived violence of tweets through an adapted version of the multiple proposition Buss-Perry Aggression Questionnaire (BPAQ). Using 6 annotators on Amazon Mechanical Turk and 14 gold questions (12 correct answers required), they still found 30% disagreement that they partly explain with the non-adaptation of the questionnaire to tweet violence.

Out of 74 papers using crowdsourcing, only 32% mention giving a definition of the concept to annotate to the annotators, such as detailed offensiveness criteria [3][4] and hate speech definition[5]. Gamback et al. [274] through several crowdsourcing tests provide a detailed question to the annotators [6]. Not providing clear definitions is an issue because the annotators might have different definitions of online conflictual language (*OCL*) in mind, leading to collected data labels that would not be suited to the application.

Data annotators. The annotation tasks are conducted on crowdsourcing platforms or programs created by the authors of the publications. Certain papers show that the type of annotators employed influences the quality of the annotations. CrowdFlower (now `Appen.com`), expert and manually-recruited annotators are equally used (23.7% each), while students of universities (13.8%) and Amazon Mechanical Turk (15%) are less. The expert category comprehends authors themselves, researchers of similar fields, specialist in gender studies and "non-activist feminist" for sexism annotations, persons with linguistic background, trained raters, educators working with middle-school children, and people with cyberbullying experience.

[3] "A tweet is offensive if it 1) uses a sexist or racial slur; 2) attacks a minority; 3) seeks to silence a minority; 4) criticizes a minority (without a well-founded argument); 5) promotes, but does not directly use, hate speech or violent crime; 6) criticizes a minority and uses a straw man argument; 7) blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims; 8) shows support of problematic hashtags. e.g., "#BanIslam", "#whoriental", "#whitegenocide"; 9) negatively stereotypes a minority; 10) defends xenophobia or sexism; 11) contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria." [868]

[4] "tweets that explicitly or implicitly propagate stereotypes targeting a specific group whether it is the initial expression or a meta-expression discussing the hate speech itself" [276]

[5] "the language which explicitly or implicitly threatens or demeans a person or a group based upon a facet of their identity such as gender, ethnicity, or sexual orientation" [275]

[6] "Does the comment contain a personal attack or harassment? Targeted at the recipient of the message (i.e. you suck). Targeted at a third party (i.e. Bob sucks). Being reported or quoted (i.e. Bob said Henri sucks). Another kind of attack or harassment. This is not an attack or harassment."

Annotation aggregation. Among the 50 papers for which the information is available (out of the 74 papers using crowdsourcing), 49 papers aggregate the annotations from multiple annotators into binary labels. 78% use majority-voting, 10% filter out samples for which there is no full agreement between annotators, 8% create rules which define how to aggregate according to different scenarios of annotations (e.g., majority-voting and removal of the samples with the highest disagreement rates and the samples for which the annotators agreed they are undecided [142]). One paper uses a weighted majority-vote scheme [373]. Only Wulczyn et al. [885] derive percentage from the annotations.

Annotation quality control. 32.4% of the papers mention techniques to obtain high-quality labels. Within the annotation task, they investigate using precise definitions and clear questions to remove ambiguities [689]. After the task, annotations are aggregated to resolve disparities between annotators' opinions, and low quality annotations or annotators are filtered, with quality scores computed over the history of the annotators, the time they take to answer each question, or their answers to gold questions [375].

Half of the tasks have 3 annotators, 15% make use of 5 annotators and 22% of 2 annotators. Using an odd number of annotators enables to break ties in annotations with majority voting, while using 2 annotators is cheap and fast. The rest of the tasks employ 1, 4, 6 or 10 annotators. The papers using more than 5 annotators per sample are rare, most probably because of the cost. Using only the cases of full agreement among amateur annotators produces relatively good annotations compared to expert annotators, and they suggest to use experts only to break the ties of the amateur annotators [867].

Different metrics are employed to evaluate the annotation quality by measuring the agreement between annotators (Figure 2.5). Most papers use Cohen's Kappa for 2 annotators and Fleiss' Kappa for more. 22.9% of the papers mention "inter-annotator agreement" or "kappa" scores without further precision. Krippendorff's alpha and the percentage agreement are less adopted, the second one making a possibly wrong assumption that the majority is correct [545]. In the publications, we notice a high proportion of low Cohen's Kappa and Fleiss' Kappa scores (under 0.6) for tasks with 3 or 5 annotators, which proves the difficulty to design unambiguous tasks and hint at the subjectivity of the concepts to rate.

BIASES IN THE ANNOTATION PROCESS

The data annotation process introduces various types of biases.

Identification of mismatches. Here we take the hypothetical scenario of developing a dataset for aggression language. Certain definitions of aggression highlight the need for looking at the context of a sentence, at the behavior of its author, and at the person judging this language, to understand how a sentence would be perceived, e.g., aggression is "neither descriptive nor neutral. It deals much more with a judgmental attribute" [572]. Psychology identified the variables that influence this judgement, mostly "cultural background" [139], the role of the judge, i.e. aggressor, target, observer, etc., "norm deviation, intent, and injury", but also "the form and extent of injuries actually occurring" [513]. To

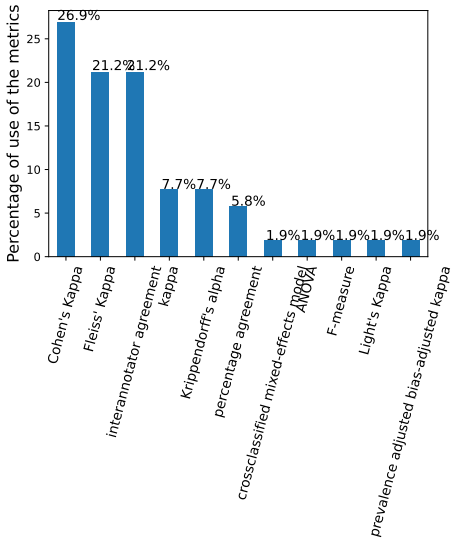


Figure 2.5: Distribution of the metrics used to evaluate the annotations.

obtain a controlled and realistic dataset and reduce ambiguity, these pieces of information around the annotators of the language would be needed, the annotators role (e.g., victim or observer) should be decided, and the context of the sentence (e.g., harm caused by a sentence) displayed. A similar example is the perceived offensiveness of group-based slurs, which depends on the perception of the status of the target group [353]. In this case, both the context and observer are of importance since the social status of a target group could be uncovered from context knowledge but can also depend on the perception of the observer.

These issues resonates with the historical biases in machine learning ethics literature [812]. In the dataset, there is a mismatch between the judgements of the annotators, the judgements of the actual targets of a *OCL*, and the judgements from external observers. Consequently, the dataset is not aligned with what the machine learning model is expected to learn.

Missing context information. Psychology literature showed that for many conflictual languages, the sample context influences the perception of a sample. Most crowdsourcing tasks however do not specify it, neither in the instructions nor within the sample presented to the annotator [167, 727]. Guberman et al. [318] put forward the insufficient context which leaves many aspects of the text to interpretation, as a reason for disagreement in harassment annotations. Golbeck et al. [296], while not including any context in their corpus, acknowledge this limitation and develop precise annotation guidelines that aim at removing ambiguities stemming from the absence of context. Ross et al. [689] provide a definition of the *OCL* to annotate, and find that the task remains ambiguous, suggesting that even for objective tasks, context information might be missing to provide an objective rating.

The type of context to include and its framing (e.g., a conversation, structured information about multiple characteristics) remain to be investigated to address ambiguities, while controlling the cost of the annotations. Pavlopoulos et al. [626] have shown that annotations with conversational context (post and its parent comment, as well as the discussion title) significantly differ from annotations without it. Sap et al. [721] have primed annotators with dialect and race information explicitly to reduce racial biases in annotations (more samples written in African American English than in general American English are labeled as offensive). Creating datasets that tackle single specific contexts such as "hate speech against immigrants" is also a direction to investigate [86].

Lack of annotator control and information. Psychology highlights that many *OCL* are subjective. Linguistics also shows the diversity of interpretation of *OCL* by different communities or within a same population [47]. For instance, a study shows that in Malta, participants typically identify homophobic comments as hate speech, but not necessarily xenophobic ones, and explains it with the recent acceptance of the LGBTQ community in the Maltese society, while "migrants are still very much left on the periphery". Similar studies in other regions of the world would probably lead to different conclusions, illustrating the importance of the annotator background. Hence, choices in the crowdsourcing task design that impact the pool of annotators (country of origin of the annotators, language, expertise, educational background, and how they are filtered) integrate implicitly biases in a dataset.

Psychology indicates characteristics of an individual that impact one's perception of a sentence relative to an *OCL*. Some of these characteristics are also observed in computer science papers, such as the differences of annotations based on gender [318]. Communication studies also investigate the characteristics of an individual that impact their willingness to censor hate speech, and identify age (e.g., "older people are less willing to censor hate speech than younger people"), neuroticism, commitment to democratic principles, level of authoritarianism, level of religiosity and gender [466]. Such factors could possibly also impact one's attitude toward annotating hate speech. While the design choices do not map to these characteristics, creating schemes to control or at least measure them, is a valuable research direction. Certain crowdsourcing frameworks [79] are a first step towards this control. Verifying that the same characteristics apply in the online and offline contexts is also important following previous contradictions, e.g., one computer science study observed that annotators from both genders usually agree for clear cases of misogyny and disagree for cases of general hate speech [880], contradicting findings in psychology literature.

Additional properties of the annotators, not investigated in psychology, can bias the datasets. For instance, annotators from crowdsourcing platforms, who have no training on what hate speech is, are biased towards the hate label, contrary to expert annotators [867]. Research is hence also needed in assessing the level of education around *OCL* that annotators have, in educating them, and in maintaining them engaged for more annotation tasks.

Simplification of the annotations. The way the annotations are processed creates biases. Aggregating the annotations into single labels does not allow for subjectivity and

skews datasets towards certain types of perceptions, generally the majority opinions [58]. This might raise issues of unfairness –non-inclusion of certain opinions–, and reinforce filter bubbles. For instance, Binns et al. [103] show that a toxicity detection algorithm performs better on annotations from male users than from female ones and is consequently unfair to women. This reflects *aggregation biases* [812]: a single dataset to train a single machine learning model for a whole platform is collected, whereas different populations need adaptation.

Subjectivity brings new challenges in measuring and obtaining "high-quality" annotations. Measures of quality are now centered around agreement –the lowest the disagreement, the highest the quality–, and post-processing methods use the majority opinion, yet the majority is only one perception of a subjective *OCL*. Instead, methods should filter out annotations that are obviously incorrect –often due to spams– or erroneous for different individuals, while accounting for the existence of multiple relevant and disagreeing judgements. For that, works from the human computation community, such as CrowdTruth [43] which provides metrics for the quality of annotations and annotators without assuming the existence of a unique ground truth, could be investigated. More annotators might be needed, and schemes to infer relevant clusters of annotators could be investigated to trade-off between quality and cost considerations. Mishra et al. [554] noted that in digital media, a small amount of users frequently give their opinions, ranking positively highly offensive posts –a form of bias towards the opinion of these few users. The researchers propose a semi-supervised method in order to identify these biased users and correct the ratings.

Leveraging psychology and human computation methods. Research from other fields could be adapted to improve *OCL* annotation pipelines, as recommendations from crowdsourcing literature or psychology are not necessarily followed for now. Only 32% of papers mention methods to ensure a level of quality (e.g., golden questions, annotator quality score, precise definitions of the terms, etc.) and few papers employ more than 5 annotators per sample, whereas crowdsourcing literature encourages that. Taking inspiration from psychology and judgement collection methods can also be a promising direction. Psychology studies use multiple questions with scales, whose answers are aggregated to collect the perception of each person (e.g., 10, 6, 3 propositions on [1;9], [1;6], [1;12] scales [591, 114, 200]). To measure offensiveness, participants rate images visualising a scenario along how comfortable, acceptable, offensive, hurtful, and annoying they are on a 7-point Likert scale [877]. Cunningham et al. [203] show scenarios with 4 situations to participants, who select the most offensive one. Example scenario and situation are respectively attending a men's basketball game and "A Caucasian, female said: "Of course we lost. We played like a bunch of girls."" While these studies are not specific to online conflictual languages (*OCL*s), the general method could be used, and the specific questions investigated. The challenge of asking such questions while maintaining the cost low would become important.

2.5. CLASSIFICATION MODELS FOR THE DETECTION OF *OCL*

In this section, we discuss the algorithmic methods used for online conflictual language (*OCL*) detection. We focus on the features extracted from data, on the algorithms, and on the selected evaluation procedures. We aim at identifying implicit biases integrated into the design choices of the detection pipelines.

2.5.1. FEATURES FOR CLASSIFICATION

TYPES OF FEATURES EXTRACTED FROM THE DATA

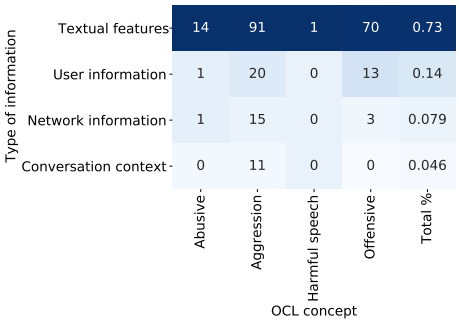


Figure 2.6: Type of information used by the classification methods according to the *OCL* concepts.

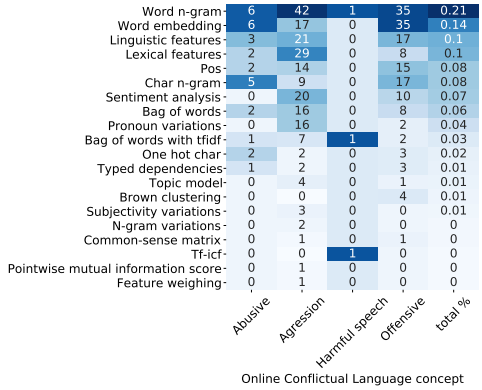


Figure 2.7: The textual features per *OCL* coarse-grained concept used in the classification papers.

Features employed in the classification models use four main types of information detailed below, and summarized in Figure 2.6.[7]

Textual features. Advantages and disadvantages of the features are explained in [263]. Textual information is represented differently depending on the classification methods. Word n-grams, bag of words (BoW) and embeddings are employed in majority because they are adapted inputs to machine learning classifiers. Word n-grams represent more information (order of the words) than BoW, which improves the classification performance, while word embeddings are recently developed for deep learning. Certain features are rarely investigated (common-sense matrix [225], tf-icf (Inverse Category Frequency) [485], pointwise mutual information score [582]), and merit more research in the future. The distributions of the textual features used across online conflictual language (*OCL*) coarse-grained concepts (Figure 2.7) are mostly similar, which indicates a potential lack of adaptation of the individual features to each task at hand.

Information about the users (emitter and reader). This is the second most used information for classification. It includes the user popularity in the social media based on

[7] Interested readers can refer to Schmidt et al. [727] and Fortuna et al. [263] for an extensive explanation of the properties of each feature.

the number of followers and friends, the user activity based on the number of posted and liked tweets [942, 264, 207], her gender [867], age [207] and location [339, 868], the subscribed lists and the age of the account [264], and information extracted from the conversation history such as the frequently used terms [867], the tendency to use *OCL* [638] or the Second Order Attributes representation of the link between documents and users [33]. These characteristics might be studied for a user across social media platforms [209].

Information about the network of the users. Often it consists in measuring how much a user reciprocates the follower connections she receives, "the power difference between a user and his mentions, the user's position in his network (hub, authority, eigenvector and closeness centrality), as well as a user's tendency to cluster with others" [264], but also graph metrics computed over the combined social networks of the sender and receiver [380, 784].

Conversation context. This is the conversation [88] or the set of questions and answers [686, 588] surrounding the data samples, the images found with the textual samples in the social media [373] and their captions [374], information about the parent-child relationships of the samples in the conversation [495], or information about the samples themselves such as the popularity of a post among its social media [816, 374] or its publication time [374].

FEATURE SELECTION

Certain papers start with a large amount of input features and then decrease the dimensionality to improve the classification performance.

For this, 12% of papers use feature selection methods: Chi-square [118] (5), Singular Value Decomposition [225] (5), information gain [605] (3) or mutual information [730] (2) based selection, Fisher score [939], recursive elimination with logistic regression (training a classifier with all the features but one, and eliminating the one leading to the worst performance) [730] or simply evaluating a classifier on different subsets of features and selecting the one with the best performance [323], backward selection (removing variables with high correlation) [374], test statistic (Student t-test) [730], PCA [196], Latent Semantic Analysis [373].

Feature weighting is used with SVM scores [642], logistic regression weights [730], or by computing a score which represents the easiness to falsify the outputs of the classifier with one feature and selecting features based on this score [278].

Yoshida et al. [910] compute an entropy score indicative of whether a word corresponds to a sentiment and define a set of rules to select the words to keep, and Lee et al. [485] compute the less common words in a set of documents.

INTRODUCTION OF BIASES

Measurement bias. The choice of features automatically biases the model towards using certain types of information, and biases its outputs towards specific types of errors. This is a *measurement bias* [812], where the choice of features might leave out factors that are relevant for inference. In the following, we identify various measurement biases.

Mismatch with psychology. We identify measurement biases in the way features are engineered. The inputs to the classification methods are mostly textual information. Although psychology shows that the context surrounding text also impacts *OCL* perception, only 23% of papers use additional information (Figure 2.6). Non-textual features are mostly used for the classification of aggression language, possibly because it is characterized by the behaviour of users, however the other types of languages are also impacted by context. The way the feature dimensionality is reduced also impacts the type of information used by a model.

The information used often does not correspond to the variables identified by psychology, which might explain performance issues [380, 933, 626]. Measurement biases also reflect the non-consideration of subjectivity. Adding to the common features other features describing users would allow to personalise inferences, what would render the models more inclusive of various opinions. One main challenge here would be to define precisely which information should be extracted from the datasets into features, and how to represent it effectively.

Lack of *OCL*-dependent features. Several experimental studies show the difficulty for machine learning models to distinguish between different *OCL* [530, 850] (e.g., difficulty to differentiate between hate speech and profanity [530]). Also, our systematic survey shows a lack of adaptation of the features to each specific *OCL*. While feature engineering might not seem entirely relevant with deep learning, we suggest to study the introduction of hand-crafted features to differentiate between these *OCL*, inspired from the psychology literature and our categories in **??**. For example, someone interested in offensive language could explicitly integrate the identification of the targeted individual or community in a language sample, instead of letting the machine learning model eventually discover these characteristics. This comes hand in hand with creating more adapted datasets where the different types of *OCL* have to be well-represented, and the necessary information present.

Recent works show promising results in this direction. Training word embeddings on a hate corpus and appending manually-crafted features specific to the target class achieves higher accuracy performance than pre-trained embeddings or more traditional features (e.g., n-grams), for the classification of various intensities of islamophobic hate speech [850]. Zhang and Luo [935] extract more informative features than classic ones like n-grams, by using deep learning structures that learn relations between words.

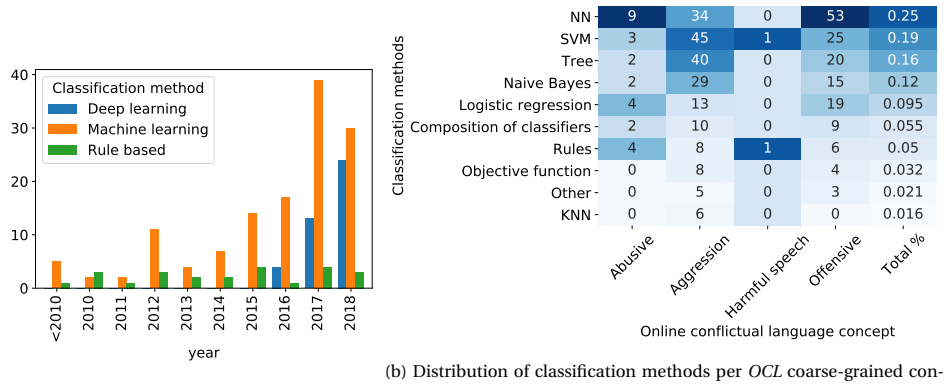
Low classification performance also come from the lack of adaptation of the features to the specific ways people use online conflictual language (*OCL*) in different social media, such as making spelling "mistakes", mixing languages in informal language [386, 461], using language which follows evolving trends over time [584, 461], using implicit *OCL* [478]. We recommend to specifically investigate how to integrate these characteristics into future models. For instance, Alorainy et al. [22] extract features specifically to identify othering language, Bansal et al. [76] and recent publications in ACL workshops [53] focus on humour and sarcasm.

Discriminatory features. Recent concerns have been voiced around the discriminatory character of certain features, especially those ones coming from word embeddings.

Caliskan et al. [155] adapted a psychology test (Implicit Association Test) to measure biases in word embeddings, and showed that these embeddings reproduce historical human biases. Garg et al. [279] showed that training embeddings on text corpora from different time periods incorporates in these embeddings the job-related biases from the various periods. Methods exist to debias such embeddings [119, 936, 135]. Although not focused on *OCL*, they could be investigated as some of them rely on training word embeddings to extract adapted features. One might search for the biases introduced when word embeddings are trained on *OCL* corpora, instead of general natural language processing corpora.

2.5.2. METHODS FOR CLASSIFICATION

OVERVIEW OF THE CLASSIFIERS



(a) Evolution of the classification methods over years.

(b) Distribution of classification methods per *OCL* coarse-grained concepts. Colours represent the frequencies of methods per concept (white to blue, lowest to highest frequency).

Figure 2.8: Quantitative analysis of the classification methods

We note three main trends in the classification methods: rule based models, machine learning models –that we define as simple classifiers–, and deep learning models. 4.7% of the papers combine several models with ensemble and boosting methods. Although computer science papers report performance measures, it is difficult to tell which are the "best" methods as the measures are not obtained from the same datasets.

The use of machine learning methods has increased over years since 2012 (Figure 2.8a), following the general increase of *OCL* research. Research on deep learning for *OCL* started in 2016 with the general increase in deep learning research, and its amount increased quickly, almost catching up with machine learning research. Research on rule-based methods has been constant over years and rarely adopted.

Among these 3 categories, various methods are used. A majority of machine learning papers use Support Vector Machines (SVM), tree-based classifiers (decision trees and random forests), Naive Bayes classifiers (NB), Multi-Layer Perceptron (MLP) and Logistic Regression (LR). Deep learning papers mainly investigate Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and their combinations. Figure 2.8b shows that regular deep learning, SVM, tree-based and rule-based classifiers concern

every type of *OCL*, while research on naive Bayes classifiers, composition of classifiers and optimization of application-tuned objective functions has been sparsely conducted especially for harmful and abusive languages.

TRAINING PROCESS

Most publications follow the same pipeline: dataset collection and model creation. However, 5% of the papers diverge. During the training process, they perform active learning [554], or semi-supervised learning where part of the training data samples do not have labels but these samples are still used (often by label inference) [910, 657, 554]. They perform feature selection and classifier learning simultaneously [947]. Certain papers employ transfer learning by incorporating a learned word probability distribution in the target domain to the classifier for training efficiency [604, 12, 733], or to reduce gender biases [613].

Besides, a few papers compare the performance of models trained on the whole dataset, or trained by cutting the dataset into domains and by learning a multi-class classifier (one class per domain) (e.g., cyberbullying related to race, sexuality and intelligence [225, 226]). Other papers detect the sub-types of the concept instead of simply detecting the coarse-grain concept (e.g., detecting cyberbullying by classifying curse, defamation, defense, encouragement, insult, threat and sexual talk [839], detecting misogyny by classifying discredit, sexual harassment, threats of violence, stereotype and objectification, dominance, derailing [32]).

INTRODUCTION OF BIASES

The choice of classification algorithm and its hyperparameters participates in the introduction of various biases in the outputs of classification models.

Aggregation bias. Such bias is defined by the development and application of a single machine learning model on various distinct populations [812]. This practice is problematic for subjective *OCL*. A solution could be to learn distinct models on sets of annotations from different populations, possibly also taking into account the context of application and learning distinct models for different platforms for instance. Sharing some information across models while fine-tuning them for specific context remains to be investigated in order not to require too large amount of data and too large computational resources.

Mitigating discriminatory biases. A large body of literature on machine learning for structured data highlights unfairness issues for decision-making systems, propose metrics [848], mitigation methods [268], and toolkits [91] to explore the causes of unfairness, and to support industry practitioners in integrating these formalisations of fairness into their practices. Recent works have introduced different methods to debias the outputs of NLP models, e.g., by transforming the features employed, by modifying the optimization objective employed to train a classifier (e.g., adversarial training of deep learning models with a regularization term corresponding to the protected attributes at hand [888]), or possibly by transforming the outputs of the classifier [807]. A more extensive account of such works is given in [807]. In certain cases, the training process is also modified to

involve a bias expert [181]. Few recent works propose sample weighing methods to account for dataset biases respectively in toxicity or hate speech detection tasks [922, 567], and integrate knowledge bases to correct datasets from biases by substituting words indicator of identity by more general entities [56].

Most works are not specific to *OCL* and need adaptation. For example, some works do not easily translate to classification tasks of more than two classes, but this becomes necessary for *OCL*. Trade-offs between discriminatory biases and performance measures [613] nudge for works at the intersection of natural language processing and human-computer interaction to understand how to set acceptable thresholds for the metrics. Toolkits could also be developed. Besides these more usual notions of unfairness, a new type of unfairness with regard to the social network centrality of a potential victim of cyberbullying is also exposed in Singh et al. [762], and would merit further investigation.

Debugging biases and other errors. Investigating how to apply interpretability methods to *OCL* classification tasks could enable to understand specific causes of the low performance or unfairness of the classifiers for specific samples. Little effort has investigated such direction until now: Risch et al. [681] with usual interpretability methods and Cheng et al. [176] with the causality angle for performance, and Kennedy et al. [429] for biases.

Human-in-the-loop methods could be developed to identify the shortcomings of trained models, by asking humans to generate samples that lead the model to a wrong prediction. This could serve to identify more social biases, or simply to make the model more robust to tricky samples. In this direction, Dinan et al. [227] asked crowdworkers to generate sentences that would break their offensiveness detector, and noted that crowdworkers identify samples of a nature which is rare in the original dataset, with less obvious profanity but more figurative language and language that requires background knowledge to be interpreted.

2.5.3. PERFORMANCE EVALUATION

EVALUATION DATASET

Data samples. To evaluate the models, the dataset is divided into training and test set, and performance metrics are computed on the test set. Some works now also evaluate their models on other datasets which have different distributions, to understand how generalizable the models are. This emulates the production set-up, where new data samples are continuously inputted, for which the distribution might differ from the training one when new users and new context are added. Few works [584] evaluate the classification performance along time.

Ground truth. While most papers consider binary labels as ground truth, some aggregate the crowdsourced labels into continuous scores to investigate whether a model learned the distribution of judgements or the majority labels. A distinction between the data samples whose labels received full consensus and the data samples of lower consensus is also sometimes made [478] for explanation's sake, i.e. better understanding where errors come from.

EVALUATION METRIC

A small number of metrics is used: F1 score (macro, micro or average) (23.8%), recall (22.9%), precision (20.5%), ROC-AUC (7.9%), accuracy score (14.3%), true negative, false negative and false positive rates (4%). Accuracy is discouraged because its measure is impacted by unbalanced datasets. Accuracy, precision and recall are calculated on average for all the classes or for the different classes separately. Few papers use the Cohen's Kappa score [703, 226], the Kappa statistic [701, 225, 167], the Spearman correlation [885, 624], the precision-recall curve with the precision-recall breakeven point [278, 509, 778] and the Hamming loss [700] as an evaluation metric. Others use error calculation based metrics such as the mean squared error [217, 575, 167, 554], the root mean square forecasting error and the mean absolute percent error [642]. Park et al. [613] use the False Positive and False Negative Equality Differences to quantify gender biases.

Some publications assess the time taken to train the models or the time to detect the online conflictual language (*OCL*) [911, 531, 684, 651]. Some papers further study the performance of the models by investigating in detail the types of sentences usually missclassified.

ACCOUNTABILITY AND TRANSPARENCY

There is generally no common dataset and evaluation metric to compare models. Benchmark datasets would ideally include context information and information about the annotators, and state clearly the scope of the dataset. Using the same metrics across publications which target the same goal would be helpful. The advantages of the less frequent metrics should be investigated. Reporting the pipeline used to build the datasets would allow to better understand their limitations and biases. As suggested by literature on transparency, datasheets [283] could support the controlled use of the datasets, both in research and industry. This relates to *deployment bias* [812], when a model is used for an application it was not built for.

REFINEMENT OF THE METRICS

Most frequent metrics reflect the accuracy of a model, which is not necessarily aligned with what end-users deem important. For subjective *OCL*, evaluations could be personalised to the different perceptions of users depending on their background [597]. To measure user satisfaction, metrics inspired from the machine learning fairness literature [848] could be adopted, e.g., measuring the accuracy of the model inferences for groups of users, and computing their ratio. These issues are termed *evaluation bias* [812], where the metrics employed or the scope of the evaluation dataset do not correspond to the type of samples or the goals for which a model would be used in practice.

Unfairness issues in datasets and classification outputs also need systematic investigation, for instance using existing fairness metrics. Yet, it is important to accurately interpret these metrics, as they might simplify too much the actual discrimination issues, and optimizing for them might not lead to fair results in practice [603, 731].

Critical studies [830, 99] have been published in computer vision, evaluating benchmark datasets and issues with performance metrics (e.g., top-1 accuracy might underestimate the performance of a model while multiple labels could be relevant), showing how they lead to correct or wrong conclusions. Inspiration could also be taken to develop better mental models of the functioning of the *OCL* detection systems.

2.6. BROADER CHALLENGES AROUND *OCL* RESEARCH

In this work, we used *online conflictual languages* (*OCL*) to refer to the multitude of hate related languages. We gave an overview of these concepts from a psychology and a computer science point of view. We then proceeded to a systematic survey of the classification methods and dataset collection methods used in computer science. We identified the main trends in the design of these methods, and reflected on the main biases that are incorporated into the detection systems, by drawing on the new insights from psychology literature and the consideration around the online context. We highlighted numerous implicit biases related to the semantic and contextual nature of many *OCL*, but also simply to the importance of a language's content in its interpretation.

To conclude, we now summarize these biases and reflect at a higher level on the causes of these errors, and the issues they reinforce. We identify additional challenges both of technical and structural nature. We particularly discuss various socio-technical research opportunities for the future, and question the structures that developed these biases within computer science research.

2.6.1. SUMMARY OF BIASES

In Table 2.5, we summarize the technical biases identified along the survey. These biases often arise from under-defined online conflictual languages in terms of semantic properties and contextual properties, or from technical difficulties in accounting for these properties. While the biases arise from different parts of the data and model pipelines, their harmful impact generally stems from the outputs of the machine learning models applied to real use-cases.

Data Collection	**Sample retrieval**	Source & time → contextual bias; Keyword and rank biases; Topic & language biases; Representation bias; Collection of context information
	Dataset processing	Data augmentation bias; Pre-processing biases
	Dataset splitting	Information leakage → Evaluation bias
	Sample annotation	Annotator *OCL* knowledge; Annotator background; Annotation instruction; Presentation of context; Annotation aggregation
Model	**Feature engineering**	Measurement bias (context, psychology); Discriminatory features
	Classification algorithms	Aggregation bias; Discrimination bias
	Performance evaluation	Evaluation bias; Data representativeness; Metric relevance

Table 2.5: Summary of biases introduced in the online conflictual language detection systems through the design of the data collection pipelines and of the classification models.

2.6.2. TECHNICAL CHALLENGES

ISSUES STEMMING FROM THE TECHNICAL BIASES

The biases identified resonate with multiple domains of machine learning research, especially unfairness, robustness to natural perturbations and to adversarial attacks, and model failures that come from the distribution mismatch between the training data and the data in deployment. Most issues are ultimately questions of ill-defined requirements. Developing methods to better identify the requirements of the systems prior to their development, and to test for such requirements, would allow to foresee such issues and

possibly correct for them [69]. A recent study (not from the *OCL* domain) refers to adjacent problems as underspecification of machine learning models [204], i.e. models trained on the same dataset with the same architecture but various seemingly "unimportant" hyperparameters (e.g. initialization seed) provide similar performance on a test set, but diverging performance on the deployment data.

As for natural perturbations, it remains to be defined what the nature of such perturbations is in the context of *OCL*. In computer vision, natural perturbations are generated artificially with prior knowledge of usual transformations of the data samples, and a model is trained and evaluated with the worst-case perturbation, or the average perturbation [351]. The equivalent in natural language could be spelling mistakes or intentional misspellings, variations of languages within a sentence, grammatical mistakes, etc. As for model failures, identification methods exist especially in computer vision, and rely on a human-in-the-loop approach to make sense of data samples, and cluster them into meaningful groups [75]. Similarly, designing tasks that crowd workers could perform in large scale for *OCL* needs attention, especially if their subjectivity is taken into account while attributing labels. Besides, a redefinition of model error formalization might be needed to adhere to this subjectivity. For computer vision and tabular data, bias mitigation methods are developed, often transforming the latent representations learned by the models [297], once the biases are identified. These methods could be similarly applied to *OCL* detection.

OTHER ISSUES

Similarly to other machine learning-heavy fields, *OCL* detection might be concerned with issues of privacy, explainability and accountability. Studying them for *OCL* might present new challenges. For instance, concerning explainability, an author might want to know why their text was flagged (local explanation), while a platform user would want to know about the general types of content flagged for them (global explanation). An unintentional author of *OCL* might need indications to express their ideas in a non-problematic way (to the extent this is), which could be inspired from works on recourse in machine learning. Few works answer these challenges in natural language processing.

As for privacy, issues could arise from the need for large datasets, or from the use of machine learning models. The sources of the datasets and the way they are stored might raise privacy issues if for instance, posts are collected from social media users –even though these posts are made public [124]. The annotation activities might also create privacy issues in cases where the data samples contain private information that the data annotator would be exposed to. A model trained on a dataset containing posts from specific individuals might also be "attacked" to identify which individuals were contained in the training set [844].

HANDLING *OCL*

OCL content can be handled in various ways. Besides filtering out the content –which might infringe freedom of expression–, or countering it, another recent avenue is to provide a warning to the recipient of *OCL* [833]. This could prevent harm of waiting for verification and removal, while not infringing freedom of expression. Gorwal et al. [301] list additional political issues with content removal, such as the opacity of the procedure, that could be handled by making transparent each decision.

2.6.3. STRUCTURAL CHALLENGES

Many of the technical, contextual and semantic challenges identified all along the survey find their causes in the ways research and development on *OCL* have been structured. While structural issues are not changed easily, it is worth enumerating some of them.

Disconnection between machine learning and social science research. While setting up interdisciplinary collaborations is difficult, the survey showed research opportunities for each discipline. For instance, while computer science would benefit introducing contextual information from psychology works in datasets and models, psychology research has not yet studied all variations of *OCL* , and computer science tools could facilitate this work [742].

Disconnection between research and real-world scenarios. Datasets often remain large-grain on the context of *OCL* and on the annotations. However, delving into specific *OCL*, possibly engaging with the communities involved, especially with the authors of *OCL* and their targets, would allow to better understand the requirements that a system should verify. Participatory design, recently raising in ML works [456], while not being the entire solution [768], would benefit the area of *OCL*, and the comprehension of human-aligned requirements. Yet, an obstacle might generally be the stronger interest for algorithmic works than for dataset works in computer science conferences.

Finally, computer science research can benefit from the tradition of social science work that usually begins with the definition of the concepts studied. For instance, psychology researchers who identify the individual and group targets of hate speech point out categories of people with similar socio-demographic attributes (race, religion, disability, sexual orientation, ethnicity, class, gender, behavioral and physical aspects [753], as well as moral [589] and mental status [353]). Clarification as such can help scope the work and avoid conceptual confusions even with disagreement on the definition. Similarly, computer science works on biases and unfairness can benefit from a clear statement about the biases and harms they study. Blodgett et al. [113] provide an extensive review of the study of biases in natural language processing publications, and provide recommendations on that end.

2.7. HARMS BEYOND THE ALGORITHMIC FAIRNESS FRAME

There are many harms due to the integration of machine learning (ML) into digital services that are not captured in the bias / algorithmic fairness identification and mitigation approach to data and design of algorithms that we briefly mentioned in the previous sections, and not necessarily discussed within the context of automatic conflictual language detection systems. In particular, by locating potential harms in datasets and algorithms, bias mitigation approaches fail to capture the impact of ML more broadly on discrimination and social inequities. This is what we outline in the remaining of this section. We argue that to avoid trivializing the problem of harms, researchers should go beyond a focus on abstract concepts like datasets and algorithms as they pertain to decision making. While these are some of the main abstractions computer scientists use in ML research, they do not account fully for the material manifestation of ML in the world. Similarly, framing systems in terms of automated decision making emphasizes a socio-technical view, but leaves out the many ways in which ML is used to produce digital services and that may raise similar concerns around social inequities. In order to go beyond seeing

ML as a technique in decision making, we sketch alternative views on ML that help to highlight its broader impact, with a focus on discriminatory effects and inequities. We propose an ML-pillar and a production view on ML. We hope that these framings can provide directions for the development of more robust research and policy-making to address the potential harms of ML.[8]

2.7.1. THE MACHINE LEARNING-PILLAR VIEW

Setting up systems requires the design and use of a number of entities, especially algorithms, training data, and protected attributes (in case one wants to apply fairness metrics and mitigation methods), etc. These entities are implicitly presented as unquestionable as they are necessary to the functioning of the technology. However, they can also be problematic. ML systems for instance use datasets with various attributes describing individuals (e.g., skills, background, etc.) and target decisions (e.g., granting a loan or not), in order to extract data patterns within these. Implicitly, this assumes that the attributes are relevant to the target decisions, and that new decisions can be made simply by comparing a new individual to individuals in the dataset. These are strong assumptions that might lead to unfairness. Here, we surface such potentially problematic assumptions, that can lead to question the use of ML itself in certain contexts.

DUBIOUS OPTIMIZATION TASK DEFINITION

It is worth taking a step back from the focus on datasets, models, and their biases, and interrogating whether the envisioned task can be performed using ML, whether the labels and data that are put forward are scientifically sound for the task, and whether relevant data can actually be found. ML relies on principles that might not be discussed often enough as they might seem obvious, but that do shape a task in possibly harmful ways. We pinpoint these principles and their issues below.

The principle of reproducing historical data patterns. ML systems performing classification or regression tasks rely on the identification of patterns in training data that reflect past behaviours, in order to learn an inference behaviour. Making inferences by mimicking past behaviours and comparing the new samples that describe the new inference subjects to past training data (generally corresponding to past subjects of a decision) can be harmful in various ways. Are the past behaviours desirable, and is it desirable to simply repeat them? This is something to question in the different contexts of application of ML. Besides, if certain types of populations were not encountered in the past, the systems might make irrelevant inferences for them. If the past behaviours were problematic or discriminatory, the new inferences would reproduce problematic behaviours. For instance, Raghavan et al. [653] question the idea of using ML for job hiring decisions. The ML process would inherently skew the task of identifying satisfactory candidates towards finding candidates resembling those who have already been hired, leaving out new, different, qualified candidates that have not been encountered before

[8] This section is based on one publication [62]. We extracted relevant subsections that explain types of harms that were not mentioned in the previous section. We removed any detail or examples about these types of harms that do not directly contribute to our exposition of the harms.

by the companies. Accounting for the new candidates would require a human to foresee all their characteristics, and to possibly build data items representing them and their desired label. This would be directly opposed to the ML principle of learning patterns in the data and automatically repeating them. Besides, "do we want to make this decision simply by comparing this individual to others?" is a question that is implicitly answered positively when making the choice of using an ML model. Yet, certain notions of justice are not comparative, in which case it is not valid to use ML to make a decision [254].

Scientific soundness of the system's task or objective. How sound is it to learn patterns between the input data available and the target label? ML relies on the assumption that there exists a relation (formalised as a pattern) between the input data and the target label. While this relation might not have to be of causal nature, the existence of correlations is also sometimes questionable [201, 901]: is the existence of correlations backed up by prior scientific evidence? Are we making an assumption that might lead to random and harmful predictions. A burgeoning critique of ML systems has therefore fundamentally been to question the scientific validity of the underlying assumptions and stated objectives of the system, before the examination of the reliability or accuracy of the system [248]. The increasing reliance on pseudo-scientific assumptions for certain systems, including lie detection, emotion detection and biometric categorisations systems necessitates an initial, broader analysis of whether the stated objectives of certain systems are even scientifically valid. However, we also caution that science or academia is not protected from accepting ways of categorizing and ordering populations that are very much based on power, majority consensus, or colonial histories. These may normalize oppressive beliefs as scientifically valid disadvantaging, for example, minorities or racialized others, as has been evident in eugenics and phrenology and the way they make a reappearance in ML [108, 19].

Desirability of the task. Even when the task is sound and its repetitive nature is acceptable, it still remains important to ask whether this task is desirable, i.e., whether the creation of an automated decision system would indeed automate a desirable task, or whether it serves to obscure a questionable one. Let's imagine a system which would equally allocate bad working conditions to different job seekers. While being fair for all its users, it would also be harmful as it would be allocating negative resources [457, 432].

SOUNDNESS OF THE DATA SCHEMA DESIGN

Once a task is agreed upon, the ML setup imposes the creation of a dataset. What does creating a dataset entail for the desired inference task? To what extent do the ways datasets are formalized reflect the real case? Building a dataset requires defining a set of attributes and discretizing the values they can take, and to collect data reflecting such attributes. These activities impact the inferences made by the ML model trained on the data, in ways which can be harmful outside the fairness framing [85, 401].

Problematic definition of attributes. The choice of attributes constituting the dataset impacts how well the model trained on this dataset performs its intended task. An incorrect choice might be harmful. The selected set of attributes might be incomplete, not

providing enough information to properly perform the inference task, such as not providing the amount of a loan one has applied for in order to predict one's likelihood to repay it. The attributes chosen might also not be relevant for the task at hand, such as using the number of siblings one has to predict whether one is likely to repay a loan. Collecting data on certain attributes might even be considered unfair and possibly illegal. This can be because they are not the result of volitional decisions (such as the age or race of an individual) to decide on jail time contrary to potentially volitional decisions like the number of prior offenses; or because they are privacy-infringing [315, 653]. The choice of certain attributes for the models might also prevent certain stakeholders from recourse over inferences, such as when the attributes are immutable, conditionally immutable, or should not be considered actionable [834]. Other attributes do not necessarily have to do with unfairness but with offensiveness [201].

Besides the attributes to train the model, the decision space for a decision-maker refers to the choice of target labels [557]. This choice defines the set of actions or decisions that a decision-maker can take with the help of the corresponding ML model. This choice might greatly impact the environment in which the ML model is implemented. It might reduce the number of possible decisions taken compared to a situation where humans make decisions without any system support. For instance, loan lending systems often decide either to accept or reject a loan application, recidivism prediction systems infer whether someone is likely or not to reoffend in order to decide whether to put or keep one into prison or not. A decision-maker could foresee other possibilities, for example, Mitchell et al argue that a decision maker may consider "a loan with different interest rates and loan terms" [557], one could also consider proposing reinsertion programs for the detainees.

The choice of erroneous data to populate the attributes. While the task could possibly be sound, it might be that the data used in practice are not valid for populating a chosen attribute, for various problematic reasons. Essentially, either the phenomenon the data should reflect is not measurable or only with inaccurate proxies, or a satisfactory proxy might exist, but errors might arise from the way this proxy data is collected.

In terms of proxy, depending on the nature of the proxy, this can raise various harms. If the proxy is too approximative of the real data or not even scientifically related to the phenomenon, then the ML model might learn to perform well solely on these inaccurate data. This is often the case when it is hard or impossible to collect the needed data as the phenomenon is not measurable easily or at all. For instance, the detection of emotional expression is a popular task in ML, that has been performed using different types of proxy data for the true, interior, emotional state of an individual. Yet, some of these proxy data, such as facial expression or heart rate, see their relevance contested following existing research on emotions, as their accuracy and suitability for emotion is limited [790]. Using such proxy for performing an inference task would then lead to prediction errors that might be harmful depending on how the system is used.

The data samples that are included in the dataset might reflect an incomplete view of the world due to limitations in the design of the sampling arising from practical reasons or human biases. For instance, in the recidivism case, only individuals who were actually released and not in jail and then followed over two years could be included in

the datasets, biasing the set of individuals in the training data, as we cannot know accurately what the individuals in jail would have done if they had been released. Mitchell et al. [557] also mention that the human process leading to the inclusion of individuals in the dataset or not may reflect oppressive social structures (e.g., overpolicing of certain minorities). In relation to that, the collected data might be wrong due to similar practical constraints. Historical human decisions might indeed be biased (such as for jail time decisions that judges might have made) and consequently can be considered wrong for certain data samples and inference tasks. In such case, the training data labels collected are flawed from the beginning. Such sampling and errors raise concerns once the dataset becomes the basis for model training and future decision-making, as it takes away from the discussion and normalizes these prior questions.

2.7.2. THE PRODUCTION VIEW

ML is not only a scientific field, but also a business. It is typically not just developed for the sake of creating 'intelligent machines', for some notion of intelligence, or solely for relieving end users of laborious tasks. As a business proposition, ML brings about other considerations that are typically not considered in computer science research. For example, deploying ML systems requires setting up production processes and associated computational infrastructures to collect, process and maintain datasets, as well as to train ML models and deploy them [341]. These pipelines and infrastructures, and their production, not only pose hard engineering problems, but are deeply shaped by the business logic surrounding them. However, the political economic considerations are typically abstracted away, despite the constraints this may pose for the application of fairness mitigation methods. By considering ML as if it exists independent of the business of computing that underlies its deployment, many of the inequalities arising from the production of ML become invisible.

Example of one use-case: chatbot-based services. ML is especially shown to be effective when applied to day-to-day operations of an organization, solving complex resource allocation or logistical problems, or improving production lines in many sectors ranging from manufacturing to creative industries[9]. This means that many applications of ML will take place in Business to Business (B2B) contexts, and not just in consumer facing (B2C) applications. In B2B applications, ML is considered a viable business proposition as long as it provides either greater or new forms of revenue, or cost cuts. To give an example, we look at the use of machine-learning-based chatbots for customer service. Chatbots can be deployed to cut costs by aiding customers in solving their own problems. A successful chatbot is one that can keep customers from contacting a call-center, reducing the cost that can accrue with each call.

In the context of chatbots, a policy approach narrowly focused on algorithmic fairness would aim to provide services to customers from different sub-populations equally, assuming the only harm of interest is that of fairness in market services. However, here, algorithmic fairness leaves out considerable factors driving inequalities between different populations and organizations implicated in production processes. More and more

[9] https://www2.deloitte.com/content/dam/insights/us/articles/4780_State-of-AI-in-the-enterprise/DI_State-of-AI-in-the-enterprise-2nd-ed.pdf

institutions delegate their fundamental operations to scaled-up ML-services and ML service providers, for whom profitability depends on the externalization of costs of contextual needs, failures, or damages, by design, to others. For example, when chatbot services are adopted, costs and harms due to removing human support may be passed onto customers who use the chatbot. Cost-shifting of this nature unfairly burdens particular populations of customers, such as people with disabilities or accessibility requirements. The cost and risks also shift from the ML service providers to the requesters of the chatbot, and create intrinsic dependencies between the requesters and service providers. Apart from cost shifting, there are also labor implications. The use of chatbot services may involve swapping call center jobs with gig workers who train chatbot algorithms with computers purchased at their own expense and who work in their homes, subjecting their household to surveillance8, potentially with even less labor protections than a call center worker [10]. Hence, besides investigating how a chatbot interfaces with end-users (e.g., focusing on matters of data accuracy, safety etc.), there is also great value in examining the way chatbots–or other ML services–transform consumer relations, organizations and labor conditions, or redistributes risks to the weakest parties in their production cycle, such as gig workers doing menial tasks and end-users. The economic pressures on the business of computing therefore also impacts the ability and willingness of tech companies to address their potential societal harms.

HARMS RELATED TO DATA USAGE

Due to market and cost-saving pressures, ML service providers can reduce costs by disregarding privacy concerns or data protection considerations when collecting data samples [457]. Most companies skirt privacy considerations by scraping public data from the Web, such as image datasets [110] and text datasets. This may not always be legal, and even if so, might not be enough to fully address customers' normative expectations of privacy and meaningful consent. Given that multiple data points can be combined from diverse sources, revealing new information about individuals that was hidden from a primary dataset, ML services have the capacity to produce undiscovered privacy concerns [547]. Users usually give their consent for a specific context where they publish the data (often a social media), but they are not aware of the other potential uses [124]. There are also more coercive scenarios in which public data are collected without user consent and then used to develop AI systems. Raji et al. also point out that the methods to collect samples might be dubious, taking the example of a startup which signed an agreement with the government of Zimbabwe to collect face images from its population through various camera infrastructures, without the consent of the population itself [660].

HARMS RELATED TO THE COST OPTIMIZATION OF THE ML LIFECYCLE

ML pipelines go beyond obtaining datasets. They also require complex processes and computational environments for the efficient development, testing and maintenance of the models and the systems they are part of. For any organization that is moving into ML, these are significant costs, often in the form of labor costs or capital expenses

[10] Olivia Solon, Big Tech call center workers face pressure to accept home surveillance, NBC News, https://www.nbcnews.com/tech/tech-news/big-tech-call-center-workers-face-pressure-accept-home-surveillance-n1276227

associated with computing machinery. The optimization of these costs can result in the development of new harms, that we describe further here.

Exploitation of workers. Crowdsourcing is employed in multiple activities of the ML pipelines, such as for data annotation, data filtering and, in some cases, even data collection. Similarly, when users have troubles with workflows managed using ML, organizations turn to low-paid micro-workers to make up for the failings of these systems. Micro-workers, for example, are tasked with content moderation, technical support, customer relations, and responding to consumer contestations. Companies/organizations save money on labor costs in these low-end jobs in part by neglecting to care for workers or address harms to workers. A study across 75 countries with 3500 workers found that despite micro-workers being necessary for the production of ML, "workers and their jobs remain invisible, poorly regulated and paid, seemingly not directly employed by the corporations that construct and run such systems" [95, 394, 561]. Multiple concerns expressed by various human-computer interaction and social science literature revolve around the treatment of micro-workers within crowdsourcing platforms. Crowdsourcing tasks are designed to prompt workers to be very fast at their job, to accept a large number of tasks per day, while they are paid low payout, and depend on this job to make a living [690, 611]. This leads to exploitative behaviours by the annotation requesters. Workers have low flexibility in time organisation [909], they are automatically considered as unreliable if they refuse tasks, and their work is not valued. Besides, the crowd workers not only sell their labor, but companies also require the exploitation of their personal assets (computer, car, bike, rented apartments, etc.), and the data captured is integrated as production data to increase the efficiency of the service providers operations and optimize worker productivity and labor costs, at times to the point of cruelty [404, 838]. Finally, to this day, crowdsourcing platforms' accessibility for disabled workers, elderly people, etc is low [948], and privacy of the crowd workers is often at risk [887].

Exploitation of resources. Besides workers, the production pipelines reinforce the exploitation of resources, as these are fundamental to developing and deploying ML. Most ML systems require large amounts of data and computational power. These systems intensify reliance on fossil fuels[11]. Besides, natural resources are exploited in order to develop the hardware components, energy resources, and infrastructures needed to build and deploy both the data engineering pipelines and the computational infrastructures for ML [327]. The production of compute-heavy systems that depend on cloud and mobile computation reinforces environmental issues in areas of the world where resources are exploited, where data are hosted, and where computations are done [132, 234, 93]. The damage from these range from "water usage, pollution from backup generators, supply chains for the rare earth minerals used in hardware, and the toxic materials involved in the production of this hardware" [12]. The ever-increasing need for energy

[11] Sarah Griffiths, Why you internet use is not as clean as you think? BBB Smart Guide to Climate Change, 6. March 2020. https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think

[12] Ingrid Burrington, The Environmental Toll of a Netflix Binge, The Atlantic, 16. December 2015. https://www.theatlantic.com/technology/archive/2015/12/there-are-no-clean-clouds/420744/

for computing, especially amplifies existing inequalities and climate injustice. Namely, politically, culturally and economically marginalized populations will suffer the consequences of climate change more severely, and they will do so even though they use vastly less fossil fuel–based energy, bear far less responsibility for creating environmental problems, and do not enjoy the benefits of technological innovations [337].

Costs hindering due diligence. The production view also draws attention to engineering and management costs and their relationship to social, political, and economic inequalities. In fact, the costs of ML rise due to the ways in which ML systems function (e.g., use of data in development and deployment) [115], are employed in practice (e.g., centralized model for making predictions over many individuals, personalization of model outputs through fine-tuning of a central model into many individual decentralized models, etc.), and are developed (e.g., need for numerous training iterations and experimentations). All of these cost factors and the complex production line they bring about have a direct impact on the application of unfairness mitigation methods. The cost of ML production is likely to either deter companies from catering to concerns about ML and discrimination, as this would require more computation, or reduce it to injecting a minimal unfairness mitigation method into their pipelines for compliance purposes. The complexity of these pipelines further raises serious concerns about the feasibility of effectively applying fairness mitigation methods across all of these optimization steps, a matter not yet considered even in research.

2.8. CONCLUSION

In this chapter, we took the example of one usage of machine learning (ML) —ML for the automatic detection of online conflictual languages. From this example, we identified a number of harms that flawed outputs (i.e., output failures) of an ML system might cause —these harms especially revolve around questions of discrimination and unfairness. Taking a look at broader literature on ML and harms, we also identified other categories of harms that ML systems might cause. Besides potential unfairness from flawed outputs, using an ML system for a certain task might not be desirable as it might reinforce historical biases or allow for harmful activities; the training dataset schema and its population, used to train the model, might be problematic (e.g., encoding offensive representations of populations); and the production process of the ML system might cause environmental harms, reinforce poor labour conditions, etc.

In the chapter, we also identified causes for these flawed outputs (and resulting harms). At the technical level, they might arise from the (flawed) configuration of the ML algorithm and its training process, or from the (flawed) configuration of the training and test datasets. At a research-community level, it might also be that current research directions in computer science are not acknowledging enough prior works from other research communities, that could be useful to design more appropriate ML systems. This, in turn, might be due to structural issues in the organization of research activities. Having knowledge of certain potential harms and their causes, we can now investigate in the next two chapters the types of solutions proposed by the research community.

3

TECHNICAL APPROACHES FOR DIAGNOSING & MITIGATING ALGORITHMIC (NON-)ROBUSTNESS

3.1. INTRODUCTION

In this chapter, we start investigating one first trend of technical solutions to the algorithmic harms identified in Chapter 2: machine learning (ML) robustness. Indeed, one of the core principles for building trustworthy (ideally hazardless) ML systems [720] is robustness [261], defined as *the insensitivity of a model's performance to miscalculations of its parameters* [583, 921]. Examples like Tesla's Full Self-Driving mechanism erroneously identifying the moon as a yellow traffic light,[1] or Autopilot being fooled by stickers placed on the ground,[2] show that ML systems might not be robust, but be susceptible to errors and vulnerable to external attacks. This may result in undesired behavior, decreased performance [891], and various physical and social harms.

In response to these issues, a growing body of literature focuses on developing and testing robust ML systems. Methodologies towards robust ML have addressed every phase of the ML pipeline, going from data collection and feature extraction, to model training and prediction [891]. Such methodologies have been applied to a wide range of tasks and application areas, including (but not limited to) image classification [802] and object detection [173] in Computer Vision, or text classification in Natural Language Processing [463]. Considering the increasing efforts devoted to this field within trustworthy

[1] https://www.autoweek.com/news/green-cars/a37114603/tesla-fsd-mistakes-moon-for-traffic-light/ (access 13.10.2022)

[2] https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf (access 13.10.2022)

ML, we analyze the progress made so far and give a *structured* overview of the suggested solutions. Furthermore, we also aim at identifying the areas that have received least attention, highlighting research gaps, and projecting into future research directions.

Our work differs from similar efforts in three main ways. (1) As opposed to some previous work [891, 298, 159], we do not limit the scope of our analysis to adversarial attacks. We argue that, as suggested by [236] or [745], natural (i.e., non-adversarial) perturbations constitute a common real-world menace that needs further attention. (2) As far as the application area is concerned, and contrary to surveys solely focusing on tasks like Computer Vision [236] or architectures like Graph Neural Networks [745], we do not limit our survey to any technology in particular. We rather conduct our search in a task-agnostic way. Such an approach helps us identify the most prominent trends within the field and compare the differences in effort and interest across applications as part of our survey. (3) Most importantly, we adopt a human-centered perspective for highlighting the technological challenges and opportunities in the field of robust ML. We argue that previous work, which is predominantly algorithm-centric, fails to identify the potential of human input when crafting robust algorithmic systems. We also emphasize the need to understand current human-led practices in order to integrate robustness into existing workflows and tools. To this end, we advocate for a multidisciplinary approach and bring insights from human-centered fields, such as explainable ML, crowd computing, or human-in-the-loop ML. We, therefore, make the following contributions[3]:

1. We give an overview of the main concepts around robust ML. We consolidate the terminology used in this context, disentangling the meaning and scope of different constructs. We pay special attention to identifying the commonalities and differentiating aspects of the used terms.

2. We systematically summarize 380 papers on robust ML and related concepts and arrange them in three different taxonomies. First, we group papers that improve *robustness by* working on different aspects of the ML pipeline. We identified three main aspects that the selected studies work on: input data, in-model attributes, and model post-processing aspects. Second, we focus on distinct architectures and application areas of robust ML systems and define *robustness for* specific architectures (e.g., Graph Neural Networks), specific tasks (i.e., Natural Language Processing and Cybersecurity), and systems conceived within other fields of Trustworthy ML (i.e., explainable and fairness-aware systems). We focus on these particular architectures, systems, and fields as they have comparatively received little attention in previous surveys despite the importance of robustness as a desired property.[4] Third, we create a taxonomy related to the *assessment* of robust ML systems.

3. We identify and discuss disparate research efforts in each of the established fields and identify research gaps. Specifically, we make a special in-depth analysis of the

[3] This chapter is an extract from one publication [825]. We only retained from this publication the overview of the technical research on robustness, and the limitations identified around this research. We left out the methodology, the main insights on technical methods from the survey, part of the future work discussion, and the conclusion, as they were not necessary for understanding the rest of the chapters in the thesis.

[4] Refer to the full paper for an extensive overview of those robustness metrics and brittleness mitigation methods [824]. For the sake of brevity and storyline, we do not include those in this thesis.

opportunities brought by one of the identified research gaps: the absence of human-centered work in existing methodologies. We highlight the multidisciplinary nature of the robust AI field and provide an outlook for future research directions, bringing insights from human-centered fields.

3.2. OVERVIEW OF THE CONCEPTS SURROUNDING ROBUSTNESS

From our collection of papers, we evinced that the notion of *Robustness* is ill-defined. A number of machine learning sub-domains refer to robustness from different viewpoints. We clarify the relations between these domains in subsection 3.2.1. We also identify that a number of concepts directly related to robustness are used in different ways across research papers (Figure 3.1). We disambiguate the interpretation of related terms in subsection 3.2.2. Finally, our analysis of the papers surfaced a few recurring themes, introduced in subsection 3.2.3, and used to organize our survey.

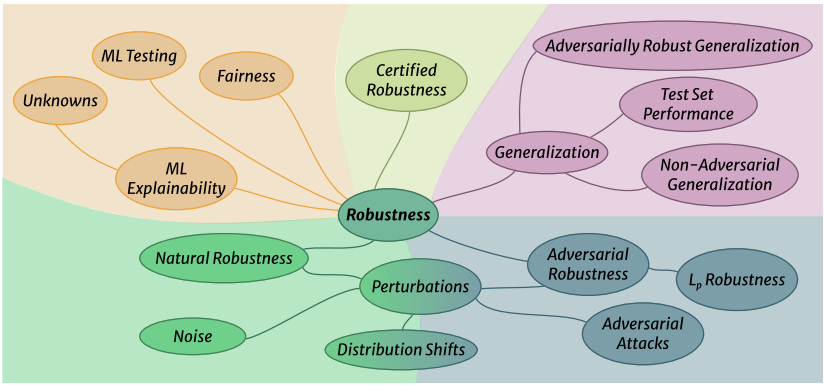


Figure 3.1: Main concepts found through our analysis of the literature on Robust AI.

3.2.1. THE VARIOUS SHADES OF ROBUSTNESS

Given the broadness of the literature on robustness and the variety of contexts in which it is considered, addressed, and analyzed, we discuss and provide a common ground about the definitions of robustness and its associated concepts. Particularly, robustness is generally defined as *the insensitivity of a model's performance to miscalculations of its parameters* [583, 921], with Nobandegani et al. [583] stating that *robust models should be insensitive to inaccuracies of their parameters, with little or no decline in their performance*. Two main robustness branches have been identified: robustness to adversarial attacks or perturbations, and robustness to natural perturbations.

ADVERSARIAL ROBUSTNESS

Adversarial Robustness refers to the ability of models to maintain their performance under potential adversarial attacks and perturbations [940]. Adversarial perturbations are imperceptible, non-random modifications of the input to change a model's prediction, maximizing its error [814]. The result of such a process is called an adversarial example,

i.e., an input x' close to a valid input x according to some distance metric (i.e., similarity), whose outputs are different [160]. Such data is employed to perform adversarial attacks, whose objective is to find any x' according to a given maximum attack distance [173]. The literature presents different classifications of adversarial attacks: targeted and untargeted [172], and white-, grey-, or black-box [562]. Targeted attacks generate adversarial examples misclassified as specific classes, while untargeted attacks generate misclassified samples in general. The main difference between white-, grey-, and black-box attacks is the attacker's knowledge about the model or the defense mechanism.

A similarity metric is often defined when generating attacks or evaluating robustness. Depending on the input domain, different metrics can be applied. These metrics are built as a function of a parameter (usually denoted with the letter p) whose value influences its computation. For example, Carlini et al. [160] define a generic p norm from which different metrics with different meanings are derived. In their case, when $p = 0$ (L_0 distance), the number of coordinates for which the valid and perturbed input are different is measured; when $p = 2$ (L_2 distance), the standard Euclidean distance between the valid and perturbed input is computed; when p = infinite (L_∞ distance), the maximum change to any coordinate is measured. A particular type of robustness is Certified Robustness that guarantees a stable classification for any input within a certain range [188].

NATURAL ROBUSTNESS

Natural Robustness (a.k.a. Robustness against natural perturbations) is the capability of a model to preserve its performance under naturally-induced image corruptions or alterations. [236]. Natural Perturbations (a.k.a. Common Corruptions [351] or Degradations [290]) are introduced through different types of commonly witnessed natural noise [863], e.g., Gaussian noise in low lighting conditions [351], and represent conditions more likely to occur in the real world compared to adversarial perturbations [236]. Temporal Perturbations are natural perturbations that hinder the capability of a model to detect objects in perceptually similar, nearby frames in videos [737]. All these perturbations result in a condition where the distribution of the test set differs from the one of the training set [448]. This condition is typically referred to in the literature with overlapping concepts, namely distribution shift [820, 229], Out-of-Distribution data (OOD) [745, 295], and data outside the training set [634].

GENERALISATION

Generalisation is another widely used term in the robustness literature. In general, it is defined as the model's performance on unseen test scenarios [617] or as the closeness between the population (or test error) to the training error, even when minimising the training error [580]. Two other types of generalization are also reported: adversarially robust [919] and non-adversarial generalizations [881, 937, 634, 295]. While the first one refers to the capability of a model to achieve high performance on novel adversarial samples, the second one is evaluated on non-adversarial samples (e.g., natural perturbations [881, 937], distribution shifts [634, 295], etc.).

PERFORMANCE

Across the inspected literature, the term performance is employed with a broad variety of meanings. Depending on the aspect of interest, it may refer to accuracy [236], robustness [472], runtime [758], or precision [905]. Given such variety, the actual meaning of performance will be addressed only when relevant to understand the concepts explained in the core survey.

3.2.2. DOMAINS ADJACENT TO ROBUSTNESS

Machine learning (ML) explainability, fairness, trustworthiness, and testing, are four research domains recurring across robustness literature. While there is no agreed upon definition of each of these fields and their goals, and we acknowledge it is not possible and desirable in the scope of this survey to provide a complete overview of these fields, we provide here explanations that are sufficient to understand the relation these fields bear to robustness.

EXPLAINABILITY

ML explainability is the field interested in developing post-hoc (explainability) methods and (inherently explainable) models that allow the internal functioning of ML systems to be understandable to humans [162]. We identify three types of relations between the explainability and robustness fields. A number of papers investigates how explainability methods can be used in order to *enhance the robustness* of models. Another set of papers investigates *how robust existing explanability methods are* to various types of perturbations. A last set of papers instead studies how existing methods for enhancing robustness *trade off* with the explainability of the models, and especially with the alignment between the model features, and the features a human would expect the model to learn.

We also consider the field of *(un)known unknowns* [532] close to robustness, as they are typically caused by OOD samples. In this field, methods to identify and mitigate the presence of such unknowns are developed and, while these methods typically fall within explainability [840, 741], they are directly applicable to increase the robustness of a model.

FAIRNESS

ML fairness in the broad sense is the field interested in making the outputs of an ML model non-harmful to the humans who are subject to the decisions made based on these outputs. Researchers in this field have developed a number of fairness metrics [848] and methods for mitigating unfairness [546]. We identify two types of relations between this field and robustness, similar to the relations between explainability and robustness: *robustness of fairness metrics and methods* to different types of natural and adversarial perturbations and *trade-offs* caused by the application of robustness methods.

TESTING

ML testing [924] is a field emanated from software testing. It consists in developing methods and tools to identify and characterize any discrepancy between the expected and actual behavior of a ML model. While this field bears a broader scope since brittleness to different perturbations represents one of the many types of unexpected behavior

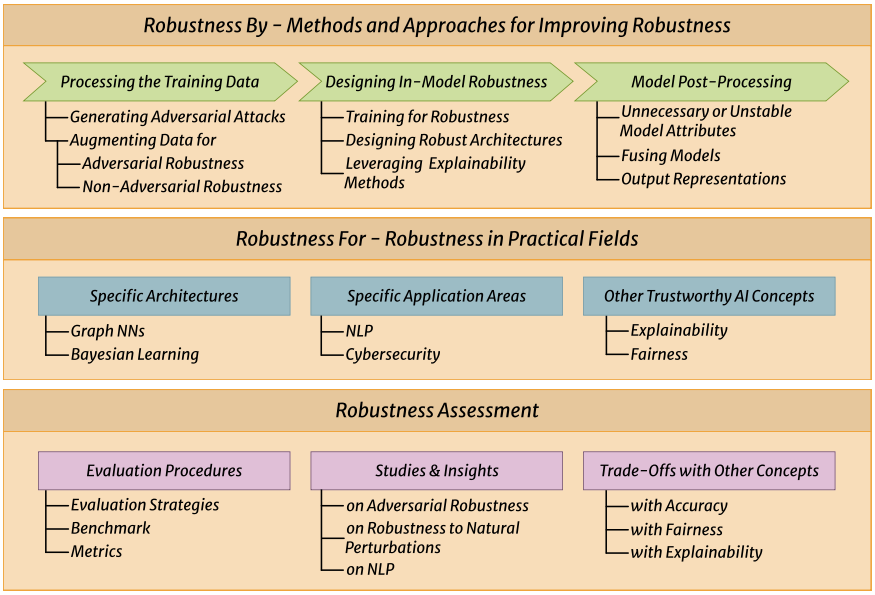


Figure 3.2: The three themes and their sub-categories that shape our survey.

of a model, it is also narrow as it is solely interested in detecting the issue, but not its mitigation. Naturally, methods developed in this field could potentially be adapted in the future to better detect robustness-related issues.

3.2.3. Themes in Relation to These Robustness Shades

Analyzing the collected publications through a thematic analysis approach [127], we iteratively and collaboratively identified three primary themes and three recurring categories within each of these themes (nine categories in total) that were deemed worth emphasizing (summarized in Figure 3.2).

Robustness Methods

The most studied methods to achieve robustness are categorized according to the stage of the ML pipeline to which they apply, that is either the processing of the training dataset, the model creation stage, or the post-processing of the trained model. Within each of these stages, the approaches vary across publications, and were further clustered into groups based on types of robustness (e.g., adversarial or natural perturbations), and specific ML component (e.g., training procedure or model architecture) they apply to. For each of the groups, we further delved into sub-groups based on the types of transformation applied to the component (e.g., different loss functions or regularizers), and investigated the main similarities and differences across transformations, e.g., in terms of technical approach and performance.

ROBUSTNESS IN PRACTICAL FIELDS

While a majority of papers concentrate their studies and the evaluation of their robustness methods around computer vision or do not mention a specific field, we also identify a consequent number of papers that bear different focuses. We separated these papers from the ones discussed above, because they present particularities that are worth investigating. We categorized these papers broadly based on their research fields. Within each of the categories, we investigated the most researched sub-types for which we retrieved the most literature. Particularly, we identified focuses relating to specific model types (Graph Neural Networks and Bayesian Learning), specific application areas (Natural Language Processing, and Cybersecurity), and specific concepts within the trustworthy AI domain (explainability and fairness). The latter is particularly interesting because it differs from other works in its objectives. Contrary to all other papers which investigate model performance under perturbations, it instead investigates evolution of the fairness and explanations of a model under the effect of perturbations.

ROBUSTNESS ASSESSMENT

The last theme we identified revolves around the assessment of the robustness of a system. Particularly, the importance of developing procedures (methodologies, benchmarks, and metrics) to evaluate robustness emerged from the papers and these procedures revealed to vary greatly across publications (be it publications whose primary contribution is an evaluation procedure, or a robustness method that requires to be evaluated through a defined procedure). We also identified a set of publications whose primary objective is to perform studies to evaluate existing robustness methods and collect insights to further characterize in which conditions each type of method performs best. Finally, the last recurring theme was trade-offs, as many papers that propose or evaluate robustness methods tackle trade-offs while striving to achieve other objectives, be it the model performance or the other trustworthy ML concepts identified earlier.

3.3. LIMITATIONS: INVOLVING HUMAN WORKERS FOR MORE INTERPRETABLE ROBUSTNESS

The survey showed a number of ways where human agents (often crowd workers) might be necessary or meaningful to involve towards making machine learning models more robust. Yet, the survey also showed a lack of investigation in their modes of involvement. We discuss next avenues for future work in that sense.

3.3.1. DEEPENING THE RESEARCH ON HUMAN INVOLVEMENT FOR EXISTING ROBUSTNESS METHODS

A number of papers we surveyed implicitly involve humans to instantiate the methods they propose, either to assess or enhance a model's robustness. Yet, they do not delve deeper into the challenges for a human agent to perform their task, which constitutes an obstacle to the development of methods and frameworks for overcoming these challenges. This merits further investigation as such human involvement is essential to the success of the methods. Especially, we identify two main areas where human involvement is necessary but lacks research.

EVALUATING ROBUSTNESS

To design appropriate perturbations or attacks on which a model should be robust, one often needs human knowledge. For instance, [410] and [462] generate adversarial attacks on text samples, that have to verify a number of human-defined constraints for them to be deemed realistic by humans. Yet, designing such constraints and empirically evaluating (through user studies) to what extent the samples transformed by the corresponding constrained attack align with the human idea of "realistic" sample, has not been investigated extensively, despite how crucial that is for engineering "good" attacks.

In a similar fashion, works on robustness to natural perturbations should ideally define a comprehensive set of domain-specific perturbations relevant to the problem at hand and its context. However, to the best of our knowledge, existing works that develop benchmarks or robustness-enhancing methods [448, 351] with regard to such perturbations have not investigated ways to be more comprehensive. While we believe in the impossibility to reach comprehensiveness (previously unheard-of perturbations can always arise), one could develop tools to support the definition of relevant perturbations. For instance, we envision the usefulness of fine-grained, actionable taxonomies of perturbations (e.g., Koh et al. [448] talk about subpopulation shifts and domain generalization, but this might vary in different domains and types of tasks); collaborative documentation of domain-specific perturbations; libraries to generate such perturbations semi-automatically; and frameworks and metrics to uncover new types of perturbations in the wild, potentially involving humans in the runtime.

INCREASING ROBUSTNESS

Various methods that aim at increasing robustness implicitly employ humans, without extensive focus. [410], for instance, collect potential adversarial examples by executing a sequence of engineered steps, that could be refined by the practitioner who would leverage existing tools for, e.g., identifying synonyms and antonyms, ranking word importance, etc. [634], [165], [574], and [581] respectively show that one can train more robust models by leveraging human uncertainty on sample labels instead of using reconciled binary labels, by integrating human rationales for the labeling process into the training process, or by actively querying the most relevant levels of perturbations from an expert during training. While these are promising research directions, these works could further be improved by exploiting existing works on human computation assessing the quality of crowdsourced outputs [391], or designing crowdsourcing tasks that remove task ambiguity and lead to higher quality outputs [259], especially in the context of subjective tasks. This could serve to understand the nature of uncertainties and define rationales that are relevant to robustness.

3.3.2. INVOLVING HUMANS IN OTHER PHASES OF THE ML LIFECYCLE

Broader ML literature has proposed other approaches to involve humans and make "better" models. Yet, none of these approaches has considered making the models more robust. Instead, they focus on increasing the performance of the model on the test set. Hence, we suggest to investigate how to adapt such approaches to increase model robustness.

ML WITH A REJECT OPTION

While ML models typically make predictions for all input samples, this might not be reasonable and turn dangerous in high-stake domains, when the predictions are likely to be incorrect. Accordingly, a number of research works have developed methods to learn when to appropriately reject a prediction, and defer the decision about the sample to a human agent [349]. Proposed rejectors can either be *separate rejectors* placed before the predictor, that select the input samples to input to this predictor; *dependent rejectors* placed after the predictor and re-using its information (e.g., confidence metrics) to decide which predictions not to account for; and *integrated rejectors* that are combined to the predictor, by treating the rejection option as an additional label to the ones to predict. Each type of rejector bears advantages and disadvantages based on the context of the decision, and would merit being adapted to robustness, as we only found few works towards that direction [608, 426, 801].

HUMAN-IN-THE-LOOP ML PIPELINES

Human-in-the-Loop (HIL) ML [843] is traditionally concerned with developing learning frameworks that account for the noisy crowd labels [672], or "learning from crowds", through models of the annotation process (e.g., task difficulty, task subjectivity, expertise, etc.). Such frameworks often rely on active learning to reduce annotation cost [897, 900]. Recent works around HIL ML also devise new approaches to build better model pipelines by involving the crowd, such as to identify weak components of a system [590], to identify noise and biases in the training data [899, 377], or to propose potential data-based explanations to wrong predictions [149]. While we could find a few works that investigate the intersection between active learning and adversarial training [552, 751, 764, 551], we could not find any work that looks more broadly at the different types of robustness, and the different ways of bringing humans in the ML pipeline. These intersections are yet promising as they constitute more realistic scenarios of the development of ML systems and they succeeded in making models more accurate in the past.

3.4. LIMITATIONS: A CONSPICUOUS ABSENT FROM THE LITERATURE, THE ML DEVELOPER

Our rigorous survey revealed one prominent research gap: the absence of human-centered work in proposed approaches, and the lack of technologies and workflows to support ML developers in handling robustness. In this section, we discuss relevant research literature, and future research directions regarding this topic.

3.4.1. ROBUSTNESS BY HUMAN-KNOWLEDGE DIAGNOSIS

One of the most notable absentee from the retrieved papers is robustness by human-based diagnosis. Existing works focus on generating out-of-distribution data, in order to make a model fail, and later expose this model to this data during training to make it more robust. Especially for robustness to natural perturbations, this means that one should always characterize the type of data the model might encounter before being able to generate such data. This is not always possible in practice, e.g., due to contractual and privacy reasons, cost, temporal variability of contextual application of the model, etc.

To circumvent this issue, a major, promising research direction surfaces from comparing the surveyed robustness methods to existing works in other computer science fields. This direction revolves around developing complementary, hybrid human-machine approaches, that would leverage research progress in human-centered fields, essentially explainability, crowdsourcing and human-in-the-loop machine learning (ML), as well as knowledge-based systems, to estimate model performance on more realistic data distributions without requiring such distributions.

EXISTING APPROACHES

Only few related works leverage human capabilities to identify and mitigate potential failures of a model. In particular, explanations for datasets [716] have been proposed, that could be leveraged by a practitioner to identify data skews that might impact the model performance. In this vein, Liu et al. [506] introduce a hybrid approach to identify unknown unknowns, where humans first identify and describe patterns in a small set of unknown unknowns, and then classifiers are learned to recognize these patterns automatically in new samples. Departing from datasets, Stacey et al. [787], and [41] have trained models whose features are better aligned with human reasoning (with the assumption that alignment leads to stronger robustness), by leveraging human explanations of the right answer to the inference task and controlling the features learned by the model during training to align with these human explanations.

ENVISIONED RESEARCH OPPORTUNITY

The above approaches reveal that instead of looking solely at the outputs of a model and its confidence in its predictions, one can leverage additional information such as the model features or training dataset, to estimate the model's robustness. Especially, even when a model prediction is correct, the model features might not be meaningful. Hence, assessing model features and their human-alignment can allow to shift from solely evaluating the correctness of the predictions on the available test, to indirectly assessing the robustness of the model to OOD data points. Moreover, understanding characteristics of the datasets that led to such learned features could later on serve to mitigate unaligned features.

Surfacing Model Features using Research on Explainability and Human Computation. To surface a model's features, one can rely on a plethora of explainability methods [716]. Certain models are built with the idea of being explainable by design [928, 809], while others are applied post-hoc interpretability methods [677, 800, 69], with different properties (e.g., different nature of explanations being correlation or causation -based, different scopes be it local or global, different mediums be it visual or textual, etc.) [773, 497]. It is now important to adapt such feature explanations to allow for checking their alignment with human expected features.

In that regard, the push towards human-centered explanations for ML developers is highly relevant. Existing explanations often leave space for many different human interpretations, for which the developers do not always have domain expertise to disambiguate the highest-fidelity features. For instance methods that output saliency maps [754] or image patches [437, 289] do not pinpoint to the actual human-interpretable features the model has learned. Yet, one might need clear human concepts to reason over

the alignment of the features [67]. Hence, further research on *semantic, concept-based explanations* acquired via human computation is needed [359, 69].

Leveraging Literature on Knowledge Acquisition for Identifying Expected Features. To reason over feature alignment, one also needs to develop an understanding of the model expected features. While very few works have looked into this problem [741], existing works on commonsense-knowledge acquisition [917] could be leveraged to that end. These works propose to harvest knowledge automatically from existing resources such as text libraries, or through the involvement of human agents, e.g., through efficient and low-cost interactions within Games with a Purpose [64, 854, 683], or other types of carefully designed crowdsourcing tasks [379, 722]. One would need to investigate how to adapt such approaches to collect relevant knowledge, and how to represent this knowledge into relevant feature-based information.

Comparing Features via Reasoning Frameworks and Interactive Tools. Finally, developers need tools to check the alignment between the model and expected features. Interactive frameworks and user interfaces [67], e.g., *Shared Interest* [116], take a step in that direction as they enable manual exploration of model features, with various degrees of automation for comparing to expected features. Inspired by the literature on AI diagnosis, such as abductive reasoning [191, 680], automated feature-reasoning methods could also fasten the process while making it more reliable.

3.4.2. SUPPORTING ML DEVELOPERS IN HANDLING ROBUSTNESS

Looking beyond the research world towards the practice, it is always an ML practitioner who builds the ML system. Hence, it is not sufficient to develop methods that can work in theory, but it is also important to understand the obstacles developers actually encounter in making their systems robust. While studying the gap between research and practice has revealed highly insightful in the past for various ML contexts [365, 498, 371, 454, 637], to the best of our knowledge, it has not been studied in the context of ML robustness. Possibly the closest work is the interview study of Shankar et al. [735] that investigated MLOps practices beyond the development of a model towards production and monitoring of data shifts or attacks.

UNDERSTANDING PRACTICES AROUND ROBUSTNESS

The human-computer interaction community (HCI) has performed qualitative, empirical, studies, typically based on semi-structured interviews with ML developers, to understand how these developers build ML models with certain considerations in mind. These considerations revolve around the different steps developers take, e.g., challenges of collaboration for each step [454, 637], and the use in certain of these steps of tools such as explainability methods [365, 498, 371] or fairness toolkits [679, 481]. These studies have resulted in frameworks modeling the practitioner's process, lists of challenges, and discussions around the fit of existing methods and tools to answer these challenges. We argue that adopting similar research questions and methodologies (e.g., semi-structured interviews with hypothetical scenarios or practitioner's own tasks, ethnographies, etc.) would also reveal useful to better direct robustness research in the future. For instance,

Liao et al. [498] have constituted an explainability question bank that highlights the questions developers ask when building a model by exploiting explainability, and that can serve to identify research opportunities through questions still difficult to answer. A robustness question bank would similarly provide a structured understanding of what is still lacking. Moreover, HCI research investigating practices around ML fairness [220] has shown a major gap in terms of guidance for developers to choose appropriate fairness metrics and mitigation methods. Acknowledging the plethora of robustness metrics and methods, we envision that user-studies around robustness would reveal a similar gap, that could be filled by taking inspiration from the fairness literature.

INTEGRATING ROBUSTNESS INTO EXISTING WORKFLOWS

Some works have also focused on developing workflows and tools to support developers in model building. These works often revolve around user interfaces to more easily investigate a model and its training dataset, and identify failures or bugs [578, 67]. Other works build tools, e.g., documentation or checklists, [125, 31, 556, 283] and workflows [770] to support making and documenting relevant choices when building or evaluating a model. We argue that robustness research should not only focus on algorithmic evaluation and improvement, but also aim at developing new supportive tools and integrating them into existing solutions. In relation to that, and possibly closest to supporting developers in handling robustness, [744] propose the idea of establishing trust contracts, i.e., contract data distributions and tasks that define the type of task and data that is in- and out-of-distribution. Yet, this remains challenging as there is no appropriate way to formalize such contracts.

3.5. CONCLUSION

Machine learning non-robustness (or brittleness) is one of the main issues a machine learning system might suffer from, and that might cause harms once the system is deployed. Hence, in this chapter, we reviewed the technical methods towards measuring machine learning robustness and mitigating potential brittleness. We also investigated limitations of these methods and avenues for future works to overcome these limitations. Next to a set of technical approaches that could be developed towards more robustness, we identified that humans are often not involved in the works. However, they could be involved in different ways, as annotators to bring more interpretability to proposed techniques, and as practitioners that might use the techniques. We especially identified one important research gap that testifies of the research/practice disconnect: we do not know until now how machine learning developers handle the task of making a model robust, what challenges they face, and what their main needs are. We will investigate and answer these questions in Part II of the thesis. Beforehand, we first investigate in Chapter 4 the second research trend for achieving less hazardous ML systems, ML fairness.

4

TECHNICAL APPROACHES FOR DIAGNOSING & MITIGATING ALGORITHMIC UNFAIRNESS

4.1. INTRODUCTION

In this third chapter of Part I, we focus on the second technical research trend for tackling hazards of machine learning (ML) systems: ML (un)fairness caused by ML systems, and arising from uncontrolled biases injected in these systems. For instance, the Amazon screening system exhibited an unfair gender bias, while the COMPAS system was accused of being racist [161]. ML-based systems have a data management component and a data analytic component, which typically utilizes ML models. One of the main sources of the unfairness of such systems lies in biases within the data on which the models are trained [329]. The ML model of the COMPAS system might have been trained on a dataset imbalanced with respect to a protected attribute such as race, and hence the decision model trained on it makes more errors for the underrepresented minority class. The Amazon system might have been trained on a dataset of previous hiring decisions where men have a higher chance of receiving positive decisions, and thus the decision model also exhibits a skewed distribution towards men. These biases are often not detected unless a deployed system behaves unfairly towards a subgroup of the population.

Works stemming from the ML and data mining communities have started to tackle unfairness from certain angles like evaluating the outputs of trained models [946]; and mitigating unfairness by post-processing the outputs of the system [336, 195, 158], or modifying the training process of the inference algorithms [445, 804, 97, 287, 757, 195, 417, 631], or pre-processing the training data [516, 926, 255, 331, 332]. Nonetheless, most of these approaches do not focus on the root cause of unfair systems –uncontrolled biases in the training data– but on the data analytics aspects. Furthermore, it is pointed out that they are not easily accessible and applicable by developers to real-life cases [369, 783]. We believe that more extensive works on bias should be undertaken by the data

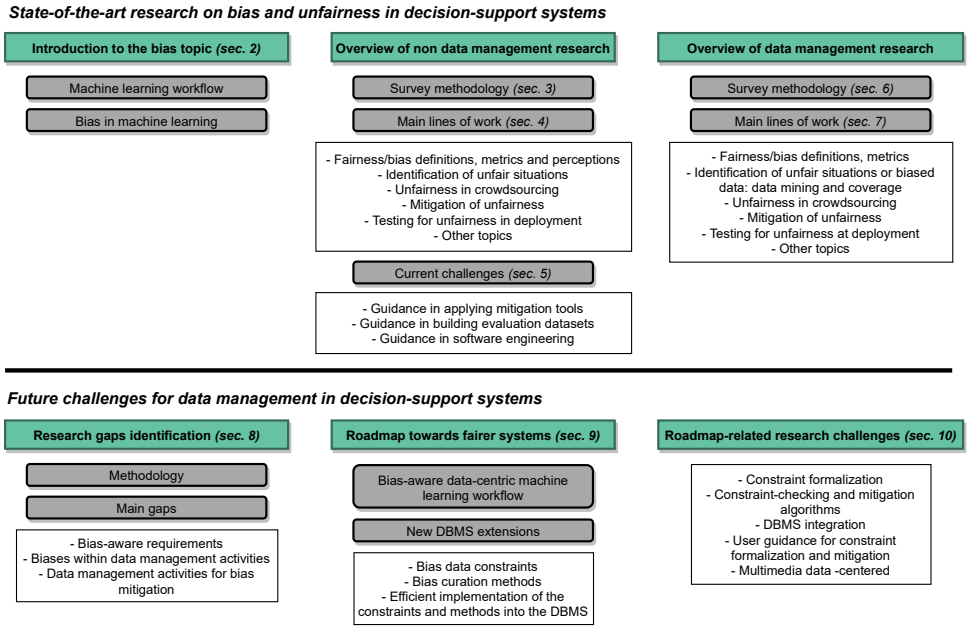


Figure 4.1: Overview of the chapter structure. After performing a survey of the state-of-the-art in various communities tackling issues of fairness and bias in some relation to machine learning, we identify research gaps and propose a set of research challenges for the data management communities.

management community, and this paper highlights the research gaps towards that goal.

To do so, we survey data management and other computer science literature on fairness separately. For this, we highlight and discuss: 1) quantitative overview of the research, 2) research topics, 3) methods and their limitations. We continue with a gap analysis that outlines issues and possible solution spaces to tackle unfairness from a data-management perspective, arguing that bias and unfairness should be a central topic in data management. Additionally, we propose a novel approach addressing several of these gaps by introducing requirements-driven bias and fairness constraints into database management systems. In Figure 4.1, we summarize in details the steps that we take.

With this survey[1], we aim to foster the interest of the data management community in unfairness in ML systems by presenting state-of-the-art literature in various fields. We also identify gaps in current data management research which, if addressed, should bring systems closer to a fair state. We discuss those gaps and provide directions for future data management work. In summary, we make the following contributions:

[1] This chapter is based on two publications. From the first one [66], we only retained the descriptions of technical works to handle harms related to fairness, and the descriptions of limitations in these works. We removed sections that describe the problem further, as this was done in previous chapters. We also removed extensive descriptions of certain technical works, in order to solely keep a level of detail sufficient to understand the rest of the thesis. Finally, we removed suggestions for future work that do not deal with harms caused by the outputs of an ML system. From the second publication [62], we only retained discussions around the most important limitations of the algorithmic fairness paradigm, and a few examples to illustrate these limitations.

- We outline the state-of-the-art of computer science domains actively working on bias and algorithmic unfairness (section 4.3, section 4.4).
- We systematically survey existing research on bias and unfairness issues related to data management (section 4.6)
- We identify bias and unfairness-related research gaps (section 4.7) in data management, and propose new research directions and challenges (section 4.8).
- We outline the conceptual limitations of the proposed algorithmic unfairness works surveyed above (section 4.9), that a developer should acknowledge when developing a model with considerations of harms in mind.

4.2. DATA ANALYTICS: METHODOLOGY

In this section, we explain how we proceeded to the survey of research on bias and unfairness outside data management, research that mainly focuses on the data analytics aspects of data-driven decision-support systems.

4.2.1. METHODOLOGY FOR THE SELECTION OF PAPERS

Our survey is based on a list of the different computer science domains that we consider to be working on topics related to the unfairness of decision-support systems, either because they use such systems, or because they have parts of such systems as an object of their research. This list is the following: machine learning, data mining, computer vision, natural language processing, recommender systems, computer-human interaction, human computation, software engineering, data management, and the interdisciplinary FAT (Fairness, Accountability, Transparency) conferences (i.e., FAT* and AIES). For each of these domains, we retrieved papers of the main conferences (e.g., NeurIPS, KDD, CVPR, ACL, CHI, HCOMP) related to unfairness using two search engines (Google Scholar and DBLP). The approach to this was two-fold: 1) using unfairness-related keywords and the name of the domain, 2) using unfairness-related keywords and restricting the search to a list of the main venues of each domain. The list of keywords can be found in section 4.6. We reviewed the retrieved research papers from the different domains, compiled a list of major research topics currently addressed, and identified the main solutions proposed and their limitations. In this section, we do not cite all of the papers but only a selection of popular ones as there would be too many publications.

4.2.2. GENERAL OVERVIEW

The literature on bias within data-driven decision support systems spans a wide range of topics. The applications of these systems are diverse. These can be to support making decisions about individuals (e.g., deciding whether an offender's jail sentence should be extended based on its likelihood to recidivism, deciding whether to give a loan to someone based on their likelihood to reimburse it, etc.). In these cases, the systems are often trained on structured data about the individuals to make a decision on (e.g., data about the number of previous reimbursed loans, data about the number of crimes the offender previously committed, demographic data, etc.), but also sometimes on image or text data (e.g., deciding whether someone should get a treatment based on the description

of their symptoms, deciding whether a scene is violent and police should be sent based on an image of the scene). It can also be to provide new knowledge for a later decision on someone or something, generally based on images (e.g., classifying whether someone is a doctor or a nurse based on their picture) or text (e.g., deciding whether a sentence is toxic). In the next section, when it is not mentioned, we report works that mostly tackle applications using structured data, as research on unfairness for other types of data is more recent, and hence not all research outcomes are directly applicable to such data.

4.2.3. MAIN RESEARCH DIRECTIONS

From our analysis of literature, we identified six main directions of research on unfairness and bias, which generally correspond to the perspective that different research communities have on the issue. While research starts with both the machine learning and data mining communities to define, formalize and measure unfairness, it then splits into two main directions –even though certain approaches are overlapping–: either identifying cases on unfairness in datasets, or developing ways to mitigate the unfairness when such datasets are used jointly with machine learning techniques for data analytics.

Stemming from the software engineering community and its recent interest in machine-learning-based systems, testing unfairness in the outputs of software is another developing direction. Finally, the human-computer interaction and the crowdsourcing communities started as well to develop an interest in the topic, respectively in understanding how humans perceive the unfairness of data-driven decision-support systems, and in investigating how humans might create certain of the biases that are found in the outputs of the systems.

As no other research community was identified with other research directions relevant to any case of data-driven decision-support systems, that is following these six directions that we organize our survey. In the last subsection, we mention other works that have not been widely adopted by computer science research yet.

4.3. DATA ANALYTICS: STATE OF THE ART

The goal of this section is to provide an overview of the current research topics and related state-of-the-art in the general computer science literature on bias and unfairness. We perform this survey through the lens of decision-support systems where bias and unfairness problems are currently most prevalent, i.e.„ where decisions suggested by the systems can be perceived as unfair or discriminating by certain stakeholders.

This section will serve as a foundation for our survey into bias in data management introduced in section 4.6, where we map the topics found in general computer science literature to the common data management workflow of most decision-support systems to identify research gaps.

4.3.1. DEFINITIONS AND METRICS

Most works first propose *definitions* and *metrics* to quantify unfair situations, often based on definitions of discrimination in law[2].

[2] A survey and comparison of these definitions is in Zliobaite [946].

OVERVIEW

The mathematical definitions vary depending on the type of decision-support system: classification, ranking, regression, recommendation, etc.; but also based on underlying fairness notions like group fairness, individual fairness, or causal fairness [848]. Recently, new notions of fairness (e.g., multi-sided fairness [141]) involving more than one type of stakeholder and protected group were proposed for recommender systems: recommendations could be fair not only for the clients but also for the reviewers or providers of a service [471], or also for items presented in the system [406, 422, 776, 941].

New fairness notions could be identified from social sciences in order to make the systems more aligned with actual fairness values. Many of the proposed fairness definitions and metrics have multiple limitations [383]. For instance, group fairness does not account for unfairness within a given group and hence individual fairness was later proposed by Dwork et al. [241]. The fairness definitions are mostly based on equality notions of fairness but others might be more relevant for certain use-cases (e.g., affirmative actions [566], equity, need [285]). Besides, the identification of unfair situations through causality is also exploited by Madras et al. [524]. Indeed, most definitions rely on notions of correlations and not causation, whereas the ultimate goal of the systems and the metrics is to support making decisions ideally based on causal arguments.

FAIRNESS METRICS

All definitions and metrics assume the preliminary definition of a protected and a non-protected group of records (usually each record refers to a different individual) defined over the values of one or multiple sensitive attributes (also called protected attributes). For instance, in the aforementioned bank example, each record would represent a client of the bank with the attributes representing the information about this client. A sensitive attribute could be the gender, nationality, or age of the client. A protected group could be defined as all the clients whose age is between 15 and 25 years old, or as all the female clients whose age is in this interval. In the rest of this section, for the sake of clarity, we will take as a non-protected group the male clients, and as a protected group any other client. Most existing metrics only handle having one protected group and the rest of the records being aggregated into the non-protected group.

The definitions and metrics also require knowing the label the classifier predicted for each record (e.g., a positive prediction when a loan is granted and a negative prediction otherwise). Most definitions rely on the comparison of statistical measures, and more specifically on checking equality of multiple probabilities, while the unfairness is quantified either by computing the difference or ratio of these probabilities. The definitions and metrics differ in the underlying values of fairness that they reflect, and on the exact measures and information required to compute them.

CONFLICTING PERCEPTIONS OF FAIRNESS

While there exists all these mathematical fairness definitions and metrics, they tend to be conflicting and it is impossible to comply with all of them simultaneously, as shown by Chouldechova et al. [182]. Consequently, few papers [883, 483, 482, 81, 305] study how the fairness of data-driven decision-support systems is perceived in order to choose the most relevant definitions taking into account stakeholders' preferences and mathematical trade-offs. Srivastava et al. [786] show that one simple definition of fairness (demo-

graphic parity) solely matches the expectations of users of hypothetical systems. Conversely, Lee et al. [483, 482] and Grappiolo et al. [305] show that different stakeholders might value different and possibly multiple notions of fairness (e.g., efficient, egalitarian, or equalitarian allocations).

Biases of the end-users of the systems are also investigated since their decisions informed by the predictions impact the (un)fairness of the systems. For example, Zhang et al., Solomon et al. and Peng et al. [932, 774, 630] study how cognitive biases of the systems' users influence how they use the outputs of the systems to make the final decision. Peng et al. [630] show in the context of candidate hiring that the final human decision might be gender-biased by the proportion of male/female candidates exhibited by the algorithm.

4.3.2. IDENTIFICATION OF BIAS AND UNFAIRNESS

DATA MINING RESEARCH

Many data mining papers, dating from 2008 to 2016, deal with discovering and measuring discrimination within datasets, the results being potentially useful for "debugging" the datasets for later training machine learning models. They investigate scenarios of direct and indirect discrimination, further complicated by additional privacy concerns [695] and cases where the protected attributes are unavailable.

Methods. At first, methods relied on learning rules based on the dataset features potentially used for making the decisions, and on identifying features leading to discrimination [627, 696]. Later, situation testing was used to account for justified differences in decisions concerning individuals from different protected groups [516]. "Unlike hypothesis testing, where a statistical analysis is adopted to confirm a predetermined hypothesis of discrimination, the aim of discrimination discovery is to unveil contexts of possible discrimination." [685]. Certain papers combine data mining methods with additional statistical testing in order to verify the potential discrimination situations discovered [698].

Example. In our bank example, rules would be mined from the available dataset with the target label as consequent and other dataset attributes as antecedent.

A rule would be potentially discriminatory with direct discrimination if the antecedent contains one or more protected attributes. Actual direct discrimination would then be verified by setting a threshold α, and comparing it to the difference of rule confidence, for rules with and without the protected attributes –if the difference exceeds α, that would mean that the protected attributes have a strong effect on the rule and hence there is direct discrimination.

Let's use the following highly simplified rules for the sake of giving an example: (*permanent job, low amount loan* → *medium risk not to repay*, confidence 0.1) and (*permanent job, low amount loan, woman* → *medium risk not to repay*, confidence 0.6). If the difference between the two confidences (here $\alpha = 6$) is deemed important with regard to discrimination, then the second rule would be deemed directly discriminating: for instance if $\alpha = 3$, then it is not discriminatory, while with $\alpha = 7$, it is.

As for indirect discrimination, it manifests in certain cases when a rule is not potentially discriminatory as its antecedents do not contain a protected attribute. If background knowledge is available about the context of the data, and protected attributes are shown to be connected to the antecedents within this knowledge, then the rule might be indirectly discriminating.

An example of such would be if a rule such as *permanent job, low amount loan, district1234 → medium risk not to repay* was found with high confidence, and from prior human knowledge, we would also know that the rule *district1234 → Black community* holds with high confidence. Then, proposed algorithms could estimate the confidence of the rule *permanent job, low amount loan, district1234, Black community → medium risk not to repay*, and identify it as discriminatory.

RESEARCH ON MULTIMEDIA APPLICATIONS

Natural language processing. Natural language processing (NLP) [807] focuses on social, undesired biases usually related to gender or race. For example, text completion models are shown to perform better on text from majority languages such as Standard-American English than on text from socially-restricted dialects such as African-American English. These works usually identify undesired biases from their knowledge around the context of the application, and propose methods to quantify these biases, often through the use of semi-synthetic datasets.

Computer vision. On the contrary, in computer vision, most papers tackle systematic dataset biases that are not necessarily related to human values but to properties of the world, such as image extrinsic properties like illumination [539, 895] or image quality [765], or intrinsic properties like the background when classifying the sentiment of a picture [607] or the actions represented in images [494], or properties of the object to detect such as face orientation [449], or object scale in scene recognition [356].

Some works however investigate the diversity of the samples with regard to their cultural provenance for object detection tasks [736] or to protected attributes (e.g., gender bias in text for image captioning [348]). For instance, facial recognition models were shown to be trained on datasets which do not necessarily reflect the diversity of the populations on which the models are applied to, leading to an imbalance of accuracy for the different populations [793, 138]. It is shown that these bias issues impact the performance and generalization of the trained models to new samples [435, 828].

4.3.3. MITIGATION OF BIAS AND UNFAIRNESS

WORKS DEALING WITH TABULAR DATA

Mitigation methods decrease the unwanted biases in the outputs of the decision-support systems, consequently decreasing unfairness. When the input consists of tabular data, these methods can be divided into three categories that focus on different parts of the systems [84]: *dataset pre-processing*, *in-algorithm treatment*, and *post-processing of the outputs*. While the literature does not provide guidance in the selection of the method to apply, it seems to primarily depend on the notion of fairness to optimize for, and on the actual context of the application. For instance, certain developers might only have access to the machine learning models and then would apply in-algorithm methods,

while data engineers might have the opportunity to transform the data before any kind of learning, which supports an earlier tackling of biases.

Mitigation through dataset pre-processing. For pre-processing, Luong et al. [516] propose a method that is inspired from situation testing, an experimental legal procedure to identify discrimination, in order to identify and later modify discriminative data labels. Zhang et al. [926] bring the ideas to use causal graphs to identify significant cases of unfairness, and to remove unfairness in the data through constrained optimization in order to maintain both utility and fairness of the dataset. Feldman et al. [255] propose data repairing methods. Hajian et al. [331, 332] target simultaneously fairness and privacy preservation in datasets through an optimization algorithm.

Mitigation through in-algorithm treatment. Algorithmic modifications of the training process mostly focus on adjusting the loss function of machine learning models through the addition of regularization terms to include the selected notions of fairness, for classification [195, 417, 631], for ranking [287, 757], for matching tasks [445, 804], but also recently in the context of recommender systems [97].

Mitigation through output post-processing. Post-processing relies on the idea that model's predictions can be made fair by defining specific thresholds that transform the continuous outputs of the inference model into binary labels [336, 195]. Specific methods vary in order to adapt to the specific group fairness metrics to optimize for, and sometimes to provide the option to defer the decision to the human operator [158].

WORKS DEALING WITH MULTIMEDIA DATA

In multimedia data research, we mainly identify two types of methods for mitigating biases. These are either pertaining to dataset pre-processing, or to in-algorithm treatment. These works are generally more recent and less numerous than for tabular data. In computer vision, in order to make the outputs of the systems less biased, datasets are often modified to increase the diversity of present objects and extrinsic properties (e.g., collection or transformation of data samples, creation of synthetic datasets [449]). However, the goal of these efforts is typically to improve model performance, not necessarily fair treatment of certain classes. This is for example addressed by Amini et al. and Quadrianto et al. [26, 649] who introduce fair feature representations that hide protected attributes. Directly controlling fairness in computer vision datasets is not a major topic yet [901, 238]. Natural language processing [807] typically modifies the training dataset (semi-manual data augmentation or annotation of samples with protected attributes), the embeddings of the samples as these have been shown to integrate unwanted biases from the large corpora of text on which they are trained, or the inference models. A more detailed account of these methods is given in [807].

4.3.4. TESTING FOR BIAS AND UNFAIRNESS

TABULAR DATA

Few works focus on evaluating the fairness of machine learning-based data-driven decision-support systems at deployment time, i.e., when ground truth for the new data samples

is not known.

Galhotra et al., Angell et al., Udeshi et al. and Aggarwal et al. [273, 30, 832, 11] propose test-suites to evaluate the fairness of software that relies on machine learning models, focusing on individual unfairness and developing methodologies for auto-generation of test inputs. For instance, the Aequitas framework [704] first proceeds to a random sampling of the input space to generate test cases, then the samples that are identified as discriminatory are used to further generate more test cases, by adding perturbations to these samples. In this case, it is not needed to know the ground truth, only the comparison between the model's inferences for the similar generated samples is important. Certain methodologies can identify more or fewer discrimination cases.

In contrast, Albarghouthi et al. [16] adopt a programming language perspective: they propose a way to formally verify whether certain decision-making programs satisfy a given fairness criterion (group or individual fairness) through encoding fairness definitions into probabilistic properties.

MULTIMEDIA DATA

For multimedia data, the same metrics are used as for tabular data. The difference lays in that the required information to compute the metrics, such as the protected attributes, are often not readily available, and often impossible to extract easily solely from looking at the data samples (for instance, it is questionable whether race or gender can be annotated simply by looking at the picture of someone without knowing the person). Additional context or expertise might be required, such as in the cases of annotating the dialects employed in text samples or the race of the person who wrote the samples.

In computer vision, a few manually created benchmarks such as Gender Shades of Buolamwini et al. [138] are used to test specific applications like face detection. In natural language processing, biases are quantified either by measuring associations between terms related to protected attributes, or by computing the prediction error of the data-driven decision-support system for the different subgroups represented by the protected attributes [807]. This often requires generating data samples where the protected attribute is controlled to perform a systematic evaluation, especially because a large set of protected attributes can be considered in these spaces.

4.3.5. BIAS IN CROWDSOURCING

Crowdsourcing is an essential component of many machine learning data-driven decision-support system workflows. It allows to collect data samples, or to label these samples so as to create ground truth labels to train the machine learning models on. From our analysis of existing works, we identify two meanings and research directions around bias in crowdsourcing. Closer to our topic, bias here refers to the way labels are attributed to data samples by annotators who project their own biases in the annotations [601, 602, 600]). Another meaning however refers more to unfairness, and the pay inequality of various annotators among each other or compared to the minimum pay in their respective countries [533, 79].

4.3.6. OTHER FOCUSES

Analysing the publications we retrieved from our systematic survey, we identify a few other emergent research directions, that have been developed to less extent until now, but that we believe are relevant to our topic, since they indirectly inform on issues around bias and unfairness either in the general development of the systems or in the data that could be used for these systems.

"FAIR" SOFTWARE ENGINEERING

Other lines of work within computer science research are also interested in fairness. We specifically highlight works on designing methods to develop fairer software [856, 487], coping with software designer biases [440, 711, 154, 390, 866, 669], fair processes to design software [285, 654, 106]. For instance, German et al. [285] see code reviewing as a decision process where codes from different categories of population might be more or less often accepted, Rahman et al. and Bird et al. [654, 106] point out that bug-fix datasets are biased due to historical decisions of the engineers producing data samples. Other papers such as [619, 383, 731, 294, 94, 78, 104, 841] reflect on how projects (data science process, creation of fairness definitions) are conducted and how unfairness is seen and might arise in general from the problem formulation perspective.

Inspired by these works, in section 4.8, we also propose expanding the software engineering process of data-driven applications with additional fairness requirements.

APPLICATION-FOCUSED ADAPTATION OF THE WORKS ON BIAS AND UNFAIRNESS

Certain works focus on bias and unfairness identification and mitigation methods for specific applications such as text analysis –e.g., Diaz et al. [222] address age bias in sentiment analysis–, social media news and existing polarization biases [228], fairness in self-driving vehicles [370], text processing [476]), web information systems and biases arising from them [712, 211, 579, 499, 616, 682, 563, 650, 782].

Certain of these works are especially important for the goal of developing fair decision-support systems since they raise awareness of potentially biased sources of data, that are later used to train the machine learning models. For example, Das et al. and Quattrone et al. [211, 650] show that user-generated content on Web platforms is biased towards certain demographics of the population due to the varied proportions of activity these demographics have (e.g., OpenStreetMap contributions are mostly from male users). We foresee this will have an impact on decision systems trained on datasets crawled from these platforms since the samples would be biased.

HUMAN-COMPUTER INTERACTION RESEARCH

Certain researchers from the human-computer interaction community work on identifying the needs of data and machine learning practitioners in relation to new unfairness issues that arise from the application of data-driven decision support systems in real-life scenarios both for public and private sectors [369, 846].

Besides, the Fairness, Accountability, Transparency (FAT*) community is also interested in problems related to social sciences, like the impact of publicly pointing out biases in company software [658], or the influence of decision-making systems on populations [566]. These works outline new research challenges for which technical processes and tools could be further developed.

4.4. DATA ANALYTICS: LIMITATIONS

In this section, we highlight the main limitations of current works on bias and unfairness, as they are argued by different research communities.

4.4.1. LIMITATIONS WITHIN EACH RESEARCH DIRECTION

The topics of the previous subsections each bear certain limitations and research challenges. Methods for identifying, testing, and mitigating biases do not allow for the development of fully fair and accurate systems and do not enable understanding where the unwanted biases come from in the systems for each of the different unfairness metrics. Besides, these methods are only adapted to increase fairness scores as measured by current metrics, but a system fair according to one metric might not be fair for humans, as existing fairness definitions do not align fully with human perceptions of unfairness. Also, due to the impossibility theorems between multiple metrics, there is currently no solution to build systems that are considered fair with regard to multiple metrics, whereas the combination of multiple metrics might be closer to the human notions of fairness. Methods do not all handle well intersectionality –when fairness is defined over the combination of multiple protected attributes–, whereas this is a closer notion of fairness than formalizations over single protected attributes. Finally, existing methods almost all assume the prior knowledge of the protected attributes but this assumption might not hold in practice. As for crowdsourcing works, not all biases coming from crowd workers are known from researchers or dataset developers until now, and hence they are not all dealt with when creating datasets.

4.4.2. LIMITATIONS IN THE CHOICE OF DIRECTIONS

Besides the above challenges tied in with the current approach of the issue that centers around machine learning algorithms, more general limitations are highlighted by certain works. Mainly, the human-computer interaction community [369] suggests conducting more research to bridge the gap between existing machine learning methods and their applicability by industry practitioners. Works with professionals have been conducted to understand industry needs to deal with unfairness and bias and compared to existing research, showing that both bias mitigation and evaluation methods might not be adapted to real uses. Also, the software engineering community suggests taking a step back on the development of the systems to consider fairness in all development and deployment steps. We discuss these gaps in more details below.

ALGORITHMS AND TOOLS FOR DATA BIAS MITIGATION

Holstein et al. [369] point out that certain practitioners have more control on the data collection and curation steps than on the machine learning algorithm development, but that existing methods primarily focus on mitigation in the algorithm. Thus, we later advocate focusing on the data aspect of biases and unfairness.

Also, frameworks to help the selection of appropriate unfairness mitigation methods accounting for trade-offs with other performance measures are needed.

SUPPORT FOR EVALUATION

Practitioners also lack tools to facilitate the building of representative evaluation datasets and to identify and apply adapted metrics.

Most metrics are adapted for cases of allocative harms, that can arise when the goal of a system is to allocate resources to multiple stakeholders. They are however not often adapted for representational harms that arise from the classification of individuals in different categories, or from the association of individuals to (stereotyped) characteristics. This would be especially relevant in natural language processing (e.g. word embeddings denoting females are more closely associated to a number of job categories like maids and janitors contrary to the male embeddings) and in computer vision (e.g. images representing Black persons are more often classified as containing violence than images representing White persons). Also, most metrics assume knowledge of individual-level features whereas for privacy reasons this knowledge is often absent.

Besides, many unknown unknowns such as identifying before implementation or deployment the populations that could suffer from unfairness remain. Most research assumes the knowledge of the protected categories of population, generally coming from legislations, but there might be additional alarming context-dependent unfairness cases.

GUIDANCE IN SOFTWARE ENGINEERING

Many research opportunities are foreseen in the software engineering process in order to build ethics-aligned software. Roadmaps to develop ethical software are proposed [54, 134], where the needs for methods to build ethical software, to evaluate the compatibility of the software with human values, and to help stakeholders formulate their values are highlighted. In this direction, Hussain et al. [382] and the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems [446] respectively argue for a collaborative framework to create software design patterns including social values (such values would be unwanted biases and different types of unfairness in our case) and for standards on algorithmic biases in order to provide a development framework that could support the creation of value-aligned algorithmic software. We believe this is also highly relevant for the data management community as, for instance, the data schemas developed in discussion with stakeholders need to be aligned with the values to integrate into the decision-support systems.

4.5. DATA MANAGEMENT: METHODOLOGY

In this section, we first explain our survey methodology for bias and fairness research specifically in data management, and establish a quantitative research overview. This will serve as a starting point to identify research gaps in the next sections. Especially, in the previous sections, we established the general state-of-the-art in computer science research, and in the next sections, we compare it to data management works. Particularly, we investigate the extent to which data management research has differentiated until now from other research, with the intuition that more data management-specific activities should be investigated in the future. Besides, we map the data management research to the workflow of decision-support systems to identify important research gaps.

4.5.1. SURVEY METHODOLOGY

We surveyed a selection of data management venues for articles dealing with unfairness. This was conducted between August 2019 and December 2020, using two search engines (Google Scholar and DBLP). We retrieved papers using the keywords "bias", "fair", "disparate", "discrimination", "responsible", "diversity" and "coverage" combined with OR clauses, appended with constraints on the publication venues, covering the full publication history of the venues. The keywords were chosen to encompass as diverse publications as possible, as we noted that "fairness' is not the only term used for describing related works, but also notions of "discrimination", "bias", "diversity", or more general notions of ethics and responsible computing are employed.

In particular, we included publications from the ACM TODS, VLDB and TKDE journals, CIDR, ICDT, ICDE, SSDBM, EDBT, SIGMOD/PODS proceedings and the Data Engineering Bulletin [3]. With snowball sampling, we also selected the data management papers cited by the initially retrieved papers. We filtered out the ones not actually addressing fairness topics of systems where some kind of decision is made, which relates to human individuals. Excluded papers mostly concern the fair allocation of computing resources or tasks between components of a computing system.

4.5.2. QUANTITATIVE OVERVIEW

From the quantitative analysis of data management papers concerning unfairness and bias, we first of all notice that only 34 papers focus on the problems of biases in data-driven decision-support systems (*DDSS*), of which only 17 full papers; other than those, we see that mainly demos (5), tutorials (3), review papers (3) or vision papers (2) are presented, next to short papers (2), workshop paper (1), panel discussion (1), keynote (1). Most of these works have been published in the last 2 years.

This number is rather low compared to other research domains in computer science like machine learning, human-computer interaction, or data mining where unfairness is a common topic since 2010 and where there are more than a few hundred papers. While this observation is hardly surprising as most issues related to unfairness stem from the application of automated, often machine learning-based, data analysis techniques to human-related data, we argue that there should also be algorithm-agnostic bias considerations on the data management side.

4.5.3. MAIN RESEARCH DIRECTIONS

All of the papers that we retrieved from data management venues, searching for a wide range of publications related to unfairness, fall into one of the topics also addressed by research outside of data management introduced in section 4.3. However, two topics identified in section 4.3 are not covered at all in data management (perceptions of fairness and testing of data-driven decision-support systems).

Yet, it is also important to note that several works are interested in questions of fair rankings, set selections, and data coverage, that are not discussed specifically in other disciplines. These questions are of importance for machine learning workflows where the pre-retrieval of "unbiased" datasets from databases could be necessary. These works

[3] The Data Engineering Bulletin has a full special issue on fairness. [137]

can also be used independently of any machine learning model, simply as data analytics tools that provide decisions on data samples, such as for the tasks of ranking or selecting a limited number of candidates for job hiring.

The application areas are diverse; most of the times, the proposed methods are of a general nature, but sometimes specific to selected use-cases such as fair web page ranking [180], fair OLAP queries [706], fairness and trust in multi-agent systems [874], or fair urban mobility [896].

4.6. DATA MANAGEMENT: STATE OF THE ART

Here, we discuss current related research topics worked on in the data management community, map them to the topics discussed in the previous sections, and outline the main existing approaches.

4.6.1. DEFINITIONS

Three papers propose formal definitions of fairness, expanding on existing machine learning and data mining literature. Yang et al. [903] propose measures of fairness in ranking tasks, whereas Salimi et al. [710] propose a fairness definition for classification tasks to overcome limitations of previous definitions solely based on correlations or causality. Farnadi et al. [252, 253] introduce fairness definitions, a first-order logic language to specify them, and mitigation methods. They argue that fairness is a concept depending on relations between the individuals within a dataset.

4.6.2. IDENTIFICATION

We identify multiple works that relate to the identification of undesired biases in datasets. These works divide into three main categories depending on the approach they follow, and the problem conditions that they define for themselves. While the first category of works is close to the data mining topics discussed in prior sections, the other two – coverage and unbiased query results– are specific to the data management community.

DATA MINING APPROACHES

Similarly to other data mining works, some papers aim at identifying biases seen as discrimination within datasets. The context ranges from datasets of potentially discriminative historical decisions [927, 330], with methods potentially encoded into the database system [697], to datasets of ranking scenarios [245, 288] where unfair treatment towards specific groups might arise (these groups are not predefined), and to text datasets [912] where the semantics of certain user-generated comments might be discriminatory.

COVERAGE

Another topic related to the identification of biases within datasets more specific to data management literature is the notion of *data coverage*. Coverage relates to the idea that data samples in a dataset should sufficiently cover the diversity of items in a universe of discourse [51]. Without adequate coverage, applications using such datasets might be prone to discriminative mistakes. For example, certain computer vision models of Google performing image classification and object detection have been reported to have

mistakenly labeled a Black woman as "gorilla", likely because the original training dataset did not cover enough images of Black women.

Dataset coverage characterization and mitigation methods. Asudeh et al. [51] first proposed a formalisation of the coverage problem. They also present and evaluate methods both to efficiently evaluate the coverage of a dataset with respect to thresholds set by a practitioner for each dataset attribute, and to identify the type of data samples that are preferable to collect to solve the coverage issue accounting for the cost of data collection. These methods are based on the idea that representing a dataset as a pattern graph allows pruning a large amount of insufficiently covered data patterns represented as pattern relationships. Their link to coverage can then be exploited efficiently, instead of linearly traversing the whole dataset to identify uncovered patterns and to reason about their relationships.

Moskovitch et al. [565] take a different approach, aiming at efficiently estimating the number of items fitting different patterns in a dataset. This is based on pattern profiling and caching their statistics under resource constraints. Estimation functions estimate the count of any selected pattern with trade-offs between accuracy and efficiency based on those cached statistics. Lin et al. [505] argue that one of the main limitations of many previous works is the assumption that the considered dataset is constituted only of a single table. Applying existing methods to a realistic multi-table setup is shown prohibitively expensive. Instead, the authors propose a new parallel index scheme and approximate query processing to explore dataset coverage efficiently.

Coverage-informed database queries. The previous approaches aimed at identifying coverage issues in a dataset that was "found" in a general fashion (as opposed to collected for a specific application in mind). Other methods focus on a setup with data present in a data warehouse, and propose to retrieve a subset of the data in such a way that the data verify a specific application-oriented coverage objective. In this context, Accinelli et al. [3] propose a method to rewrite queries whose results would violate a specific coverage constraint into a similar query whose results now fulfill the constraint. In a similar fashion, Salimi et al. propose a way to identify biased results of OLAP queries, and rewrite similar queries to obtain unbiased results [706, 705].

Dataset nutritional labels. Some works promote the idea of creating *nutritional labels* for datasets, similar to the machine learning community which proposes to make datasheets to report on the creation of datasets [283] or to describe machine learning models [556]. In machine learning, these datasheets are intended for accountability, easier auditing of models, or for understanding of the limitations of models or datasets with respect to generalization abilities to extended tasks. Nutritional (data) labels in data management take a lower-level and more in-depth look at the datasets, and allow practitioners to interactively explore dataset distributions to identify diversity and coverage issues within the datasets themselves.

Particularly, Sun et al. [806] develop MithraLabel, which aims at providing flexible nutritional labels for a dataset to practitioners, showing the distributions of each selected attribute, functional dependencies between attributes, and the maximal uncov-

ered patterns. When a dataset is added to the system, a set of dataset labels that summarize information about the dataset are shown, such as how representative of minorities the data is, how correlated the different attributes are (especially with respect to the protected attributes, the number of errors (e.g. missing values), etc. In addition to showing such data, its back-end optimizes for the trade-off between the amount of information given (through the widget), and the space the widgets use, by "learning" how preferable each widget is for different tasks based on logs of practitioners' use. Additionally, MithraCoverage [411] allows interaction with aforementioned coverage methods, e.g. to filter out the invalid patterns, but also to fix the parameters of the method such as the coverage threshold, or the attributes the practitioner wants to investigate particularly.

UNBIASED QUERY RESULTS

Most previously presented works focus on retrieving a fair or diverse set of data tuples from a single dataset. Orr et al. [599] adopt a different setup and problem. They assume that existing databases are biased in a sense that they might not accurately reflect the world distributions of samples, and that practitioners can have additional access to aggregate datasets which contain information that might reflect the real distributions. From this new framing of the bias problem, they propose Themis, a framework that takes as input the original dataset, the aggregate dataset, and a practitioner's query, and outputs results that are automatically debiased by learning a population's probabilistic model and reweighting samples accordingly. This is the first work in the area of open-world databases that aims at debiasing query results in that sense of bias.

4.6.3. MITIGATION

Mitigation methods focus on modifying datasets, e.g. for classification tasks [710, 815, 465], or ranking tasks [48, 465, 317]. Most methods are seen as data repair methods where the tuples or labels are modified, and would merit being unified with other data cleaning methods as their application might influence unfairness [815].

We identify three main trends in mitigation methods, that focus either on data or feature representations. Data works consist in transforming data for classification tasks by relying on causality notions [707, 708, 709], or in tackling the problem of retrieving fair, possibly ranked, data subsets [50, 903, 797]. Feature representation works aim at learning data representations for which the outputs of classification tasks are fair [465].

4.6.4. CROWDSOURCING

Unfairness in crowdsourcing is also investigated, similarly as in the other domains studied in the previous sections. Works either look at unfairness towards the crowd workers, such as Borromeo et al. [122] who propose a list of axioms to guide the creation of fair and transparent crowdsourcing processes –task assignment, task completion, and worker compensation–; or look at resolving unwanted biases in labeled data. It is argued that such biases in labels can stem from personal preferences or differing expertise of crowd workers [925], from labeling "trends" [355, 560], or from the subjectivity of the object to review in evaluation systems [473].

4.6.5. DATA SCIENCE WORKFLOW

Different from works in the other domains, a few recent works are interested in developing tools at the intersection of data management and machine learning. For instance, Schelter et al. [726] note that the existing tools developed for fairness do not support practitioners (and researchers) fully in developing the whole data science workflow responsibly. Instead, they simply let them apply various fairness metrics and bias mitigation methods without being aware of their interaction with other parts of the workflow such as data cleaning, separation of the datasets into independent training and test sets, etc. They build FairPrep, a framework on top of the existing IBM toolkit AIF360, in order to fill this gap: practitioners input data and their desired pre-processing methods, as well as choose a machine learning algorithm, and the framework automatically processes this information, trains the model and outputs its complete evaluation based on both performance and fairness measures. This allows avoiding errors in building the workflow, such as for instance leaking data information from the training to the test set when handling data errors such as missing values, when engineering features or tuning a model's hyperparameters, etc. Besides, experiments with their framework show the lack of consideration of existing fairness works from the machine learning community for critical data engineering activities such as data cleaning.

With the same idea that the data pipelines might unintentionally inject biases, Yang et al. [902] developed a tool that automatically extracts a directed acyclic graph representation of the data pipelines and data flows from the code of the pipelines, and provides information on the way each vertex impacts the distribution of samples based on protected attributes and target labels. By generating a report with the graph and this information, a practitioner can investigate potential bias issues of its pipelines.

4.7. DATA MANAGEMENT: RESEARCH GAPS

In this section, we identify research gaps between data management research on bias and unfairness (section 4.6), bias and unfairness research in other fields of computer science (section 4.3), and typical development practices of data-driven decision-support systems. These gaps are summarised in Figure 4.2. This is the basis for developing a new approach to the issue in the next section.

4.7.1. METHODOLOGY

Approach. To identify these gaps, we first outline all activities performed over the full lifecycle of a data-driven decision-support system, from development to deployment. This list provides us with the basis to reflect on potential research gaps, as it encompasses the necessary set of activities to develop the systems, and these activities are by design both the sources of bias and unfairness and the opportunities to solve these issues. These activities can be associated with one or multiple general unfairness-agnostic research areas, usually stemming from machine learning and data management. For instance, the construction step of a decision-support system consists of building both a data management and a data analytics set-up. Data management activities at this step map to multiple research areas within data management such as data integration or data curation.

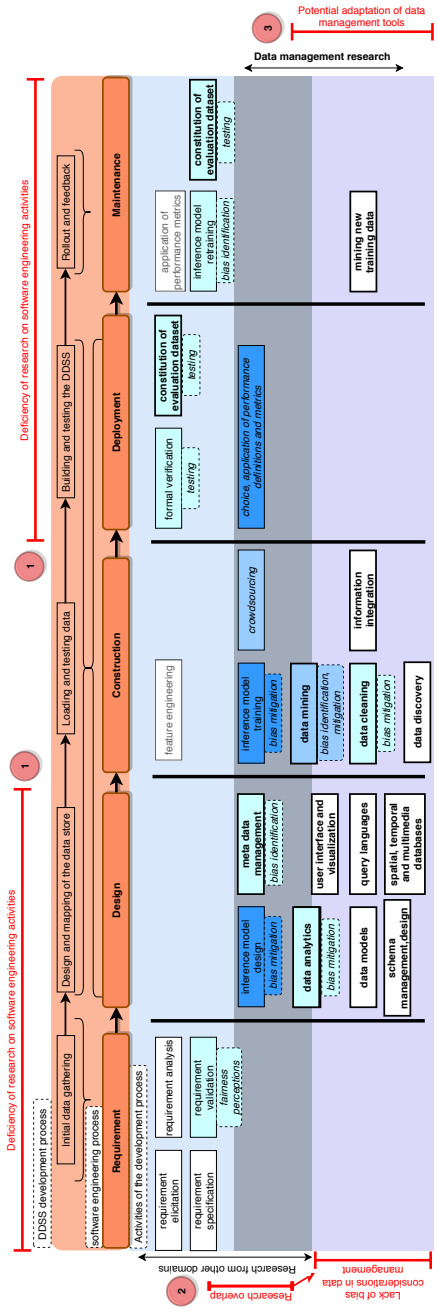


Figure 4.2: Activities relevant to bias and their amount of bias-related research (white: no research; light to dark blue from few to plenty of research). Data management activities are bold.

Then, we map the research activities identified in the previous sections onto the aforementioned mapping. This allows investigating the extent to which the different unfairness-agnostic research and non-research activities are covered by unfairness-related research. In cases where an activity is not covered, it might be because it does not interact with unfairness at all, or because it has not been studied yet. In any case, we analyze it because it could still be useful to resolve certain unfairness issues. Such analysis brings us to identify three main gaps, either related to data management activities for addressing unfairness, or to data management activities that create unfairness, or more generally to whole stages in the lifecycle that have not been thoroughly investigated.

Lifecycle of a data-driven decision-support system (in orange). The development process of a data-driven decision-support system is divided into five main stages as described in [643]: 1) the initial data gathering, 2) the design and mapping of the data store, 3) the loading and testing of the data, 4) the building and testing of the system, and 5) its rollout and inclusion of feedbacks from its users. These stages are easily mapped to the typical software engineering process [123]: 1) requirements engineering, 2) system design, 3) system construction, 4) system testing, and 5) maintenance of the system after deployment. While the description of the lifecyle of the decision-support systems focuses on the distinction between data and other aspects of the system, the software engineering description mostly focuses on the general stages of development.

Activities performed during the lifecycle (activities placed in boxes, we differentiate between data-related activities in bold, and other ones). We identify the specific activities performed in each stage of the lifecycle. To do so, software engineering literature [123] indicates the activities which are general to any kind of software. These activities span the requirement engineering stage (requirement elicitation, analysis, specification and validation), the design stage (system and user-interface design), and both the testing and maintenance phase (these last two stages are not detailed for simplicity and because they might not be applied thoroughly yet for the specific case of data-driven decision-support systems).

Data management literature presents activities or topics that are specific to the data aspects of the lifecycle. These are extracted from the common list of research topics in data management venues [4]. For the design stage, we identified data models, query languages, schema management and design, meta-data management, user interface and visualization, data analytics, and specific issues on spatial, temporal, and multimedia databases. For the construction phase, we found data mining, data cleaning, information integration, data discovery, and crowdsourcing.

Additional activities that are specific to machine learning [24] are found in the design stage (inference model design), and in the construction stage where we identify data collection (shown as data mining because of the overlap with data management literature), data labeling (shown as crowdsourcing for the same reason), feature engineering, and inference model training. In the testing stage, only model testing is added. For the maintenance stage, model monitoring and model update are identified. These last two stages

[4] List from https://vldb2020.org/research-track.html.

are further subdivided. Testing is composed of the choice and application of performance definitions and metrics, the constitution of evaluation datasets (these two are for experimental testing), and the formal verification. For the maintenance phase, we found mining new training data, inference model retraining, application of performance metrics, and constitution of new evaluation datasets, since the context of application of a system might shift or expand, and hence new data must be collected, and the machine learning model must be retrained to account for this shift.

Mapping to current research on bias and unfairness (colors of boxes). We map current research on unfairness (from light to dark blue, representing the quantity of current literature on that topic) outlined in sections 4.3, 4.6 to these activities (the topics identified in the previous sections are in italic for easy identification). This enables to identify where research is focusing and where it is lacking. In the following, we explain the findings of this analysis, grouped by their topics.

4.7.2. BIAS-AWARE REQUIREMENTS

A first observation is that some stages of the development process are more researched than others. Specifically, the design and implementation of inference models are the most covered topics [402], along with metrics or definitions for fairness. There is also a shorter line of work on data mining, mostly focusing on structured data and text data.

In contrast, works on requirement engineering and subsequent database design (elicitation, translation to specifications), system testing, and maintenance (continuous testing with respect to the identified requirements) are much fewer. These limitations are also partly highlighted within the Human-Computer Interaction (HCI) and the Software Engineering communities, as explained in section 4.4. Yet, many researched methods mostly focus on bias mitigation in the algorithmic part. Hence, developing tools to model, design, and construct better datasets should be a priority.

4.7.3. BIASES IN DATA MANAGEMENT ACTIVITIES

A second observation is that for many traditional data management activities which might introduce unwanted biases, there is little to no research investigating their impact on biases at the output of the system. This covers for example data cleaning, data discovery, or data integration [49]. On that note, Stoyanovich et al. [796] encourage the exploration of the possibilities to mitigate biases early in the data life cycle of the decision support systems. Abiteboul et Stoyanovich [2] further outline that several principles from regulations about responsible data-driven systems, possibly outside the scope of bias and fairness such as the right to "data portability", would require investigation and adaptation of the data management community. For instance, ensuring "the right to be forgotten" for an individual would mean investigating how this right translates in every layer of a database, while accounting for possible dependencies with the data tuples representing this individual and other connected individuals.

We could not identify any significant effort on bias and unfairness considerations in data modeling, schema design, and data provenance topics, even though these activities define the information on which the inference model and decisions are based.

4.7.4. DBMS ACTIVITIES FOR BIAS MITIGATION

A third observation is that part of the encountered research efforts in data management mirrors the works in other domains on bias and unfairness for data-driven decision-support systems (section 4.3) with similar approaches and limitations. Especially, there is also a focus on definitions, metrics, and mitigation at the algorithm level. However, further re-purposing or adapting some of the approaches developed in other data management works could serve to identify or mitigate certain biases already in the datasets. This holds especially for data cleaning methods like error detection and data repairing, data analytics and efforts in data modeling, and also research on multimedia data.

Only a small part of current data management research makes use of such methods. The idea of mitigating unwanted biases through data repair methods is similar to those proposed in data mining, but tends to be more general and agnostic with respect to the employed analytic methods as presented by Salimi et al. [707]. Two vision papers are of note on the topic. The first one proposes to unify data pre-processing and inference systems arguing that fairness, accountability, and transparency could be seen as database system issues before applying ML and outlining how a platform for data analytics could help solve these issues [795]. On the other hand, Stoyanovich et al. [794] claim that methods to automatically attribute labels to datasets and ML models (meta-data) to prevent their misuse are needed to prevent the creation of additional biases.

Asudeh et Jagadish in a tutorial [49] suggest that works around data profiling and provenance could be adapted to fulfill the need of practitioners for tools to explore biases in data. Besides, Abiteboul et Stoyanovich. [2] discuss how various regulations such as the GDPR in Europe advocate for responsible development and use of data and data-driven decision support systems, and make the case there that the data management community could support progress on principles like transparency by adapting existing works for instance on data profiling to better expose the data statistics for a richer interpretation of the systems' outputs.

Orr et al. [598] proposed an in-DBMS method for practitioners to query a database and retrieve results which are automatically cleared from dataset sampling biases introduced during the data collection step. This work is the closest to the approach we advocate in the next section since it aims at helping practitioners to mitigate biases within the database, although it is not made for the purpose of further training an ML model.

4.8. LIMITATIONS: ROADMAP FOR FUTURE TECHNICAL RESEARCH OPPORTUNITIES

In the previous sections, we identified both limitations and gaps stemming from the current approach to tackle unfairness of data-driven decision-support systems, i.e. approaches focused on the machine learning algorithms themselves, and general research gaps stemming from existing data management activities. The main limitations are the difficult application of existing algorithmic methods by practitioners, and the fact that such methods do not allow to build fully fair systems. In this section, we reflect on a way forward to overcome these limitations. Particularly, the limitations hint at a possible research shift in order to solve existing unfairness issues: not only should we develop algorithms robust to unfairness but also data methods to mitigate unfairness, and prac-

tical tools to support and ensure the use of such methods by practitioners. In the next section, we discuss the challenges arising from this way forward.

4.8.1. ELICITING AND ENFORCING FAIRNESS REQUIREMENTS

We advocate focusing on eliciting and enforcing bias and fairness requirements already early in the system design workflow. This allows to clarify the goals of a system in relation to fairness, and then brings the possibility to guide practitioners along the system development cycle to create a system that verifies these goals. Thus, the fairness requirements serve as a foundation of a bias-aware data-engineering pipeline. Here, we outline how such bias and fairness requirements can be applied conceptually and how they integrate into existing database management system architectures.

PROPOSED WORKFLOW

We propose a new workflow for practitioners building data-driven decision-support systems, encouraging fairness-by-design.

Ideally, before designing and building a system, a practitioner would define a list of requirements, including fairness requirements.

These requirements would then be translated into constraints on both the data used for training the system and inputted at deployment time. These constraints would impose statistical conditions with regard to defined protected attributes that would ensure that a dataset could be considered fair for the requirements at hand. At training time, this would increase the likelihood that the outputs of a model trained on such dataset are fair (note: an "unbiased" training dataset does not guarantee an unbiased resulting system since new unwanted biases might arise from the machine learning algorithm used or small unwanted biases in the data might be reinforced by the machine learning model, but helps); while at deployment time, it would monitor whether the predictions made for new data points are fair. Constraints at training and deployment time might differ depending on the initial fairness requirements, the associated characteristics that a training data should bear, and the appropriate slack for such training data characteristics needed to ensure reasonable fairness measures.

Continuous checks of bias constraints on the system's outputs are needed, analogously to continuous testing in software deployment, since the fairness of the system might vary in case a distributional shift happens between the training data and deployment data.

In cases where the data would not follow such constraints, either data curation methods could be employed to remedy such issue at training time, or this would be an indication that it is mathematically impossible to verify simultaneously the multiple fairness requirements and other requirements, and hence the system should not be developed or the requirements should be reviewed. At deployment time, the constraints not being verified would indicate the necessity to defer the decision to a human agent, or the necessity to retrain the model on updated data.

ADDRESSED LIMITATIONS

This new approach considers the quality of the data as a core issue. Our intuition is that it would overcome multiple challenges that are typical concerns of different research

communities, besides unfairness, and that interact with unfairness considerations: cost, time, robustness and practicality for the machine learning and software engineering communities, societal impact and trust for the human-computer interaction community. They are the following challenges:

Fairness. The main source of biases is data, hence investing research to understand, detect, and control bias in data allows to build less biased datasets with regard to specific fairness requirements and consequently to train fairer systems.

Robustness. Modifying optimization functions of machine learning algorithms or post-processing decisions can have unforeseen effects in cases where the application context and data would change. In contrast, we argue that enforcing inspection of data biases in the early stages of development and during deployment would result in more robust systems since potential issues would be identified earlier.

Practicality. Practitioners might understand issues and methods in the data stages of the development of a data-driven decision-support system better than those related to the inference model. For example, obtaining extra training data to balance a dataset might be easier than adjusting machine learning algorithms; hence, data-focused tools could be more applicable than current methods. Considering that transfer learning is becoming a common practice (i.e. using pre-trained general models and then fine-tuning them for a specific application), the availability of "unbiased" data for the fine-tuning phase is crucial.

Cost and Time. By ensuring that training data has no bias issues, the resulting trained models will likely behave in a more desirable fashion, thus fewer costly training and re-training cycles are needed to achieve the desired system behavior. Ultimately, the process would be more effective and less costly.

Societal Impact. Establishing requirements would encourage considering societal impact already in the initial stages of development. Past cases which did not explicitly state and enforce their fairness requirements showed the potential negative impact of building these systems without accounting for potential issues: Microsoft's chatbot Tay became racist after its deployment because it was constantly retrained on data fed to it by layman users and had to be shutdown [403], while the automatic CV screening tool of Amazon was shown to be discriminating against women after release [408]. Many of these issues could have been foreseen and mitigated if undesired bias identification and fairness were central design goals of these systems.

Trust and Informed Decision-Making. Finally, by explicitly communicating bias and fairness design goals and validating systems respectively, trust can be facilitated between the system and stakeholders or users who will have a better understanding of its behavior. This can also support building an accurate abstract model of the capabilities of a system. This will lead to better decisions, as the performance of a human decision-maker is

dependent on his/her mental models of the problem and of the system and on tools at hand [630].

4.8.2. REQUIRED DBMS EXTENSIONS

By shifting the focus from the algorithms to the data, we foresee the need for two new core extensions to database management systems, that would support the application of the proposed workflow.

Bias Data Constraints. Fairness requirements identified in the requirements elicitation phase need to be formalized such that they can guide the system's development. Furthermore, they need to be validated or verified across the system's lifecycle. New *bias data constraints*, expanding on existing data constraints, could be used to encode and enforce data-related bias requirements.

Bias Curation Methods. Data curation methods addressing bias by transforming, adding, or removing data instances would be needed in cases where the constraints are violated. While also algorithmic mitigation techniques (see section 4.3) can be used, we argue that data curation is often more effective or practical [369]. If the constraints are violated, the system designers would be warned to take action or prevented to train the models.

Embedding into the DBMS. To support and enforce the use of bias constraints and curation methods, existing database management systems should be extended to integrate them, an idea also suggested in [49]. This will be important as checking bias constraints can be very data-intensive. By embedding this into the database management system, we can take advantage of existing components like indexes or system catalog information, allowing for more efficient implementation. The creation and integration of these components bring a multitude of data management research challenges that we highlight in the next section.

Here we highlight the specific research challenges which need to be addressed for realizing the bias and unfairness-mitigating extensions proposed in the previous section.

4.8.3. FORMALIZATION AND MODELLING CHALLENGES

BIAS-AWARE SCHEMA DESIGN

While selecting fairness notions for a specific use-case is not an easy task, defining the exact attributes and their allowed values to base the constraints on and the subsequent design of the database schema is also complex. Formally understanding how the granularity and ranges of the values in the database schema influence performance of the system and measurement of its bias remains to be investigated. For example, let's assume that the loan attribution model should not discriminate against young black men, and that the dataset contains gender and race as categorical attributes and age as an integer. After choosing a fairness definition, deciding how to transform age into a categorical attribute can have direct bias consequences. Defining protected classes (male, black, [10-23]) or (male, black, [10-25]) as protected attributes would both surface and measure different biases. Different mappings of age to its protected class "young" can

create different system behaviors: the granularity of the categories chosen would influence both the performance and fairness of the trained inference model. This gets even more complex when the bias constraints are defined over several attributes to transform. Similarly, this transformation might have an impact on the similarity measures used in the constraints for individual fairness since tuples similarity depends on their attributes.

PREDICTING THE FEASIBILITY OF A DATA-DRIVEN DECISION-SUPPORT SYSTEM

At the start of the workflow, determining whether bias constraints can be verified along with other requirements (e.g. accuracy performance, cost, amount of data) and other data constraints before designing and implementing a system would enable to save a great amount of time and computing power, while it would also allow to possibly refine requirements and resources allocated for a system. For instance, in case a practitioner has a specific amount of loan data and wants to build a data-driven decision-support system to automate the decision of giving out a loan, knowing before building the system and training a model that it will not be able to reach a minimum required accuracy and fairness would save efforts. Until now, few theoretical works [442, 182] have been proposed that investigate such feasibility of requirements. Existing results focus on the diverse fairness notions that can contradict each other.

Using impossibility results for fairness notions [182], certain impossible scenarios can already be determined analytically. Predicting a measure of each requirement, potentially via simulation through the training of simple inference models could also give empirical indications of the feasibility.

4.8.4. ALGORITHMIC CHALLENGES

There exist few bias curation methods from the data mining and machine learning communities, however, they are still limited in scope (e.g. the intersectionality of multiple protected attributes is not usually handled by current methods). More research is needed to establish approximation algorithms that would guarantee bias constraint satisfaction on the training data. These algorithms could transform existing data (like data resampling, data label modification, or variants of database repairing methods [707]) possibly with inspiration from existing data cleaning methods, synthesize new ones, or guide the collection of additional records.

Additionally, nearly all data-driven decision support systems rely on elaborate data engineering pipelines for preparing, transforming, integrating, cleaning, and finally ingesting training data, test data, and live data. Bias curation needs to be integrated within such data engineering pipelines. Also, existing steps of data engineering pipelines might have unforeseen and insufficiently understood consequences and effects on data bias. For instance, cleaning a dataset from its outliers might remove data from the protected minority class and hence a bias curation method would not have access to such data anymore, missing-value imputation methods might skew the dataset towards the protected or non-protected group and hence might add unwanted biases, so new methods would be needed to allow for the application of the bias curation methods, etc.

Only the interaction between bias and data cleaning has received preliminary attention [815, 726]. Hence, future work needs to investigate the impact of the previous activities on data biases, and the interaction with the bias curation methods. This would

lead either to providing guidelines on the workflow to follow, or to the creation of new algorithms that would integrate curation and integration or cleaning simultaneously.

4.8.5. SYSTEM-ORIENTED CHALLENGES

Adapting existing mechanisms in database management systems for supporting the bias constraints exhibits multiple challenges. The bias constraints would bear some similarities with existing database constraints, but also differences that would make their implementation and use not straightforward. We develop here the comparison with traditional constraints and highlight foreseen challenges.

CONSTRAINT EXPRESSION

Translating fairness metrics into SQL constraint language, possibly by additionally using user-defined functions, is the first step and challenge to allow the support of bias constraints. The way to encode these constraints would need to be as flexible as possible to accommodate most definitions of fairness and possibly new ones.

Certain constraints would be specified on protected attributes, other attributes of the data, and possibly on the decision attributes (actual decisions and/or predictions). The exact test of the constraint could cover statistical tests for undesired biases such as unwanted correlations between protected and other attributes or checking for potential "wrong" decision labels (e.g. [696]). For instance, in case fairness towards groups is important, the acceptable data distributions for each protected class can be specified. In many cases, these would be egalitarian distributions [848], but also non-egalitarians constraints could be relevant. For example, an AI-assisted hiring tool might want to positively discriminate against female applicants to address issues with employee diversity.

Inspiration from existing ways to encode data cleaning rules could be taken to express the bias constraints. For instance, denial constraints which are declarative specifications of rules a dataset should respect [186], could be investigated, especially for individual fairness which relies on the similarity between tuples.

CONSTRAINT CHECKING MECHANISM

A new set of challenges in order to implement bias constraints efficiently using current database technologies is the result. The use of triggers could be investigated as a tool to check for the constraints.

Because the constraint functions are expensive to compute, an envisioned research direction is to investigate how to incrementally compute the statistics that make the constraints over multiple batches of data, in order to avoid the whole re-computation at each check. Possibly existing system catalog statistics used for query optimization could allow to speed up such computation while reducing the resource consumption.

Bias constraints could be checked when a sufficiently larger number of records has been added or modified. Several policies for monitoring them would be useful: checking for constraint violations after initially populating the database, checking for violations when training data is retrieved for training an inference model, or when adding a large number of training tuples during system maintenance phases, and finally checking for violations when a significant number of new decisions are suggested by the system before accepting them.

4.8.6. GUIDANCE FOR DBMS USERS

As a major practical challenge, we identify the need for guiding a practitioner through the process of specifying fairness requirements and bias constraints. Certain applications might rely on country-specific regulations, while others might not have well-established policies. As there are a plethora of different fairness definitions, choosing the correct metric and setting the correct parameters is far from trivial due to the abstraction gap between application (fairness as an abstract norm) and constraint model (fairness as a mathematical object). Therefore, we envision a guidance component that could come in form of wizards, or an IDE that can provide suggestions based on data profiling of potential biases and on existing regulations.

A human-in-the-loop approach could highlight these biases, and then from feedback provided by the practitioners about the biases, it could uncover the undesired ones and automatically infer related fairness requirements, bias constraints and their prioritization. User studies could also be conducted to understand the actual difficulties and questions that practitioners would like to address.

Similarly, practitioners could be helped by having guidance frameworks and interfaces for deciding on bias curation methods to apply, that would visualise their impacts on different categories of population and on the other important factors in the requirements (e.g. cost, time, accuracy, etc.).

4.8.7. MULTIMEDIA DATA-BASED CHALLENGES

Applications using multimedia data such as images, texts or videos have typically the same aforementioned challenges, but additional difficulties arise.

For instance, for checking bias constraints, it is difficult to extract protected attributes or other semantically interpretable features from an image or text. Hence, it is difficult to generate necessary meta-data to apply the constraints, and to generate new representative test cases to check for the constraints. This task is currently performed manually for images and semi-automatically for text which hampers scalability and real-world applicability.

A similar issue arises when curating data for bias. Structured data algorithms would not be easily applicable since no interpretable attributes would be available to reason on. One direction to investigate could be to transform multimedia data into structured representations on which to apply the aforementioned algorithms. Possibly, crowd workers could be asked to annotate protected attributes, to produce or collect new related samples following certain templates (such as in [840]), or new automatic methods like GANs (Generative Adversarial Networks) could be used conditioned on meaningful attributes, in order to generate data with specific meta-data.

4.9. LIMITATIONS: THE NARROW, UNPRACTICAL FRAME OF ALGORITHMIC FAIRNESS

In this chapter, we provided an overview of the state-of-the-art computer science works that address algorithmic unfairness issues of machine learning systems. While we showed that these works focus primarily on developing definitions and metrics for unfairness, and algorithmic approaches to mitigate this unfairness in the underlying machine learn-

ing models, we also observed that there are still only few works emanating from the data management community that exploit existing data management research to approach unfairness. This led us to highlight research gaps that future data management research could fill. We then argued for a new data-centered approach. Realizing such approach would present novel data management research challenges that we described further. Finally, we also investigated and outlined the research works that have identified various types of conceptual limitations of the algorithmic fairness paradigm, and especially of proposed metrics, and of certain mitigation methods. Our proposed data-centered approach would need to acknowledge these limitations.

Next, facing the lack of research around the practices of developers who are the main stakeholders typically handling algorithmic harms, especially via the use of the technical algorithmic unfairness methods we identified, in Part II, we investigate to what extent these developers use the proposed technical methods to reach algorithmic fairness, what challenges they face in doing so, whether they are aware of the conceptual limitations of these methods, and if so, how they handle them.

Besides the limitations of algorithmic fairness research highlighted until now in terms of the lack of a data-centric approach, the literature has also discussed a number of additional practical and conceptual limitations in existing methods. Especially, the use-cases for these methods are limited, the proposed conceptualisations of fairness can oversimplify matters of discrimination, and the effectiveness and usability of mitigation methods and auditing tools are yet to be established. Researchers have further criticised that fairness mitigation employs both a techno-centric lens (as opposed to socio-technical or community-centric approach [629]) and a theoretical research lens (as opposed to a practical one [231]) on issues of discrimination in ML. As a result, it is possible to argue that algorithmic fairness tools are not yet adapted to tackle discrimination in broader terms and in practice due to this current algorithmic-centered view.

4.9.1. THE SCOPE OF ALGORITHMIC FAIRNESS

The distinction between ML and ADM. Policy documents refer to any kind of ML system when they discuss discrimination and bias (or at least they do not mention or use more specific typologies of ML techniques). Instead, most computer science research on algorithmic fairness targets automated decision-making systems (ADM) that rely on ML techniques to make decisions about individuals, or decisions that can impact individuals directly. Yet, AI-based systems might create discriminatory harms due to a variety of applications, that do not fit in the mold ADNs. ML can be used throughout digital services, for example to optimize the performance of a chatbot to improve efficiency, to test the colours on buttons to increase usability, or to recommend the film that will generate the greatest engagement from users, etc. –which are not examples of ADMs but still rely on ML. Overdorf et al. [603] for instance mention that ML and inequality are not limited to employment, income, and housing allocations. As an example, the game Pokemon Go was shown to place less Pokemons in rural areas and low-income neighbourhoods with racial minorities, creating a disparate "allocation of resources". This reflects assumptions about who has leisure time and how it is spent. This system leads to inequities, not due to automated decisions applied to individuals, but due to the optimization of the distribution of seemingly trivial digital objects.

The range of applications and domains studied. With bias and algorithmic fairness, computer scientists refer to a very specific set of problems and techniques. Such problems typically involve the allocation of resources in finance (e.g., loan application acceptance/rejection), justice (e.g., recidivism prediction for jail time/bail decisions), or hiring (e.g., selection of a candidate for a job), or the association of representational characteristics onto images or text (e.g., gender identification from facial images). Despite the plethora of fairness metrics proposed until now, it is not always the case that a metric exists for a specific ML task, and that a mitigation method has been developed for it [369]. Certain issues do not receive as much interest as the issues that directly relate to individuals. For instance, conversational AI or image captioning need a manual, tedious identification of what would serve as a protected attribute in sentences or images, instead of more automatic methods. For example, one could define the association between gender (as apparent on the image –which is questionable) and various job-related captions as problematic (e.g., systematic association of images showing women to the label "nurse" or "housewife" and of men to the label "doctor" or "chef"), which would require to identify both gender and the potentially problematic labels in relation to that.

4.9.2. SIMPLISTIC CONCEPTUALISATIONS OF FAIRNESS

Algorithmic fairness methods aim at making the outputs of a system "fair", "unbiased", "non-discriminative". With technical definitions, it means that individuals who are similar based on protected characteristics should be treated similarly by the system, i.e., should receive the same outputs. Yet, having different outputs is not necessarily what makes discrimination. Instead, it is often more the way these outputs impact differently the different individuals (potentially of a same protected characteristics) in the environment. Yet, algorithmic fairness relies on conceptualisations that cannot capture the complexity of discrimination due to the limitations of the ML set-up. Researchers develop methods centered solely around the inputs and outputs of the ML models. However, when the system is used in an actual environment, its outputs might be used differently by different stakeholders, and the actual outcomes of the system might be different for different elements of the environment. Focusing on outputs instead of outcomes cannot then accurately reflect the discrimination issues that take place. Such simplifications are necessary when taking a techno-centric approach to the problem of discrimination in order to allow for the operationalisation of bias. Yet, they leave out the real social context of the systems, and might reinforce harms more than address them in certain cases.

MODEL-CENTRIC VIEW OF DISCRIMINATION

We expect that the focus on ML model's outputs in fairness metrics is due to pragmatism. Computing fairness metrics typically requires accessing the outputs of the models (and possibly the ground truth information about data samples) and the sensitive attributes associated to each data sample. In other words, it only requires accessing the smallest set of information that is almost readily available to developers. The fairness metrics rely on checking some simple notions of parity between aggregates on this information (e.g., equal rates of getting a positive output across two groups of population corresponding to two sensitive groups), which does not require any additional contextual information. In-

stead, if one would try to check for certain inequality between aggregates (e.g., the rates of getting a positive output should be twice as high for one group than the other), they would first need to establish a value for this inequality by translating contextual information into a meaningful and mathematically relevant value –which can be a challenging task to perform. While such metrics are practical, they do not reflect the different stakeholders' desired conceptualisations of fairness. That is what we explain further in the next subsections.

Parity as the unconditional desired outcome. The metrics fail to account for applications where parity (technically defined as equality of outputs for similar individuals or groups of individuals) is not necessarily wanted for certain stakeholders [515]. Also, there is no direct, obvious mapping between the outputs of a system and the benefits it creates [553]. Instead, the benefits depend on the users, on their perceptions of the outputs in their own context [369], and on how the outputs impact them [557]. Parity might be more harmful for certain populations than others. By equalizing an error rate between groups, the disadvantaged groups for which detrimental errors are made might have less time and abilities to ask for recourse over erroneous decisions [361, 553].

Besides, equalizing an output distribution across groups does not mean that the outcomes within the groups are fair. This is why individual fairness metrics which focus on the similarities between individuals while ignoring the protected attributes have been proposed. However, these metrics are also limited, first due to the subjectivity and difficulty in defining what similarity means for various use-cases. Besides, similar individuals, even though treated similarly, might all be treated in unjustified ways [571]. For instance, all highly qualified university candidates having studied a specific field could be rejected, while the individual fairness metric would return a fair measurement. What's more, arguing for having fair models with individual fairness metrics where the similarity measure does not include protected attributes, implicitly makes the assumption that it is equally easy for different groups to obtain the same output [105]. Yet, this assumption is often wrong due to the existence of structural disadvantages.

Mutually exclusive notions of fairness. Within a single application, different stakeholders might deem as important different notions of parity. However, parity notions are shown to be mutually exclusive in the ML setup (impossibility theorems [182, 442] say that multiple fairness metrics cannot get high measures simultaneously in an ML model), due to the statistical functioning of ML models and the unavoidable inference errors it leads to make. This forces to choose to prioritize one stakeholder. When multiple metrics are considered important, due to the impossibility theorems, either the requirements of the system should be revised, or one needs to accept that it is not possible to fulfil the requirements for fair outcomes in an automated manner and the deployment of the system needs to be questioned. In the current literature, the impossibility theorem is addressed in a simplistic manner. Authors necessarily make a choice on the fairness metric to mitigate the unfairness of the model. This means that, from their vantage point, they get to determine the trade-offs with the other notions of fairness and accuracy relevant to the model. This choice unavoidably biases the model towards harmful outcomes for certain populations to the benefit of others. This decision about

the requirements and/or the non-implementation of the system should not be up to the technologists alone, especially given its societal implications. Instead, individuals or institutions who are more aware of the context in which the system will be deployed could possibly make an informed judgement.

The questionable definition of protected attributes. Most fairness metrics and subsequent mitigation methods rely on the definition of protected attributes. However, the act of defining protected attributes and the values they can take is reductive and harmful. Certain attributes cannot be reduced to a simple fixed vector as their conception might be more complex, possibly ambiguous and with multiple definitions. For instance, race attributes in existing data reflect only few aspects of the multidimensional concept of race [335]. The ways in which the values of an attribute are defined (e.g., gender as a binary concept) might ignore certain populations completely, or force individuals into non-representative values. Besides, the phenomenon an attribute is expected to reflect might not necessarily be fixed in time, location or context e.g., notions of gender or age might change over time depending on how a person identifies at different moments, and might be multidimensional in nature [694]. However, current data schema and data management infrastructures for the datasets do not support the multidimensionality and the flexibility of the concepts (e.g., once the data is collected, it is not easily modifiable anymore). In turn, when ill-defined attributes are used for unfairness assessments or mitigation, an incorrect or incomplete notion of fairness is tackled. For instance, a system might seem not to be gender biased according to one definition of the protected attribute gender, but this definition might be missing certain values (e.g., non-binary genders), which, if included, could lead to a different conclusion.

A SYSTEM'S VIEW OF DISCRIMINATION

When enlarging our view of ML models from their outputs to how these outputs are used in practice by different stakeholders, we identify a further misalignment between actual discrimination issues and their conceptualisations in computer science. This misalignment is a real obstacle to ensuring non-discrimination in practice. In many cases where parity could seem fair in theory, its realization fails to account for the whole context. The ML setup on which parity is verified is insensitive to the decisions individuals actually make based on the outputs, and to the specificities of individuals for which these systems are actually not beneficial [194, 457, 515, 557].

The misalignment between system's outcome and decisions. Contrary to the assumption that fairness metrics make, the user of an automated decision-making system does not necessarily take the decision suggested by the system's output [846]. For instance, not all judges follow the recommendations of recidivism prediction models, and not all doctors follow the diagnostics outputted by XRay-based disease classifiers, since they do not all trust the systems in similar ways. Consequently, the predictions outputted by the model might be considered unbiased according to certain metrics, but the following human decisions could be biased [731]. Conversely, claiming that a system is unfair due to biased outputs is not always adapted since the final human decisions might re-establish "fairness".

The limited impact of fairness mitigation on causes of discrimination. Equally reallocating a resource often fails to address the causes of the inequalities. It might serve as a satisfying patch for discrimination in the short-term, but it might also reinforce certain harm that cannot be formalized with fairness metrics [571]. Fazelpour and Lipton [254] take the example of college admissions in the US, where students of different sensitive groups are disproportionately represented for various reasons (historical and institutional discrimination). Unfairness mitigation methods would enforce equal admissions for all groups. However, they might reinforce existing biases such as gender stereotypes. They might identify women based on certain sub-fields they are more likely to choose and increase the number of women in these sub-fields specifically to achieve admission parity, while keeping a lower number of women in the sub-fields where they are already in minority —whereas more might apply recently.

Discrimination short of intersectionality. In order to analyse unfairness in the case of intersectional discrimination [718], researchers and practitioners employ fairness metrics. For that, the different protected attributes that form the intersectional issues are simply combined into a single attribute with which a protected and a non-protected group can be defined, e.g., gender and race would be the two axes of discrimination, which would be collapsed into a single attribute whose values indicate different permutations of gender and race in the dataset. This approach fails to address the complexity of the intersectional forms of discrimination people face in the environment of the system [361]. Intersectionality, originally developed to expose the specific ways in which the discrimination of Black women in institutions and social relations are not recognized, is not so much about belonging to a subgroup which receives different outputs or less correct outputs than other subgroups. A large body of theory and empirical studies identifies the different and complex modes of discrimination that threaten people who sit at the intersection of different oppressed groups. They expose the ways in which intersectional discrimination is produced in interactions and is socially contingent [361]. By treating intersectionality as a comparison of subgroup outputs, the complex manifestations of intersectional discrimination are flattened out, and the possibility to contest them is eliminated. By misunderstanding intersectionality as the sole membership in subgroups, fairness metrics ironically stipulate exactly that what intersectionality intends to dispute: that discrimination is one and the same for all.

The erasure of broader externalities. A system is made up of the "machine" in which the models are integrated and an environment in which this machine is deployed [400]. While fairness metrics do not account for such broader environment (except the end-users of the "machine"), the environment can also be negatively impacted. Selbst et al. [731] highlight that introducing a technology into an environment necessarily impacts the initial environment, its organisation, and possibly its values. Verifying that a system is fair with the current focus on models' outputs is then not enough, as we also need to analyse the negative impact the new system might have on the entire, original environment —this is what they term the ripple effect. Especially, the fairness metrics create "unbiased" systems for the "end-users" of the models (i.e., the inputs of the models). Doing so, they leave out other stakeholders and entities in the environment of the

systems, that can also be indirectly impacted by the models. Particularly, various negative externalities remain unconsidered in bias and fairness frameworks. For instance, routing applications might fairly route their users (e.g., each of them have similar travel time), while neglecting other issues caused by the applications such as congestion and damages on roads that are often recommended [603].

4.9.3. PRACTICAL LIMITATIONS OF ALGORITHMIC FAIRNESS

It is reasonable to ask how well, how efficiently and how effectively algorithmic fairness methods address the issues for which they were made? How usable are these tools, and how feasible are their applications in practice? In case of a gap with expectations, to what extent does this gap affect the initial objectives? We tackle these questions and show that many obstacles render the application of algorithmic fairness approaches questionable in practice.

PERFORMANCE LIMITATIONS

Even under ideal conditions where any practical issue would be resolved, algorithmic fairness methods exhibit limitations in their performance. They do not necessarily allow to reach a fully "unbiased" model, and they often come at the expense of a model's accuracy.

As we mentioned earlier, there are necessary trade-offs to be made between various metrics, not only with regard to fairness metrics but also to performance metrics. For instance, fairness through unawareness consists of removing protected attributes and their proxy attributes from a dataset. It has been shown to not achieve high accuracy performance, and high fairness for most fairness metrics [460], due to the limited information available within a dataset, and the limited control given by not having the protected attributes available. It is statistically impossible in many scenarios to have both an entirely fair and accurate model. While this can be due to incorrect datasets that do not perfectly reflect the expected outputs or the diversity of population, it is also often due to the statistical nature of ML algorithms. The way ML algorithms function imposes a trade-off between the diversity of data patterns to learn, and the complexity of the selected algorithm.

Systems might also appear unbiased in development but reveal to be biased when deployed on the new data inputted to the system in deployment. Yet, there exist no principled method to deal with such arising biases. Such biases are due to differences in data distributions between development and deployment time (data shifts), that can arise for multiple reasons. The populations on which the models are applied might simply change over time. The data engineering pipelines themselves might also differ between training and deployment due to external constraints, making the data inputted to a model different from the training ones. For example, a government might install a data capture setup to perform facial recognition, that is different from the one used to capture the training dataset, for practical, cost, or scale reasons. Besides, what the model is expected to infer might also change over time, due to changes in the ways humans think and behave (concept drift). These changes would potentially decrease the accuracy and fairness of the system's inferences [759].

PRACTICAL CHALLENGES IN SETTING UP METRICS AND MITIGATION METHODS

Fairness metrics and mitigation methods both require choosing a fairness metric, and then either applying the metric to the outputs of a model for evaluation, or applying a corresponding mitigation method for making the outputs "fairer". In practice, it is often challenging to proceed in each of these activities.

This can be due to the difficulty in anticipating potential harms, and translating contextual information about discrimination into a formal metric [369]. In many cases, the process functions backwards. The potential harms are identified after the system is deployed, and after the system has negatively impacted certain populations, and this will be reported to the service provider. For instance, Raji et al. [660] mention that transgender Uber drivers have not been able to log onto the application as the facial recognition models did not perform well for them, but this issue was not identified by bias audits.

It can also be due to the difficulties in accessing relevant data about a specific ML system, or about specific individuals to apply the metrics or mitigation methods. Auditors or even developers, either internal or external to the system's creators, might lack information about the individuals on which the system makes inferences (the inference subjects) [653], especially because the pool of inference subjects might evolve over time. Hence, it is complicated to know the kind of samples to collect or create. Besides, both auditors and developers might be in a situation where they need to collect additional data samples, which can be challenging. For example, there are naturally less data readily-available representing minority populations. The datasets, especially the ones used to train ML models, might be scraped from the Internet, which is inherently biased as certain populations have an easier access to the Internet, and have more data representing them than others. It is hence naturally more difficult to include underrepresented minorities in the datasets [216]. Certain populations might also generally not provide certain data for several reasons, e.g. overweight people might not communicate their actual weight to insurance companies in cases where they could be incriminated for it. In this case, collecting such data would be harmful to these people, as the ML models trained on this data could make inferences that disadvantage them. Paradoxically, auditing, while aiming at monitoring the fairness of a model's outcomes for unprivileged, often minority populations, raises further harms for them, since collecting more data leads to over-policing minorities and mass-surveillance.

4.9.4. THE DEPENDENCE ON SERVICE PROVIDERS

One last hurdle for performing "accurate" audits or "effective" unfairness mitigation as envisioned by the technical measures of bias is the willingness of the service providers. Since unfairness mitigation and auditing require access to data and models, only willing service providers can grant such access. The service providers could also easily perform misleading actions when auditing their system, in order to make the outputs of their systems look unbiased [718, 653]. Externally regulating the audits or verifying that unfairness mitigation has been performed is challenging, since it is close to impossible to define and collect appropriate datasets for arbitrary use-cases. The recent ban that Facebook imposed on researchers who collected data about the platform in order to study its

ad system illustrates this difficulty[5].

4.10. CONCLUSION

In this chapter, we provided an overview of the state-of-the-art computer science works that address algorithmic unfairness issues of machine learning systems. While we showed that these works focus primarily on developing definitions and metrics for unfairness, and algorithmic approaches to mitigate this unfairness in the underlying machine learning models, we also observed that there are still only few works emanating from the data management community that exploit existing data management research to approach unfairness. This led us to highlight research gaps that future data management research could fill. We then argued for a new data-centered approach. Realizing such approach would present novel data management research challenges that we described further. Finally, we also investigated and outlined the research works that have identified various types of conceptual limitations of the algorithmic fairness paradigm, and especially of proposed metrics, and of certain mitigation methods. Our proposed data-centered approach would need to acknowledge these limitations.

Next, facing the lack of research around the practices of developers who are the main stakeholders typically handling algorithmic harms, especially via the use of the technical algorithmic unfairness methods we identified, in Part II, we investigate to what extent these developers use the proposed technical methods to reach algorithmic fairness, what challenges they face in doing so, whether they are aware of the conceptual limitations of these methods, and if so, how they handle them.

[5] https://www.theverge.com/2021/8/4/22609020/facebook-bans-academic-researchers-ad-transparency-misinformation-nyu-ad-observatory-plug-in

II

Practices Towards Hazardous Failure Diagnosis

In Part I, we identified the main machine learning (ML) harms and the main research directions (ML fairness and robustness metrics and mitigation methods) proposed to solve these harms, as well as their potential limitations. We found a lack of understanding of the practices of ML developers in relation to these harms and technical solutions. Hence, according to our approach outlined in the introduction to this thesis, we now investigate the practices and attitudes of ML developers. Such understanding allows for contrasting to current research directions. This allows to characterize the research / practice gap, and to identify the most urgent research problems to solve. This is what we do in the next chapters. Specifically, respectively in Chapters 5, 6, and 7, we ask:

> ***RQ4:*** *How do ML developers debug their models for robustness issues in development? How does the research/practice gap manifest in this step of the ML lifecycle? What are the main challenges and limitations in these practices?*
> ***RQ5:*** *How do ML developers envision and tackle unfairness issues and other harms that might arise from ML models? How does the research/practice gap manifest in this step of the ML lifecycle? What are the main limitations of their practices?*
> ***RQ6:*** *What are the underlying factors that impact the attitudes and practices of ML developers, and that might represent challenges leading to the persistence of harms?*

To answer these research questions, we adopt the following grounded theory methodology. We conduct semi-structured interviews with over 50 ML developers, and analyse the resulting transcripts with inductive and deductive coding methodologies. During the interview sessions, we describe to the developers ML scenarios, and prompt them to walk us through their typical workflow to tackle these scenarios. Then, we analyse the reported workflows and their rationales, and the factors that seem to impact those workflows and rationales. We conceptualise the main goals and steps taken by the developers. We study the methods and tools they use along the workflows and whether they are aligned with propositions from research publications. We also identify limitations in the workflows and rationales, that might cause harms, based on the knowledge brought by the surveyed literature in Part I. Finally, we surface various factors that drive the workflows and limitations, for instance in terms of practical challenges developers face, often due to a lack of research supporting the design choices they have to make, or in terms of social factors, such as organisational and educational ones. The main methodological differences across the research questions revolve around the scenarios proposed to the developers, and the focus of the interviews' analysis. These scenarios respectively call more easily for a focus on robustness issues (RQ4) or on other types of issues that might cause harms (RQ5, RQ6), and the analysis either focuses on workflows (RQ4, RQ5) or on factors impacting these workflows (RQ6). Having distinct scenarios are necessary as we noticed that the developers are not necessarily able to reflect both on robustness, fairness, and other harms altogether if not prompted specifically about those concepts (and it is not possible to discuss all these within the scope of a single interview).

Chapter 5 stems from a publication at CHI'23 [68]. There, we investigate practices related to ML robustness during the development phase of a model. As we realized that debugging an ML model is a complex procedure that has not been investigated in terms of practice despite the amount of technical research proposing various solutions, we

focus on model debugging in development (once an initial data engineering pipeline and model have been developed, until the deployment of this model). Among other findings, we find that most developers do not make use of any input from research, such as explainability methods, in order to debug their models. We also identify that certain of these methods could fit the needs we identified from the interviews. We also find that they typically do not envision all types of issues (failures and bugs) their models might suffer from, despite their potential for harm, and that both education and guidance tools could potentially support them in becoming aware and tackling those issues. This is what we focus on in Part III.

Chapter 6 and Chapter 7 stem from a manuscript submitted at CHI'24 and another one accepted at AIES'23. There, we investigate specifically practices related to ML fairness and broader harms that are not covered by the fairness literature according to Part I. We especially find a large diversity of conceptions of harms and methods to tackle these harms, that developers adopt. Among those, certain conceptions (e.g., the fact that ML fairness issues only stem from dataset biases and not from flawed model design) and methods (e.g., fairness through unawareness) are flawed according to the literature in Part I. Other conceptions and methods we identify had not been mentioned in the literature as of now (e.g., the fact that certain issues can be mitigated by varying the way input data samples are collected, instead of by solely employing so-called fairness mitigation methods). We also find a number of activities developers conduct to tackle harms, that had not been mentioned in previous literature, and that are only conducted when developers reflect about harms (they are less present in reflections around robustness solely). Besides, we identify a plethora of factors that impact the conceptions and methods the developers adopt to tackle harms. Their attitudes towards harm vary, and these attitudes are especially impacted by internal human factors such as their knowledge and willingness to address the harms as well as the inherent subjectivity of the perception of harms, and by the developers' environment, their organisation, the tools they have access to, etc. These findings open up the way to a diversity of future research directions, be it technical in order to better characterize conceptions of harms and develop well-adapted methods to tackle those, design and HCI to further support the developers in the human challenges they face, or policy and regulatory efforts in order to face certain of the environmental challenges. In Part III, we develop technical and HCI tools, that we hope can better support the developers in their reflections about robustness, that might cause certain of the harms they identified in these interview sessions.

In summary, we contribute three qualitative studies about the practices of ML developers, that result in a rich characterisation of the research/practice gap in ML. While one study focuses on practices around ML robustness [68] and the other around ML fairness [61], the last one provides further information about the factors that impact these practices [71].[6] In all the studies, we find that developers have incomplete or flawed conceptions of ML failures and of the ways one can mitigate them, and that existing research does not necessarily support them in overcoming the technical challenges they face, let alone the under-explored human and contextual factors we identify. In Part III, we will tackle the development of supportive tools towards the technical challenges.

[6] We do not make any modification to the corresponding publications, except in terms of reconciliation of vocabulary across publications, and small changes to the introductions and conclusions.

5

PRACTICES FOR DIAGNOSING & MITIGATING MODEL ROBUSTNESS

5.1. INTRODUCTION

In this chapter, we investigate the practices of machine learning (ML) developers around ML model robustness, specifically for computer vision models. Deep learning models are the basis for many computer vision applications[1,2]. Yet, safely using these models is still challenging, as they suffer from issues such as spurious correlations, brittleness, and overfitting, leading to erroneous and harmful outputs [656]. Plenty of recent accidents testify of this challenge. For instance, models that distinguish between benign and malignant moles have been found to be inaccurate when used in practice for dark skin colors due to data biases [428], even though they seemed to be correctly built and perform well in the development phase.

The computer vision lifecycle is composed of many activities that might all introduce or mitigate faults in the models. While we cannot study all these activities at once, we note that growing efforts from machine learning, data management, human-computer interaction, and software engineering communities focus on proposing materials for "debugging" the failures of a model, i.e., testing the presence of potential issues, and mitigating the ones of interest, before deploying this model [293]. These materials are frameworks to test the performance of a model or to automatically mitigate inference errors [924, 152, 671, 519, 419], tools to trace issues in the outputs of the models back to problems in the code [514, 304], user-interfaces that highlight issues during model development [923, 875, 673, 729, 728, 25], and explainability methods [677, 437, 289, 69, 810]. It remains unknown how much these materials are used in practice, and to what extent they fit the hitherto unknown needs and processes of developers. It is even unclear whether the stated goal of these materials (typically increasing model accuracy)

[1] https://www.grandviewresearch.com/industry-analysis/computer-vision-market

[2] https://www.globaldata.com/media/thematic-research/global-computer-vision-market-will-reach-nearly-33-billion-2030-driven-larger-data-sets-advanced-deep-learning-models-says-globaldata/

is aligned with the goals of developers. Hence, in this paper, we focus on practices for handling failures in the first crucial phase of a model: its development phase until the decision of deploying it.

One could argue that no research on failure handling practices in computer vision models has been conducted because there already exists works around *software debugging* [34, 328, 475, 541, 179, 265, 479], and computer vision applications are a type of software. Yet, identifying and mitigating failures in computer vision models is potentially more challenging than for non-data driven software systems, due to the opaque nature of the inference process, and the unlimited set of inputs to the models [924, 488, 395, 858]. Hence, we shall study specific difficulties with computer vision failure handling. In this work, we ask: ***(RQ5.1)*** *which goals (i.e., types of failures to prevent) do developers aim at fulfilling before deploying their models?*; and ***(RQ5.2)*** *how do they proceed in terms of workflows, artifacts, and tools to do so?* These questions allow to reflect on the limitations within existing practices, on the challenges faced by developers, and on the (mis)alignment between research and practice. We perform 18 semi-structured interviews with machine learning developers having different levels of experience in computer vision, but all currently working in industry or public organisations as data scientists, data engineers, or software engineers for machine learning, for at least three years. We task them to investigate a hypothetical model to decide on deploying it or on mitigating its failures. We investigate their objectives, workflows, challenges, and needs, summarized through the questions they answer in the process. We further analyse the extent to which they use existing methods and tools, and limitations in their practices, which allows us to surface opportunities for future work.

Our results reveal that the process of making a model ready for deployment is subjective and not standardized, and that it is not a lonely process but involves various stakeholders. Our results also show that machine learning "debugging" literature is not known by most developers despite its potential usefulness for certain steps of their process. developers can identify and correct failures and bugs to a certain extent, yet pain-points and limitations, e.g., missed model bugs, are often observed. While we do not argue for standardization as the process is highly use-case dependent, our work highlights the need for more guidance and more comprehensive failure handling tools addressing various bugs (e.g., dataset content bugs) and failures (e.g., brittleness). These observations also highlight changes needed to support an education on aspects broader than machine learning algorithms, and to facilitate the communication of relevant information to the stakeholders involved in the process.

In summary, our work contributes: a) a structured understanding of computer vision model failure handling practices towards model deployment, synthesized into a framework (Figure 5.3) and a list of questions one might ask during the process (Table 5.4); b) an analysis of the relation between existing methods for failure handling such as explainability methods, and the practice of handling failures in computer vision model; and c) a critical reflection about the needs of developers highlighting several design opportunities.

5.2. RELATED WORK

In this section, we present key works on model failure handling, from which we extract the main concepts (Table 5.1), and working assumptions (summarized in Figure 5.1, and highlighted in **bold** in the text) we investigate next. We also relate our work to studies around software debugging and machine learning practices.

Table 5.1: Main concepts identified around failure handling in computer systems.

Concept	Description
Failure	The observable manifestation of an issue (difference between expected and observed behavior). [434, 27]
Bug	The cause of the issue, and hence the place where to correct for it. "Any imperfection in a machine learning item that causes a discordance between the existing and the required conditions." [924]
Artifact	Tangible information one might use in order to search for a bug or verify its validity. The approaches from literature all rely on various artifacts. [923, 875, 673, 149, 87, 152]
Precautionary attitude	The attitude that one has when performing failure handling, geared solely towards explicit failures, or also searching for less obvious failures. [34, 328, 475, 541]
Workflow	The steps taken in order to identify and mitigate a failure. [475, 541, 34]

5.2.1. FAILURES & BUGS IN MACHINE LEARNING SYSTEMS

Similarly to software engineering, in this paper, we talk about a *model failure* to designate "an external, incorrect behavior [of the model] with respect to the requirements." [434, 27], and about a *bug* or *fault* to designate the root cause of a failure. The literature on machine learning failures discusses multiple types of failures and bugs. When a script doesn't execute, the failure is due to a *program implementation* issue [924, 152, 823, 930, 808, 488]. Instead, when a script runs, according to the machine learning testing literature [924], one can observe **failures that revolve around inference outputs** (correctness, robustness, fairness) or around **processes** (security, privacy, efficiency). In this case, the failure has two possible causes reported in the literature: a *faulty configuration of the data and of the machine learning model itself,* or a *faulty translation from the intended data and model configuration to the implementation* [622, 396] (e.g., unintentionally transforming the image features that represent the inputs to the model into the wrong format).

We focus on issues of the *configuration* nature, as they are arguably challenging to handle and novel compared to software engineering, and to existing literature on model failure handling practices. Configuration issues [924, 488] relate to the design of the model architecture, i.e., the choice of architecture itself and its hyperparameters. For example, convolutional neural networks –CNNs– are often used for image classification applications; there are several CNN architectures one can choose from, each bearing different (dis)advantages depending on one's goals and constraints [433]. Other configuration issues relate to the design of the training datasets (e.g., too small dataset for the model architecture leading to overfitting, different ways of pre-processing and filtering

the data might impact differently the accuracy of a model [655]); or the choice of training procedure through which the training dataset is used to train the weights of the model architecture (e.g., a number of training "tricks" and "tweaks" can significantly improve model performance [344]). Typical terminology to designate configuration-bugs include **structural bugs** ("sub-optimal model structures such as the number of hidden layers, the number of neurons"), and **training bugs** ("the mis-conducted training process, e.g., using biased training inputs") [519, 396].

We investigate whether developers do consider these different kinds of failures and bugs, and more broadly how they judge that their process has reached a **satisfaction point** making the model ready for deployment. This is especially important to investigate because the scientific literature proposes different types of failures that can often be measured via different metrics, yet does not guide developers in choosing the eventual metric and its value under which one would consider the model failing. For instance, in terms of correctness, a model can never be completely accurate, and one needs to define in practice under which accuracy metric, threshold, and evaluation dataset, they would consider their model failing, or ready for deployment. Besides, one might account for broader information than solely metrics evaluations.

5.2.2. APPROACHES FOR FAILURE HANDLING

As we did not find any study on configuration failure handling practices for computer vision models (only studies around program failures [930], or general machine learning with end-users [258]), we focus on failure handling methods and tools. In our study, we investigate the process followed by the developers, and whether they are aware of and use tools or relevant artifacts that are similar to those proposed in the literature, as literature assumes these could potentially be useful for their processes. In case they are not used, this would bring a number of future research opportunities to understand precisely the reasons for this, e.g., unawarenesss, technical or practical inadaptability.

End-to-end methods. **Methods** are developed to support various model configuration failure handling activities. To identify failures, existing works propose methods to generate test inputs that are likely to break a model [924], or to monitor its outputs based on human-defined assertions [419]. To identify components of a system that might cause model failures, Lourenco et al. [514] develop a framework to systematically test different versions of the model training pipeline. Between the identification of failures and their bugs, Singla et al. [763] support the human exploration of training bugs: they help identify problematic model features by finding visual attributes in the data that lead to poor performance. To correct failures, Ma et al. [519] automatically identify neurons responsible for certain inference errors, and gather relevant training samples that should increase the model performance.

Tools & corresponding artifacts. A few **user-interfaces** [923, 875, 673, 149, 87] and other **tools** [152] have been introduced to support the handling of correctness failures (although not necessarily for computer vision applications). They rely on displaying or automatically checking diverse **artifacts** of a machine learning system, that might lead to a failure or bug. Around the model structure and training, UMLAUT [729] guides de-

velopers in proactively identifying failures through warnings about the choice of training and model hyperparameters, while Cockpit [728] visualises curves and statistics of the trained model, that can indicate bugs in training hyperparameters. On the dataset side, ModelTracker [25] visualizes interactive distributions of images to facilitate the identification of bugs in the data, and Deblinder [149] provides tentative explanations for each misclassification observed. Symphony [87] allows for further data and model analysis through interactions with various visual exploration components such as an interactive confusion matrix, and various functionalities to process the training data. The Amazon SageMaker Debugger [671] monitors a list of artifacts in different parts of the system design (e.g., poor initialization or too small updates for model weights, vanishing or exploding gradients, etc.) that help to reason about the existence of potential bugs.

Explainability. Within our study, we give particular attention to the realm of **explainability methods**, that we assume would be one of the prominent tools stemming from research and used in practice. Indeed, they represent a consequent amount of research papers both in machine learning and human-computer interaction conferences, and they are recurrently argued to be useful tools for handling model failures (explanations can then be seen as a type of artifact). Besides, some studies [889, 365, 371, 100, 810] discuss "debugging" and model "validation" as purposes of explainability, however almost no work [677, 67] has rigorously verified such a claim. Researchers have conducted user-studies around explanations for certain stakeholders and data types [23, 889, 409, 174, 185, 427], but none involves computer vision failures. Explainability methods can be categorized in various ways [45, 773, 503, 497], based on their scope (e.g., a local explanation [754, 302, 595, 366] explains a prediction for a single input data sample, and a global explanation [437, 289, 69] explains the overall behavior of a model), medium (e.g., visual or textual hints), audience (e.g., developers of a model, model users, decision-subjects, etc.), faithfulness (explanations are known not to be equally accurate [766]), etc. We study for what purpose and to what extent developers use explanations for failure handling, and which categories of explanations are used.

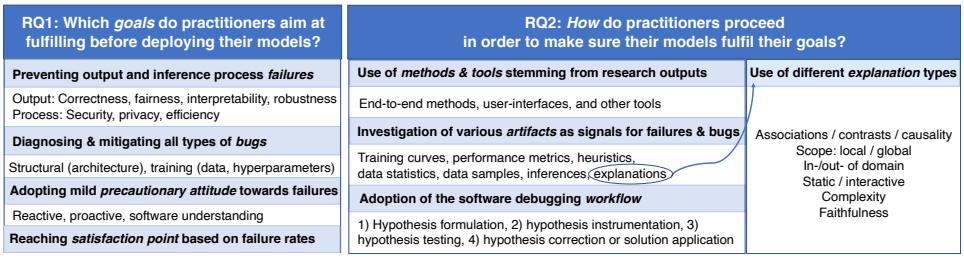


Figure 5.1: Summary of the research questions, and of the related insights from literature used as initial guides for the exploration of the research questions, and as working assumptions to assess. Each working assumption (bold text in the light blue boxes) involves one major concept of the debugging literature (in italic) and its different instances (plain text in the white boxes), and is formulated solely based on the assumptions the literature seems to implicitly make about practices.

5.2.3. STUDIES OF DEBUGGING PRACTICES

Software debugging. Software engineering literature around debugging practices provides an additional lens to analyse our interviews. In terms of debugging goals, it describes three levels of **precautionary attitude towards failures**: *reactive correction* of program implementation bugs when a failure is identified [34, 328], *proactive debugging* when developers look for the existence of bugs while no explicit failure manifests, and broader *software understanding* for later on identifying failures [475, 541]. In terms of debugging approach, this literature describes a **debugging workflow** that consists of four steps [475, 541, 34] (the usual scientific approach): 1) gathering context to generate and formulate hypotheses, 2) instrumenting and 3) testing the hypothesis, 4) correcting the initial hypothesis, or applying a solution. We investigate further whether these objectives and workflow are reflected within computer vision practices. For instance, while it is well-known that developers pay attention to explicit correctness failures through the use of accuracy metrics [83], it is not as clear whether developers might proactively investigate less visible failures, such as unknown unknowns or problematic features the model might have learned (cf. subsection 5.4.1).

Machine learning model building. Recent works [578, 365, 31, 889, 220, 498, 371, 920, 454, 637, 149, 125] investigate practices of developers in different steps of the machine learning or data science lifecycles. Yet, they primarily focus on machine learning model building, but not on failure handling. Besides our method inspired by these works, relevant discussion points are outlined, such as the types of stakeholders involved in the lifecycle [920, 371] and the challenges of the communication between them [149, 637, 454], or the complexity of evaluating models, e.g., for unfairness [220]. We investigate specifically (configuration-type) failure handling during model development, and specifically for computer vision applications, as this is a type of model, failure, and lifecycle stage that might present particular challenges and methods, that have not been investigated yet. For instance, while research has focused on the behavior of machine learning models based on tabular data [427], that can be assumed to be relatively-easy to interpret thanks to the directly interpretable features these models are trained on, it remains unclear to what extent and how the behavior of computer vision models is understood and its validity checked, as one cannot easily make sense of the model features (raw pixels).

5.3. METHODOLOGY

We conduct our study in three steps. We study literature to understand the state-of-the-art research around computer vision failure handling (section 5.2). This provides us with working assumptions related to our research questions, whose validity in practice is to evaluate. We then perform semi-structured interviews to collect practices, test the assumptions, and identify broader themes that answer our research questions. Finally, we analyse the results to synthesize a failure handling framework, and to surface limitations in practices, and research opportunities.

5.3.1. SEMI-STRUCTURED INTERVIEW PARTICIPANTS

We recruited our participants through our network and searches on professional social networks, and by snowball sampling strategy. Their experiences span a wide variety of fields, from automated diagnostics based on X-Ray images, to the automated surveillance of luggage at the airport, to applications in banking and business analytics, and automatic fraud detection with natural language processing. They have at least three years of experience within industry or public organizations, e.g., hospitals, (17 different ones in total) currently working as data scientists, data engineers, or software engineers. We made sure that they all have experience with machine learning classification tasks, for them to understand the basic concepts around model failures. In total, we recruited 18 participants (13 males, 5 females), and categorized them based on their level of experience with computer vision (CV). Low-CV experience participants (4) have developed a CV model only a few times; mid-CV experience participants (7) have less than 4 years of model development experience; and high-CV experience participants (7) have more. We span such diversity of experiences not to bias our study towards highly-experienced developers, as the level of experience is one of the factors impacting failure handling practices. Before each interview, we asked the participant for agreement on recording the interview. We then transcribed the recordings into anonymized transcripts, and destroyed the recordings. The interview process has been approved by the ethics committee of our institution. No financial compensation was given to the participants, who were intrinsically motivated to participate to our work.

5.3.2. INTERVIEW GUIDE

We performed semi-structured interviews that lasted around one hour each, and went as follows. *Step 1.* After briefly introducing our project, we enquired about the machine learning-related background of the participants. *Step 2.* Then, we presented the participants with a design brief of a failure handling scenario, and asked them to describe out loud the approach they would follow to answer the brief (**RQ5.2**), and the reasons for this method, as well as how they would decide the model is ready for deployment (**RQ5.1**). We further questioned the reasons for focusing on certain types of bugs and failures. *Step 3.* At the end, we looked back at their workflow, and questioned assumptions and gaps that had not been discussed. Especially, we questioned neglected steps of the debugging process, reasons for using failure handling tools, and explainability methods. We also showed slides with examples of model explanations (cf. Figure 5.4) to elicit further uses of explainability, e.g., saliency maps [754], SECA [69], TCAV [437]. We also prompted the participants for additional remarks, e.g., challenges they have to overcome, imaginary tools that could improve their process. The design brief and questions were finalised after performing two pilot studies. These studies informed us on how well the participants could relate to our brief and the way to present it in a concise manner, on the type of information about the machine learning model (e.g., data processing methods, previous experiments performed, etc.) the participants expect to know, and on questions useful to prompt the participants about their workflows.

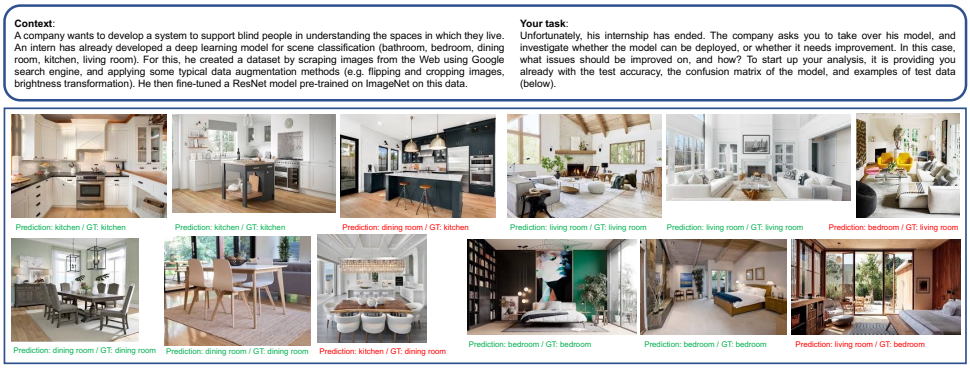


Figure 5.2: Top: our design brief, inspired by the multitude of computer vision works on scene recognition, as support for visually-impaired individuals to create mental maps of their environment [7, 321, 835, 399, 177]. Bottom: example images of four dataset classes shown to the participants, next to their ground truth (GT) and the class inferred by the model (prediction). These examples indicate feature errors in the model. For instance, among all the `kitchen` images, only the one which received an incorrect prediction contains `stools`. This hints at the potential use by the model of this concept with a higher weight than for more relevant `kitchen` features such as the `oven`.

5

DESIGN BRIEF

Our design brief (described in Figure 5.2) presents a scenario where one is developing a model, and has to decide whether it can be deployed or whether failures should first be handled[3]. Our brief is inspired from prior studies [729, 214] on the development and debugging of machine learning models, where the researchers build a simple model in which they inject various kinds of bugs, that the study participants are tasked to explore. The brief is typical and simple enough for participants to reflect on their own practices without envisioning entirely new workflows. Choosing a scene classification model allows for an easy discussion without requiring domain expertise. The brief is kept vague voluntarily to investigate what developers naturally do when asked to decide whether a model is ready for deployment, or to "debug" it for potential failures. This brief conveniently prompts for both reactive and proactive "debugging". Next to the brief, we presented the participants with a blank template (see Figure 5.5) to trigger them to think about their workflow. We also showed them example dataset images (e.g., images in Figure 5.2), along the corresponding model predictions and ground truth. We describe in the following how these images are created.

MACHINE LEARNING MODEL

The dataset images are selected with the idea of simulating explicit (low accuracy) and implicit (e.g., irrelevant model features) failures and bugs. While no prior study has focused extensively on different types of configuration failures and bugs, we select the kinds of bugs to inject into the model based on the gathered scientific literature (section 5.2), and follow proposed procedures for dataset skewing to inject these bugs. Fea-

[3] Scene classification is a common task in the computer vision literature with application to accessibility [7, 321, 835, 399, 177], although we recognize the existence of a multitude of assistive tools for visually-impaired individuals beyond vision-based techniques.

ture bugs are introduced by simulating a) potential data shifts between training and deployment data [448], and b) statistical biases in the data [69, 214, 741, 67, 437]. For a), we hint in the brief and images shown at a distribution shift between the training dataset (fancy-looking scenes, high-resolution images) that is not realistic for the target application, and the deployment data (pictures of simpler rooms taken from simple cameras). During the sessions, we only show training dataset images, but insist on the fact that they were collected from the Web (a Web query retrieves higher-resolution, professional images), and that the deployment data would come from daily-life pictures taken by the users of the system, in order to observe whether the participants reflect on the content of the datasets and the distribution shifts. For b), content biases are both around class-specific features (e.g., all `living room` images with a `television` and none of the other classes with one in training, and changing this in deployment), and less-specific features (e.g., `cats` present in all the pictures of certain classes). Other typical errors are also included, e.g., `living room` images wrongly predicted as `bedroom` all contain a bed-like `sofa`. This allows to investigate the awareness of the developers towards a diversity of issues.

5.3.3. ANALYSIS OF THE RESULTS

We analyse the results of the interviews by coding the answers in a mix of inductive and deductive thematic analysis following the process outlined by Braun and Clark [127]. We defined initial categories of codes based on the structure of the interviews, for instance the background of the participant, on our working assumptions and additional information related to the research questions that appeared during the interviews, and on our broader readings of the literature, e.g., stakeholders. Within each category, subcategories of codes are annotated inductively by identifying the response declinations relative to each interviewee (e.g., not considering structural bugs), and grouped into broader meaningful themes (e.g., limited attention towards specific bug categories). For that, the two interviewers independently coded the 10 first interviews, and discussed to reconcile the codes (e.g., choice of more or less fine-grained codes), and refine them. They then went on to re-code all the interviews, and discussed new emerging codes. Overall, we created the codes to be all-inclusive, not excluding any part of the response declinations, and mutually exclusive, as each example could not fall into two declinations of the same category. Multiple categories of codes were applied simultaneously to show the chronology and co-occurrence of process steps, goals, artifacts, and stakeholders. A total of 197 codes are identified, clustered into 30 groups, that are themselves grouped into 14 themes. The resulting codes are analysed with a focus on co-occurrence within steps, main failure handling concepts, and in relation to specific typologies of users.

5.4. RESULTS

In this section, we describe the themes resulting from our interviews, that we organize into four macro-themes (each subsection) in relation to the two research questions. We start with the goals of the participants in terms of failures and more broadly how they decide the model is ready for deployment (RQ5.1), and then describe the workflows they followed and artifacts they used to address these failures, with a specific focus on

the use of explainability methods (RQ5.2). We mark with an asterisk * the themes that (in)validate working assumptions from section 5.2.

5.4.1. RQ5.1 - GOALS: DISPARITIES IN IDENTIFIED FAILURES AND BUGS

Overall, our participants focused on a few types of machine learning failures, with various, arbitrary, subjective, qualitative judgements about their importance. Besides, they did not all choose to tackle the same instances of failures within each failure category, showing the existence of relevant sub-types that we outline below.

* FAILURES: MODEL CORRECTNESS.

As found in scientific publications, the primary focus was on correctness of the inferences, as this is the principal evaluation of the quality of the models. For instance, P3 high-CV[4] started by searching where the model makes wrong predictions *"The confusion matrix is where I start. This can give an idea of where the network might fail."* Differences appeared for the exact failures to handle. Most participants focused on high-rate failures (P10 low-CV) *"I'm looking at this confusion matrix and think about which class is the most error-prone."* Instead, two experienced participants started with rare issues as these pinpoint hard challenges for the model, and solving these issues could solve the high-rate ones (P16 high-CV) *"I look at the rarest events, where the most information lies. It is handy because you can analyze everything going through the images."* A last participant saw both frequent and rare issues as fundamental (P17 mid-CV) *"I focus on the extremes, the very good ones and the very bad ones. It helps me to find features of interest."*

* FAILURES: OTHER FAILURES.

Other types of output or process failures (e.g., model robustness to natural perturbations or adversarial attacks, privacy, unfairness, unknown unknowns), although discussed in the literature, were mentioned by just a few developers. For instance, only two high-CV participants were concerned with the robustness of the model to natural perturbations, i.e., distribution shifts occurring unintentionally in the data [448] (e.g., the brightness of the training images is much higher than the one of deployment images, where users of the system might not be able to ensure a level of brightness for the pictures they take) (P4 high-CV) *"I will find another dataset to check the model performance again. These images are always very bright. But this might not be the case in practice. It could be like using the phone to take the images. Also, if the weather was cloudy, the images would be very dark."*, (P9 high-CV) *"The data in deployment (houses of people) may be different from the ones in your training dataset, probably from catalogs. So I would not expect the model to work well."* Some failures were also considered without explicit naming with the "technical" term, such as for unfairness discussed in the following terms by P13 high-CV *"What is called the dining room and what is called the kitchen is person and culture dependent. So, whether a prediction is wrong, that is heavily dependent on what use-case we are talking about"*[5].

[4] We denote participants by "PX k-CV" with X the index of the participant and k the level of experience of the participant with computer vision.

[5] While most examples of algorithmic unfairness from the outputs of a machine learning model consider disparities between errors rates for different categories of populations [848], other works [216, 859, 736] have

Other developers did not envision the existence of these failures, e.g., only six participants were concerned with unknown unknowns that can be seen as a subset of correctness failures (data samples for which a model makes wrong predictions while displaying a high confidence, hence particularly challenging to identify in production) [506, 938, 52]. A last set of developers considered them unimportant (e.g., several developers mentioned not caring for distribution shifts as they would anyway try to obtain a "representative" training dataset); or irrelevant for this use-case (e.g., P8 high-CV said unfairness issues are not a concern, yet this is questionable as one could imagine that the different scenes the model should recognize would look different in different parts of the world [736]).

FAILURES: MODEL FEATURES.

While this is absent from the machine learning testing literature, some participants were also concerned with the meaningfulness of the features learned by the model. They identified feature failures by scrutinizing specific samples (see subsection 5.4.3) (P4 high-CV) *"The overall test accuracy is 80%. This accuracy for the initial model is fine. Next, I use a visualization method like T-SNE to see if this model truly learned something."* They talked about failures when the model did not seem to have learned any relevant feature looking at the overall shape of a few saliency maps, or when the model did not display specific, expected features for specific samples (e.g., the model classifies correctly an image as a `kitchen`, but does not seem to use the presence of a `fridge` or `oven` for that, while a human would have looked at these elements). This shows the duality of model features, seen either as goals here or as means to explain and solve correctness failures (see subsection 5.4.4).

Other participants explained not knowing or recalling that the model can reach correct inferences using questionable features. They would however handle the features after the correctness failures considered more urgent (P14 high-CV) *"That's a second step. I focus at the beginning on the errors. When I understand globally why and how, then I go through the correct answers. And I investigate if the model understood the classes."* A few participants also never handle feature failures, arguing that handling correctness failures automatically solves the relevant feature issues. They first evaluate the model with new samples representative of the deployment data, and if the error rates are higher there, the model might use wrong features. Otherwise, irrelevant features are not considered errors: while not relevant for humans, they are acceptable as the model makes correct inferences. This approach does not always hold depending on the use-case requirements, and the feasibility of collecting a representative dataset (e.g., due to contractual or privacy issues).

* PRECAUTIONARY ATTITUDE: DIFFERENT ATTITUDES ACROSS LEVELS OF EXPERIENCE

We note a disparity between participants in their level of precaution towards failures. Participants with low-CV experience spent more time on *general understanding* as they did not know where to focus. Later, they focused on *reactive* debugging (explicit correct-

considered broader algorithmic harms, where the model would not perform equally well on a same type of object or scene that presents different representations across geographical locations or cultures. Hence, we (and a few of the developers) consider potential unfairness issues in our scenario.

ness failure) when choosing specific correctness errors. Proactive debugging as a workflow objective, i.e., the idea of searching proactively for non-obvious model failures, such as the use of wrong features by the model, was not a familiar concept to the participants, who did not envision the existence of such implicit failures. Proactive debugging is especially useful given that the distribution shifts cannot lead to explicit failures when one evaluates their model on an evaluation dataset taken from the same data distribution as the training dataset: one could proactively reflect on the eventual distribution shift and the type of additional training data that could be needed to solve it. Participants with mid-CV experience focused primarily on obvious manifestations of correctness failures, and experts discussed all goals. However, 75% of these mid- and high- experience participants only discussed proactive debugging when prompted. This disparity is concerning considering that our design brief was implying a strong distribution shift (the fact that the training data were collected from the Web but the deployment data would be pictures taken by visually-impaired individuals in everyday environments) calling for proactive debugging.

* BUGS: REFINEMENT OF BUG CATEGORIZATIONS

Overall, the bugs addressed by developers were both structural and training ones. Yet, similarly to failures, we observed differences in the bugs identified by developers of different expertise, differences that we discuss further when explaining the specific failure handling workflows.

Coding the interviews, especially the goals of the participants, and the explanations they were providing for identified failures, led us to propose a more fine-grained categorization of these latter bugs. We distinguish between *dataset bugs* further sub-divided into *data-statistics bugs* (e.g. distribution of data samples across classes) and *data-content bugs* (e.g. distribution of specific visual elements appearing across samples and classes), *data-engineering bugs* (e.g. how the data samples are scaled, filtered, augmented, labeled, etc.), and *training-parameter bugs* (e.g. loss function, batch size, etc.). This distinction should allow developers to be more structured in their reasoning about bugs, but is also useful for researchers to develop bug-specific debugging methods. For instance, to the best of our knowledge, data engineering bugs are not discussed in the machine learning literature[6] while addressing them early could avoid retraining models.

5.4.2. RQ5.1 - GOALS: DISAGREEMENT ON THE SATISFACTION POINT FOR DEPLOYMENT

While the participants were focusing on diverse types of failures along their process, we explicitly asked them to clarify how they would judge the model ready for deployment. We discuss their process here.

AMBIGUITY

The point of satisfaction at which the participants stop their process was ambiguous.

[6] Possibly because data engineering typically belongs to the data management literature, inadequately disconnected [66, 304] from the machine learning one.

Trade-offs between failures. Along their process, the participants mentioned various types of failures with minimum requirements on the absence of certain failures (e.g., overfitting was unacceptable for P8 high-CV), and needed trade-offs across the different categories. For instance, P13 mid-CV did not consider meaningful features (feature failures) as important as long as the accuracy is high (correctness failures) *"The accuracy is what counts the most for lots of my projects. If something hits 99.9% accuracy, I don't look at the saliency maps anymore."* (P10 low-CV) *"We cannot even interpret how our brain works. So why we are so focused on interpreting how the model works?"* These trade-offs were also made for specific instances of failures within a category, as discussed in subsection 5.4.1. Yet, none of the participants expressed a precise way to judge how severe each failure is, and to establish when the trade-offs are acceptable.

*** Disconnect between failures and metrics.** The participants also based their decision on the values of certain correctness-related performance metrics. A direct mapping between such metrics and the failures implicitly appeared from the low-CV participants, as they considered correcting failures as the mean to their goal (increasing performance metrics). Instead, for participants with more expertise, the relation between failures and metrics was perceived as less clear.

Expert participants were cognizant of the limitations of using metrics, and used them as a preliminary indication of the model's quality, before observing inferences on individual samples. This was the case a) when the test dataset is erroneous (e.g., wrong label) or ambiguous leading to over- or under-estimating the model *"If you're talking about hard labels, there is an error. But if I understand why the network classifies this kitchen as a dining room, I no longer consider it an error."* P3 high-CV; b) when a mistake could also be made by a human (P7 mid-CV) *"it is confusing even for humans to classify these images. So I tolerate some error."*; c) when the mistake is rare (P14 high-CV) *"it's not a fundamental but understandable mistake. I will be OK with it. This kind of bathroom, there are one out of 1,000,000."*; d) when the error has a high confidence (a few expert participants used the model confidence to judge an error's gravity (P14 high-CV) *"I check the probabilities that the model gives to see if it's really wrong or a bit wrong. If it's 60% dining room and 39% kitchen, then I say OK."*; or e) in cases when an expert would judge the error acceptable[7].

* VARIABILITY IN CHOICES AROUND METRICS

The way correctness metrics and the threshold of acceptability were selected greatly varied across participants. Some participants made an intuitive choice (P11 low-CV) *"My goal is to have as much accuracy that I can get."* Or they deferred the choice to domain experts or model requesters, judged more qualified or responsible (P14 high-CV) *"What would the business be happy with? As a system that they would put into production, there is a definition of good enough."*

Others emphasized that errors are not avoidable, and adopted a nuanced, class-based evaluation, accepting errors on certain classes to balance correctness for other classes (P7 mid-CV) *"One cannot be perfect in all cases. Let's say you are more interested in classifying images about kitchens. If you confuse the dining room with the living room,*

[7] a) to d) can be questionable when the model has high-stakes.

then you are okay. Then, you reach high recall classifying kitchens. You would be satisfied." Two thirds of these participants recognized that different use-cases require emphasis on different metrics (P10 high-CV) *"It depends on the application. If I want as many kitchens as possible, then recall is more important. But for autonomous cars, recall is not as important as precision."* As for the choice of threshold, some developers proposed absolute numbers based on the characteristics of the task and their background knowledge (P4 high-CV) *"The accuracy should be higher than 95% because this model is for the blind people so safety is the top priority."* Others chose based on the performance of existing baseline models (P9 high-CV) *"I don't know how hard this task is, so I don't know what accuracies can be considered acceptable."*, or on human disagreement (P7 mid-CV) *"When you know whether people would agree, you know the human accuracy. Then, you would not beat yourself up if your model doesn't reach an accuracy higher than the human one"*.

5.4.3. RQ5.2 - PROCESS: DRAWING THE FAILURE HANDLING WORKFLOW

* A WORKFLOW SIMPLER THAN FOR TRADITIONAL SOFTWARE SYSTEMS

The participants followed a trial-and-error workflow similar to the one for debugging traditional software systems. However, they often simplified the workflow, and typically did not test their hypotheses rigorously before acting, or even did not formulate specific hypotheses before experimenting on different models. As the software debugging literature does not directly apply to each step of the workflow within the machine learning context, in the following subsections, we describe further how our participants conducted each step —when they did conduct it— (we detail bug correction strategies in Appendix).

IDENTIFYING A MODEL FAILURE

Depending on their type of precautionary attitude, participants did not adopt the same approach to start tackling a failure. Reactive debugging starts by exploring the confusion matrix and identifying areas with low or high error rates (subsection 5.4.1) (P3 high-CV) *"The confusion matrix is where I start from. [..] Also regarding class overlap, I would expect that classes that are closer, are also closer together in the network embedding space, and that it would lead to increased errors."* Then, the workflows described next are employed.

Proactive debugging follows the same workflows, the difference being that the failure first needs to be detected. Participants interested in feature failures scrutinized the features through saliency maps to reflect on their validity. To find failures due to distribution shifts, they compared the training dataset to imaginary deployment data ((P9 high-CV) *"The domain of the dataset where you train the model can be distant from the house the blind person enters, so I'm not sure if solving the current model issues would solve the problem of the blind person."*), or when feasible searched for more diverse images, to identify potential limitations in what the model learned. Often, the participants did not purposefully identify these implicit failures. They discovered them serendipitously during reactive debugging, when scrutinizing samples or features with incorrect predictions.

GATHERING CONTEXT AND FORMULATING HYPOTHESES FOR NON-DATA BUGS

Overall, the participants tackled the gathering of context and the formulation of hypotheses around bugs differently based on their experience with computer vision.

Skewed sets of envisioned bugs. Experts participants took a sequential, bug-elimination approach. They always started with structural and data-statistics bugs, later on turning to data-engineering or training-parameter bugs, and to dataset-content bugs as a last resort. They took this approach for practical reasons. (P8 high-CV) *"Looking at the images is the last step. If the training is poor, there are things you can do before. For example, dining room and kitchen might share many pieces of furniture and because of that, it's harder to distinguish between them. This, I can assume without looking at the pictures, from prior knowledge."* They also assumed structural bugs to be limiting factors for a model (P14 high-CV) *"When I reach some performance [with experimentations on the model], the main problem is not in the architecture: the model is learning but in a bad way. Then, I check the augmentation of images, or try other datasets."*

In the rest of this subsubsection, we describe the way these high-CV participants investigated the first batch of bugs (non-data bugs). Less-expert participants took a less structured approach, and focused on the bugs they were most familiar with, essentially dataset ones (described in the next subsection) (P6 low-CV) *"hopefully if it has stronger data, it can learn something deeper. And if not, the model itself should change, but I'm not so familiar with CV and how you can improve it from the model perspective."* They sometimes wrongly assumed that mitigating dataset bugs can serve to correct all failures forgetting to account for the bias-variance trade-off, e.g., if more training data is added, the model hyperparameters might not be adapted to the dataset anymore, leading to underfitting (P5 low-CV) *"My first step would be to pick one angle: either the data (because the model performs only as good as the data it was trained on), or the system parameters (some learning rate or model hyperparameters)."*

Truncated and oriented context and hypotheses. To deal with structural and training-parameter bugs, expert participants tried multiple models with different architectures, training hyperparameters, and data processing (P3 high-CV) *"Going a step back, I would employ augmentation techniques to see if I can get higher performance, and I would use a method to further regularize the model to make sure that it's not falling into the overfitting regime."*, (P3 high-CV) *"I suppose that the input has been sufficiently preprocessed? I would normalize, typically by the max value if we are talking about standard RGB images. I would also standardize the data, so force inputs to have zero mean and unit variance."* until they reached the "best" model among these tests (P14 high-CV) *"There is something that I do dumbly at the beginning: I try different architectures to see if there is a problem of this kind. I'm not sure that the architecture is the main issue. But it can help to add more dropout, or change the architecture, especially when I have a problem of overfitting."* Developers have learned through experience typical "good" hyperparameters that they test in priority (P3 high-CV) *"One thing that could lead to increase performance is to force those classes to be more separated by employing another form of loss, like the contrastive loss."* This process truncates the software debugging workflow as it directly consists in testing various potential "solutions" to improve the model performance, solely with an implicit hypothesis (non-data bugs: the model hyperparameters have not been explored) and no gathering of context for hypothesis formulation.

*** Supporting artifacts.** During this process, participants mentioned monitoring a subset of the artifacts discussed in the literature such as learning curves, and overall shapes of saliency maps that might indicate model overfitting, to orient further the search of the "best" model (P3 high-CV) *"I will see some training curves. The optimal case would be that the further the training process is, the lower the training and validation losses go. This means that the model is learning something without sign of overfitting."*, (P9 high-CV) *"I would see how the training curves look like with the Tensorboard, to see if the model is overfitting on the training set. If that's the case, you can add some regularization or augment the training set."* We did not delve deeper into these bugs during the interviews, as only expert participants discussed them, and existing research primarily provides support with similar artifacts for these bugs.

GATHERING CONTEXT AND FORMULATING HYPOTHESES FOR DATA BUGS

Artifacts as context. Data bugs were typically connected to correctness, robustness, or feature failures. They were specified by investigating test set images and/or saliency maps for recurring visual elements the model might have learned as features, rare visual elements that might confuse the model, or signs of problematic data processing (image size, resolution, unrealistic data augmentation). The link to the activities that led to such bugs was then made, and bug correction strategies were devised. For that, participants used different sets of images. a) The images corresponding to a confusion matrix cell (P14 high-CV) *"There are a lot of false positives of dining room and kitchen. Let's see in the images what kind of situations cause these mistakes. I would plot heatmaps. Probably it would put the salient part here, and that's the problem."* b) The images that received correct inferences for the classes at stake, searching for common concepts with the wrongly predicted images (P7 mid-CV) *"I focus on cases where the model made a mistake and the ones where the model is correct. I figure out the pattern that was correctly detected."* c) One participant looked at a random sampling of images of a class to understand how diverse the dataset is, and compared it to mis-classified images of the class (P17 mid-CV) *"My goal is to understand how diverse are the images of kitchen visually and how well they capture the essence of a kitchen. There might be some similarity metrics to use."*

Diversity of hypotheses. Participants formulated five types of hypothesis (cf. Table 5.2) around model features and data content, using the above artifacts and their background knowledge (P2 high-CV) *"I would compare a true positive and a false positive from these classes, apply some domain knowledge, and see if there are elements which should be used for a specific class."* The first one was however not formulated by participants with low-CV experience as they did not think features can be wrong, or did not know how to identify features. For all these hypotheses, the notion of granularity is important, i.e., different levels of description of the visual elements a model has learned. For instance, the participants often mentioned the style of an object the model is expected to use for classifying an image (P14 high-CV) *"I make an assumption by trying to understand why it makes these mistakes. This bed is not classic, so maybe the dataset needs more not-classic beds."*, parts of an object, and remarkable textures and colors of these objects.

Table 5.2: The diverse hypotheses formulated by the participants around model features and data content.

Participants' hypothesis	Explanation
Irrelevant features	(P1 mid-CV) *"The model might learn wrong rules, like the presence of a sink to predict a living room"*, (P7 mid-CV) *"Once we know the wrong patterns the model learned, we add more examples that reflect the wrong behavior in the training data for the model to learn the extreme cases."*
Incomplete features	The model has not learned enough features to correctly make inferences for certain images. Incomplete and irrelevant features are always mapped to dataset biases (P17 mid-CV) *"What comes into my mind is rules, but it will defeat the purpose of having machine learning. I model what's a kitchen in a symbolic fashion like "needs an oven, stove". And then I make sure that the data set is reflecting those adequately."*
Over- or under-emphasized features	(P7 mid-CV) *"The first step is to use an interpretability method to detect what the model has learned. For example, when the model classifies kitchens, it does not look for a sink or cooking stove. It looks for under-relevant patterns like tables that can be used for other classifications like dining rooms."*
Unknown unknowns	Three participants related the incorrect or incomplete features to unknown unknowns (P7 mid-CV) *"knowing what to expect from the model and what it learns allows to identify unknown unknowns"*, (P12 high-CV) *"A blind-spot happens because of systematic data biases. You have to see how the data distribution looks like to figure out whether there is a blind spot. You should use crowdsourcing because automatic methods are not reliable."*
Absence (presence) of (ir)relevant elements in images	This makes the model confuse the ground truth for another class (e.g., the lack of a bed in a bedroom makes it being classified as a kitchen) (P7 mid-CV) *"This image is missing hot spots."*

INSTRUMENTING THE HYPOTHESIS

Most participants did not instrument and test their hypotheses. Instead, other proxy methods were employed when feasible.

- *Artifacts for hypothesis invalidation.* Between the observation of a failure (e.g., false negatives for a certain class) and the identification of its potential causes i.e., the bugs (e.g., overfitting on other classes) and remedy (e.g., decreasing the number of layers), participants often used intermediate artifacts (e.g., training curves, data statistics) for context gathering. These artifacts were serving both to search for the potential bug, and to quickly check that no other information about the model would invalidate their hypotheses.

- *Correction as instrumentation.* Instrumenting the hypotheses was often about making a correction and checking for a positive change in the model, followed by further fine-tuning the correction (see subsection 5.4.3).

- *Hypothesis testing.* Only three participants tested their hypotheses with other instruments, even though it is probably more efficient than retraining a model for each hypothesis. They searched for data samples or transformed available samples to present only the features (or anything but the features) of interest, and check whether the inference of the model matches expectations (P17 mid-CV) *"I take a perturbation approach. Once you see commonalities, let's say "white", you mask out the non-white*

thing, and see if the probability is increasing. If so, I may be looking in the right direction and need more non-white kitchens." Such activity needs more support as participants argued it is challenging.

5.4.4. RQ5.2 - PROCESS: EXPLAINABILITY FOR FAILURE HANDLING

* NARROW SUBSET OF EXPLANATIONS

Our participants typically did not mention any tool or method inspired from the ones we identified in the literature. Our participants only mentioned using saliency maps among other explainability methods, except P4 high-CV who also mentioned T-SNE [837] for faster image exploration through image clustering. A few participants without experience with explainability described the desire to have explanations that correspond to saliency maps, without being aware of their existence. A few participants wished for other types of explanations. For instance, they would like to automatically obtain statistical summaries of visual elements across images to fasten their hypothesis formulation and validation process (P6 low-CV) *"I want to see the entire distribution of objects, and subdivide these 25 mislabeled dining rooms into smaller segments that I can understand, like photographs of dining rooms with the kitchen in the background."*. They also insisted on getting textual explanations besides visual ones to query whether the model has learned expected or known problematic features (e.g., a participant mentioned that the models shouldn't pick up on potential pace-makers), or to quickly explore the training data distribution.

DIVERSE PURPOSES FOR EXPLAINABILITY

From the interviews, we also found out that the use of explainability methods is not standardized. The purpose for and way of using the saliency maps (the primary explainability method that was employed) varied across participants. Overall, we identify four uses; non-expert participants only focusing on the first one.

- *Artifact for data content or data engineering bugs*: Saliency maps were used to identify problematic features, and to further investigate potential solutions for correctness failures. This was done by scrutinizing the image patches highlighted by saliency maps, and reflecting on the points in Table 5.2. Certain participants disagreed that it is feasible to look into the actual data content because it is hard to define what one would expect a model to pick-up on (P4 high-CV) *"In a bathroom you expect the bath to be highlighted. You expect the dining room table in the dining room, but in the kitchen there can also be a table, so it's not convenient."*

- *Artifact for bias-variance trade-off*: Saliency maps were used to make sure the model learned something meaningful, and is not over- or under-fitting. For that, participants analysed the shapes of the maps across images, and their coverage of pixels reflecting human-interpretable concepts (P4 high-CV) *"I first see if this model truly learned something (the objects, not some nonsense). Saliency maps are really tiny: it over-trains. It's about the general aspect of the map, more than what it's highlighting."* This was used by expert participants who have formed over time an idea of a meaningful saliency map, and how it relates to model failures (e.g., overfitting).

- *Final verification*: Certain participants used saliency maps as a last step to quickly validate the relevance (P15 high-CV) *"I first fix my model, then my data. Once I'm sure this is the model I'm going to use, I check that images are analyzed fairly according to what we expect. I see the actual visual clues that the computer bases its decisions on."* and possibly fairness of the model features in a random subset of saliency maps (P9 high-CV) *"It is very important if you're afraid the model is biased towards categories with ethical implications."*

- *Stakeholder communication*: Most participants used saliency maps for communicating about the model performance (P9 high-CV) *"You measure the success from the accuracy. If successful, you understand what the model is looking at with explainability. It is nice to explain to your clients why the model works and what it looks."*

5.5. DISCUSSION & IMPLICATIONS AROUND THE RESEARCH / PRACTICE GAP

Our interviews brought new insights into computer vision model failure handling practices (summarized in Table 5.3), that are corroborated by the few HCI studies that compare non-machine learning practitioners with machine learning experts [906]. Instead of relying on the (potentially useful) theory, methods, and tools published in literature, the developers in our study develop an error-prone workflow based on their prior experiences with machine learning, and they do not systematically address every machine learning failure and bug. This is concerning as other stakeholders within an organization might also not be aware of and in charge of these failures. We now discuss implications of these results for future research.

5.5.1. SURFACED DESIGN DIRECTIONS

Our results led to identify obstacles for developers to correctly handle failures. These obstacles can serve as design principles or challenges to further support developers. In relation to these, we discuss a few avenues for future work.

1. *Challenging need for workflow diversity.* Failure handling requires diverse workflows, as it is a highly use-case dependent task (use-case, stakeholders, structure of an organization and allocation of responsibilities, etc.), and no one-size-fits-all process has been developed. Hence, we do not argue for standardization, but emphasize the need for a plurality of workflows, that brings about new research challenges to create supportive methods and tools.

2. *Confusing fluidity of concepts.* One surprising insight was the fluidity of the concepts in the participants' workflows. While we had envisioned identifying independent sets of failures, bugs, artifacts, and steps, related by how one serves to identify or solve the other, we realized these sets are permeable. For instance, features can either be considered failures when they are irrelevant or incomplete according to human judgement, or an artifact to identify the dataset bugs that caused correctness or robustness failures (same observation for overfitting). Bug correction was also either the actual

Table 5.3: Summary of the insights obtained through our study.

Category	Insight
RQ5.1: Stated and verified goals of the failure handling process.	
*Failures	Failure handling practices for computer vision models often focus on a narrow set of failures (compared to literature), centered on output correctness, with however model problematic features as an additional, typically understudied, failure.
*Bugs	Developers address the same bugs as discussed in the literature, with more refined bug categorizations (structural, dataset, data-engineering, training).
Satisfaction point	Ambiguous decision boundary, made of trade-offs between various failures and correctness metrics, to declare the model ready for deployment.
Differences across developers	Participants have disparate knowledge about "debugging" concepts, and limited attention towards different bugs: sequential bug-elimination approach for high-CV participants, incorrect trade-offs between bugs for low-CV participants. They also show disagreement on the importance of correctness and feature failures, and disparate precautionary attitude.
RQ5.2: Failure handling process.	
*Workflow	An ad-hoc, trial-and-error workflow that is simpler than for traditional software system debugging is adopted. Typically hypothesis instrumentation is missing, as well as hypothesis formulation for non-data bugs. Its steps are based on developers' experiences.
Hypotheses	Hypotheses related to data bugs are around problematic features: incorrect or incomplete features, over- or under-emphasized feature importance, absence/presence of ir/relevant visual elements in images.
Corrections	Various correction methods: modifications of dataset, training parameters, model structure, and way the model is setup.
*Artifacts	Next to known model artifacts, primarily visual content across images is used. Need for domain knowledge is polemical.
*Methods & tools	None of the methods or tools developed in the literature are used. Only TensorBoard [152] has been mentioned.
RQ2: Use of explainability methods for failure handling purposes.	
Purposes	Diversity in purposes: scrutinizing dataset bugs, bias-variance trade-off, stakeholder communication, and final verification.
*Types	A narrow subset of explanation types (saliency maps) is used in practice. Wishes for global, textual, query-able explanations about the model and potentially the data are unfulfilled.

correction step taken by the practitioners, or one way to test their hypothesis. Concept fluidity is already known for certain non-functional, trustworthiness-related, requirements of machine learning systems, such as fairness [571], interpretability [427], and contestability [517]. This fluidity brings confusion to the research and practice, and should be acknowledged, e.g., to clarify the available tools and steps, and to reassure developers about their process. One can take inspiration from these other works to handle the fluidity of the failure handling concepts, for instance by proposing a comprehensive overview of the different uses of the terms by different developers and research communities (e.g., also highlighting the dissimilarities with traditional software engineering), as a boundary negotiation object [571].

3. *All developers are not equal in confidence and effectiveness.* Low-CV participants lacked a clear workflow, spending a large part of the interview on model understanding, instead of reactive or proactive debugging. A few of these participants expressed not being confident in their process, discussing a (P5 mid-CV) *"very empirical process" that "reflects a human feeling of what's going on".* They posed that this way *"the suc-*

cess of debugging is left to the sensitivity of the expert". Participants with more experience were instead more confident, faster, and effective. This result displays similarities with the way people working on non data-driven software develop an ability to debug their software, with experts learning debugging heuristics, the effective use and application of debugging tools, etc. [542]. The development of new tools should hence bear in mind the various levels of AI literacy of the developers and their confidence. AI literacy literature [164] refers to four literacy dimensions (technology, work, learning, and human-machine -related dimensions) that should all be considered to tailor the tools to their users.

4. *Difficulties in using new tools.* The participants had difficulties envisioning uses of new tools. When we showed low-CV participants saliency maps or global explanations, they could not envision how to employ them. Similarly, when showing more experienced participants explanations they were not familiar with (global, textual explanations outputted by the SECA method [69]), only half of them could envision using them.

Besides, Liao et al. [497] built an explainable AI question bank where each question reflects a need for explainability. Inspired by this bank, we built a failure handling question bank for computer vision models in Table 5.4, that summarizes the information needs developers might have when handling failures. For that, we reviewed the transcripts and workflows described by our participants, and extracted their explicit questions and questions that were implicitly answered by the actions they took. Compared to the XAI question bank, we added new categories of questions, revolving around the algorithm design and the way the model was trained, around iterations of the model, and expectations on the model behavior (reflecting the need for domain knowledge). These questions revealed to be essential to tackle structural and training bugs, and to understand when the model is satisfactory. We also refined the questions about model features, their nature, relevance, completeness, as features were an essential artifact to judge the validity of the model and to identify correction methods. The question bank can be used by developers as inspiration to identify the relevant questions (and whether methods for getting answers exist) to ask for handling failures in their model, and by researchers to identify important research directions that have not been tackled until now.

5.5.2. OPPORTUNITIES FOR THE DESIGN OF NEW SUPPORTIVE TOOLS

NEED FOR GUIDANCE

We argue that developers need more guidance on the process. Proposing high-level (sequences of) steps and intermediate objectives for structuring the workflows in relation to different failures, associated artifacts, examples of bug correction methods and pitfalls, would allow for a more effective and efficient process. The exact form of this guidance requires further investigation, e.g., a tutorial, a checklist, an interactive framework, a tool suggesting a workflow and artifacts, etc. Previous works around software debugging and machine teaching provide hints for its design, highlighting the importance of *structured steps* [622, 590]; structured *documentation* [283, 556, 31, 346, 125]; or *warnings against graphical user-interfaces* [906]. Research is also needed to balance this guidance with

Table 5.4: Questions developers ask when handling failures of models. In bold the ones also found in the XAI question bank [497], and with a triangle △ the ones that have not received extensive attention in terms of study of practices or technical solutions. Questions without a triangle are formatted in italic when they can be (partially) answered using existing explainability methods, the others being answered using other debugging artifacts.

Topic	Question
Input	**What kind of data does the system learn from?** (and all related questions of the XAI question bank) *To what extent is the data diverse enough to represent each class?* To what extent is it balanced over the different classes? △Does the test dataset cover the complete range of situations the model can encounter in deployment? What do the samples look like for each class? *What are the difference between these two classes?* How have the data been processed? and augmented? Is it easy to augment the dataset by collecting new data?
Model performance	**How well does the model perform generally? Where does the model typically make errors? for what type of images? into which classes does it incorrectly classify them?** Does the model make errors with high or low confidence? △Are there unknown unknowns?
Expectations	△What is the expected performance for the model? for which metrics? Can we consider the model to be fair and unbiased? △Is this inference really incorrect? or can we accept it? What should the model pick up on to distinguish these two classes?
Model structure	**What is the structure of the model? How were the parameters set?** How was the model trained? What loss function was used? what were the training hyperparameters?
Model training	*Is the model overfitting or underfitting? Is the model too large/small for the task? compared to the training data?* Is the training dataset of the pretrained model relevant for the target task? Does the performance improve when simply adding training samples?
Features (global - how)	*Has the model learned anything relevant?* ***Does the model use (or not use) this feature?*** *Which visual elements does the model use to predict this class?* ***Which visual elements does the model generally use?*** *Which visual elements does the model use to make (in)correct inferences?*
Features (local)	***What features of this instance lead to this inference?*** ***Why is this sample predicted P instead of Q?*** *Which visual elements might have triggered this wrong inference?*
Features (comparison)	*What are the features used for both classes? What are the features different for the two classes?* ***Why are instances A and B given the same/different predictions?*** ***What are the top features/rules used by the model?*** **How does the model weigh different features?**
Questionable features	*Are these visual elements relevant for this sample? or class? What features do we expect it to learn for this class?* △Should the model pick up on more visual elements for this image/class? △Should it learn additional features? *Does the model make correct inferences using wrong features?* Are the features fair to use?
Inferences (what if)	***What would the model predict if this sample is changed to ...?*** *What would the model output for a sample with these visual elements?*
Iterations	△How to improve the model? △Should I focus on the data or algorithm and training hyperparameters? How well does the model perform after doing X? *Have the features changed after doing X?*

the freedom developers need for failure handling, and to leave the flexibility to envision usages of new artifacts.

There is no comprehensive resource accessible by developers to learn about failure handling. We suggest the community to build an open, collaborative repository of practices to share heuristics (similarly to UMLAUT [729]), methods and tools, as well as the-

oretical knowledge[8] (e.g., list of failures, bugs, relations to artifacts). Such library should provide both general information, and information that is specific to certain types of use-cases, models, etc., since the participants regularly referred back to previous use-cases they encountered with similar considerations. Research on software debugging again provides recommendations for the design of such library, with lists of relevant information to include—e.g., patterns [549], debugging diaries [633]—, and methods to collect this information [542]. As a first step towards establishing such a library, we propose a failure handling framework (Figure 5.3) designed by synthesising our participants' practices. It summarizes the various objectives, main steps, and artifacts of the failure handling process.

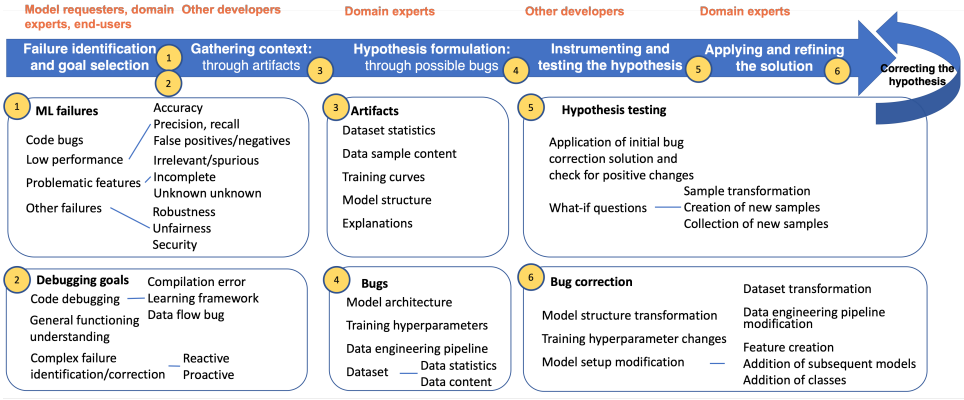


Figure 5.3: Summary of the failure handling practices identified through the interviews. In orange, we show the stakeholders that can intervene in each step of the failure handling process.

NEED FOR ADDITIONAL TOOLS

Our study and especially our failure handling question bank point out to specific needs and wishes from developers, that would merit further research at the intersection between machine learning and human-computer interaction. On one side, the questions in our bank are partially overlapping with the ones of the XAI question bank [498], reinforcing opportunities for explainability works to serve in the failure handling process. On the other side, the questions that are not present in the XAI question bank can serve as invitations for researchers to build new methods and tools, requiring algorithmic research (e.g., "should I focus on the data or algorithm and training hyperparameters?"), or human-computer interaction research especially to facilitate communications between stakeholders (e.g., "is this inference really correct? can we accept it?") and data visualisation (e.g., "does the model make errors with high or low confidence?"). We discuss a few of these research opportunities.

Novel types of explanations. Certain developers mentioned desiderata sometimes similar to existing but rare explanations. These insights corroborate the results of Hong

[8] Similarly to existing initiatives, such as https://docs.microsoft.com/en-us/security/engineering/failure-modes-in-machine-learning

et al. [371] on potential uses of and needs for explainability. In previous works [931], these explanations are summarized as global textual [69, 67] or visual concepts [437, 289], examplars [438] (samples that contrast or are similar to others), and cues (hint on the main differences or similarities). From the identified workflows, it seems that global explanations could greatly speedup certain steps of their process, textual explanations could lead to more accurately identify bugs and support communication between stakeholders, and interactivity could help navigating these explanations.

Data & feature exploration. Our participants spent a large amount of time exploring the dataset for understanding what it represents, to identify potential biases, unknown unknowns, shifts, etc, and to identify and judge model features (with the help of other stakeholders). They (wish to) do so through various types of interactions, e.g., getting random samples for each class in the dataset, clustering images with similar visual content, querying samples with various visual elements, etc. Interactive visualisation tools could greatly support them in easily accessing such information. Existing tools for data exploration in the context of machine learning [367, 116] could be refined for the specific needs identified. Particularly, facing the diversity of hypotheses one can extract from explanation artifacts (subsection 5.4.3), it appears highly relevant to develop user-interfaces for feature exploration, allowing the search of model features at different granularities, the comparison of feature importance, and the matching of model features with expected ones, to investigate the dissonance between human features and machine learned features [934, 116]. One key challenge would be the uncertainty within these features —expected ones are not always known, while learned ones are never entirely known due to the interpretability gap for existing explainability methods [69]—, requiring constant fine-tuning [555]. This highlights the importance, despite the complexity of it, of involving domain experts in the failure handling process, as they can support the developers in identifying additional failures by reporting on their own experiences with challenging edge-cases, and with priorities in terms of correctly-classified data samples and meaningful features, etc. Prior works, especially in the medical context [151], have shown the potential ease in designing a library of test cases, that should be further investigated not only for supporting the responsible use of models by end-users, but also for developing appropriate models.

Model comparisons. The iterative process requires to frequently retrain the model and compare it with its previous versions in terms of performance, features, and other artifacts. Yet, few practical tools [87] support such comparison. As retraining deep learning models is computationally intensive, methods are needed to provide estimates of the changes in these artifacts, e.g., by building simpler surrogate models that would be less heavy to retrain.

Hypothesis testing and bug correction. Hypothesis instrumentation and testing are the main steps our participants skipped compared to the traditional software debugging workflow, due to the lack of methods existing to do so efficiently. Yet, this could certainly save further training time. Recent works such as *Deblinder* [149] or an explainability-based debugging framework [67] start to propose support, by displaying model failures to the developer who has then several options for generating and testing hypotheses, yet targeted bug correction is still not supported by any tool. We recommend to develop such functionalities to allow for faster testing.

5.5.3. INCREASING CLARITY IN THE FAILURE HANDLING GOALS & PROCESS

Our study showed the importance for our participants to access various types of knowledge during the failure handling process. Hence, clear communication with various stakeholders or clear documentation appeared necessary (more information in Appendix 5.7). These results reflect previous works around the data science lifecycle [454, 920, 637, 220, 371]. The information needs and associated communication challenges in these studies and ours are overlapping (e.g., misaligned vocabulary and knowledge). We list below additional challenges.

DESIGNING METRICS FOR CLARITY

The participants rightfully recognized that a model cannot make perfect inferences, and consequently that not all misclassifications should be considered failures but instead that certain should be treated as acceptable. Differently from software engineering, the end-point criteria for deploying a model revealed to be subjective. This subjectivity has been illustrated in prior studies [151], where, similarly to model developers, model users decide on the acceptability of model misclassifications based on their expectations for the model, especially in relation to their own locus of expertise to allow for a successful collaboration between them and the model. Our participants however did not tend to extensively account for this notion of human-model collaboration to decide on failures and the model readiness for deployment, despite the increasing number of research works on the topic [929, 860, 73]. The end-point criteria was also ambiguous, e.g., expert participants, while not considering all model errors equal, did not have a clear process besides trying to attribute different levels of severities to ad-hoc categories of observed failures. Ethnographic work in a data science team has similarly shown the equivocal nature of performance metrics both for the developers and other stakeholders judging the trustworthiness of the models [620], our work expanding these findings to models that are not built in order to discover new insights from data but to automate a process that can typically be performed by humans. This was also observed in prior studies where participants implicitly attributed "cost" to the different wrong predictions [906, 258], and pointed out to the discrepancy between the perceived performance of a model, and its performance as measured by a metric [609, 724, 786, 338]. We suggest to develop metrics or frameworks that would document and account for these various costs. Recent research directions on disaggregating evaluation metrics [83, 521] could include these concerns in their propositions. This would especially allow to adhere to new concerns for accountability and transparency, facing the subjectivity in defining an end-point.

Feature issues are not discussed in machine learning testing research, and only mentioned sparsely within literature around statistical biases in dataset [828, 826], or explainability methods [69, 754, 763], despite their importance (discussed by 17 out of 18 developers). The absence from research could be explained by the lack of metrics to evaluate them, yet one prior study [620], although in a different context, also identified the importance of valid model explanations for stakeholders to decide on using a model in practice. Recent works such as Shared Interest [116] constitute a first step towards quantifying feature failures. Its categorization of samples depending on the correctness of model predictions (proactive or reactive debugging), and whether the model features are aligned with human expectations, is highly reflective of the feature hypotheses iden-

tified in Table 5.2. We however also identified a discussion around the features' weights, not addressed in the literature.

INCREASING TRANSPARENCY BETWEEN DEVELOPERS

Our results, especially certain questions in our question bank, showed the need for developers to communicate with each other. Documentation, although often not used by the interviewed developers beyond model versioning, seems like the right avenue to facilitate such knowledge sharing across developers, similarly to what previous studies also concluded [220, 346]. Next to detailing how a dataset was created [283] or the performance and scope of a trained model [556], future documentations should also focus on "intermediate models" and on logging the experiments conducted across models for a single system and the reasoning behind the choices of experiment. While this could be saved as code, making the steps clear in the form of textual descriptions [678] could fasten the process. E.g., the participants asked what kind of data processing had been conducted, which could be answered without looking into the specificities of the code.

5

5.5.4. BEYOND FAILURE HANDLING: ADDITIONAL CHANGES NEEDED

LACK OF COMMUNICATION BETWEEN RESEARCH AND PRACTICE

Our participants do not use the methods and tools stemming from research publications (except a few explainability methods, and common code development tools such as TensorBoard [152]) due to a lack of awareness. This does not necessarily hint at a technical problem, but at a structural one. It highlights a lack of knowledge or time, from certain developers to search for these materials. Hence, disseminating further the outputs from research to developers appears to be an avenue for future work.

AI EDUCATION

The challenges identified also reveal limitations in the way computer vision is learned. Our participants, while having followed a computer vision course and/or learned computer vision through reading resources around the Internet, primarily build their failure handling process over time by discussing with colleagues (P15 high-CV) *"I never learned computer vision in school. I learned it from the Internet and I had few experiences in internships."*, reading about practices (P16 high-CV and P3 high-CV mentioned specific blog posts about failures and bugs [424]), and through practical experiences (P9 low-CV) *"To improve performance, it would be horizontally (you add more lines to a dataset), or vertically (more columns, that is more features). I'm speaking out of my experience about records. For images, more lines could be data augmentation, more columns could be features that correspond to specific objects."* None has been taught a failure handling process in a curriculum (P5 low-CV) *"I did the deep learning course in the Masters and then some computer vision projects. From that, I learned the basic tools and common libraries for deep learning."* This corroborates prior observations around machine learning practices [24, 788], and debugging of software [549].

Developing education around failure handling for computer vision models could benefit developers, as is suggested by position papers [739] and successfully experimented with in research on teaching debugging. Particularly, research around software debugging teaching [549, 542, 596], and data science teaching [474, 292, 788, 869, 805,

512, 504] proposes teaching through exercises with examples of workflows or hierarchical lists of questions to ask for correctly "debugging". Computer vision developers could also exploit online communities to get further training (none of our participants mentioned using these frequently), similarly to data science developers [749]. Failure handling tasks could be shared online and executed in collaboration. Yet, one would need to investigate how to share relevant materials (e.g., trained model, datasets), information (documentation about the task and model), and solutions.

5.6. LIMITATIONS & THREATS TO VALIDITY

There are several limitations in our study. While we do not think they impact the validity of our results, tackling them in the future would improve the generalisability of our findings. We used one simple scenario, that enabled our participants to easily describe their usual practices, as the various examples the participants brought from their own use-cases and some comments testify, e.g., (P13 high-CV) *"My very first thought was: this is a very realistic use case"*. Yet, using such scenario might obfuscate specificities of their own use-cases, such as competing incentives they might encounter (they primarily referred to constraints around data collection). However, using a different use-case per participant would have not allowed to fairly compare practices, and would have posed confidentiality issues. Freeing them from competing incentives places them in a more ideal situation to discuss their process. Besides, our scenario presented the participants with information about the model to "debug", without the actual development code —that they did not ask for. A task where they would be presented with the training code could provide additional insights, but would require longer interview sessions. Our methodology inspired from previous works [220, 365, 67] already provided us with main challenges.

We focused on failure handling in development. Practices might differ after deployment, as other failures and constraints might occur, and additional stakeholders might be involved. We looked primarily into correctness and feature failures that are still understudied. Yet, many more types of failures might arise. We interviewed a considerable amount of participants and devoted our efforts to cover practitioners with various levels of experience. Such qualitative approach can never completely assure that we gathered all failure handling practices that exist. In the future, one might want to perform studies with other methodologies, e.g., ethnographic work for in-context practices, code-based studies, different focuses, and in specific domains of application, to complement our results. Finally, we focused on models for image-based computer vision applications, and hence we cannot conclude certainly on the applicability of our results to other types of models. We can however mention that our discussion on the organisation of the field echoes prior discussions around other applications such as the ones relying on tabular data [427]. Besides, the design opportunities we highlight are applicable to other applications as they are not application-specific. However, the required technical work would differ to leverage the relevant artifacts, that are different across applications —and more or less researched until now (e.g., more research on explainability for tabular-data based applications has been performed than for image-based applications). Whether these design opportunities are necessary for practitioners developing these other applications, should be investigated in the future, and our work can provide inspiration to do so in

terms of insights to look for. It is fair to assume that certain of the insights would hold as our participants and other practitioners have typically received the same training, and many machine learning models across applications share similar properties.

5.7. CONCLUSION

In this work, we conducted 18 semi-structured interviews to outline the practices of machine learning (ML) developers for ensuring robustness of computer vision models and handling the failures of these models (Figure 5.3). We showed that, while practices broadly follow the traditional software debugging workflow, they differentiate by the ambiguous way the model requirements are defined, by the type of hypothesis formulation and instrumentation activities performed in the machine learning context, by the artifacts employed to facilitate the workflow, and by the fluidity of the relevant concepts. Besides, failure handling workflows are typically performed manually and in collaboration without resorting to methods developed specifically for machine learning models (Table 5.4). Finally, developers tend to have a narrow understanding of the failures and bugs that any machine learning model might suffer from, skewed by their prior experience. This understanding yet includes problematic model features that are not typically investigated in scientific literature. These insights point out to various limitations and challenges in the current failure handling process, that should be tackled through both structural changes and socio-technical research. Especially, we drew a list of research opportunities at the intersection between HCI and machine learning, going from the creation of a collaborative library of best-practices, to the development of failure handling methods and user-interfaces, and of support for communication between stakeholders. Besides, we identified a gap between the research that develops certain types of explainability methods, and the practice where developers sometimes express the need for a different type of explanations that would better support them in diagnosing model failures. In Part III, we aim at bridging this gap, by proposing new types of explanations, and investigating to what extent they do support the developers further. For now, in the remaining of Part II, we zoom in on specific types of harms caused by the use of machine learning models, algorithmic unfairness (and a few other societal harms), and we study how developers tackle such harms. We also investigate further what are the deeper factors that impact developers' practices.

APPENDIX

RESEARCH METHOD

Figure 5.4 and Figure 5.5 respectively present example explanations and the workflow template shown to the participants during the semi-structured interviews.

ADDITIONAL RESULTS

CORRECTING BUGS TO SOLVE THE FAILURES

The participants used one of four strategies (followed by model retraining) to correct bugs, depending on the bugs, and on their familiarity with the models.

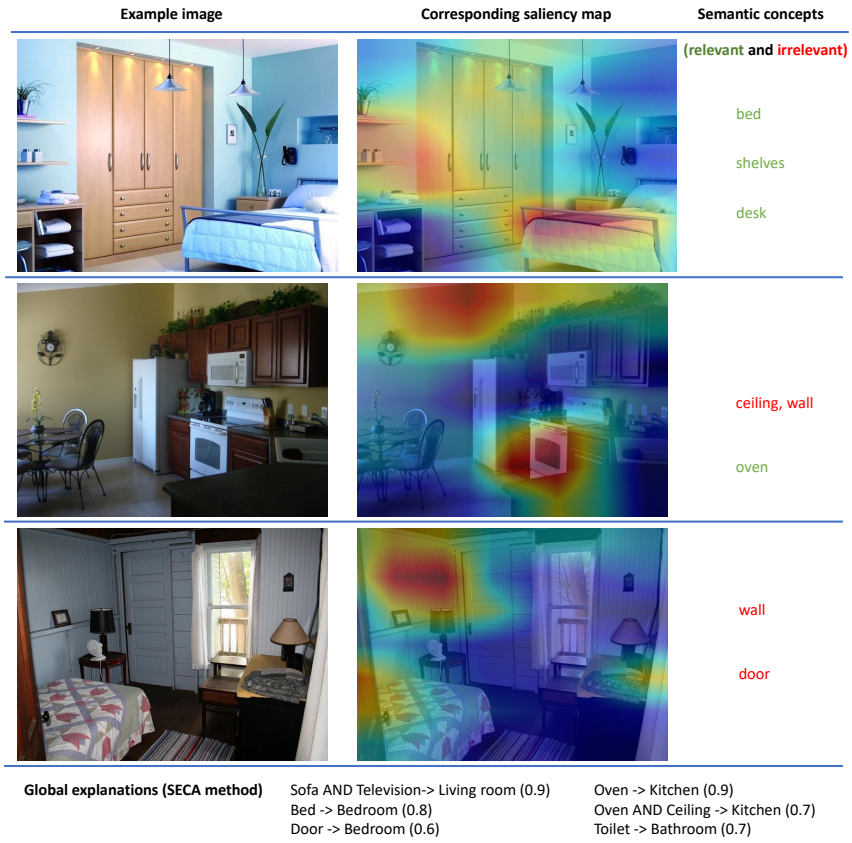


Figure 5.4: Example explanations (local visual and textual explanations, and global textual explanations) showed to the participants, when they would mention them, or at the end of the interviews to trigger further reflections about them.

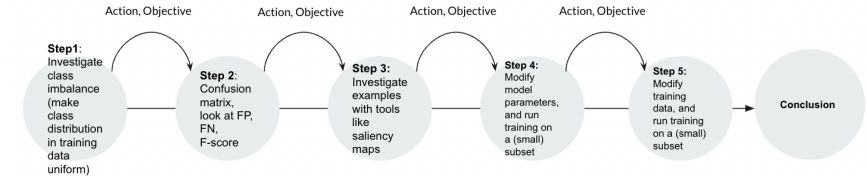


Figure 5.5: Example template provided with the design brief, and filled in by one participant. The template shows empty circles and arrows representing objectives, actions, and transition triggers, reflecting each step of the failure handling process, and helping the participants to structure their thoughts.

Dataset transformations. Participants with no experience in explainability and experts who do not wish to engage deeply with the data content tried to resolve correctness failures through *typical data augmentation* methods such as applying mirroring, rotation and colour contrast algorithms. *"I employ some augmentation techniques or ar-*

tificial data to see if I can get away with this. This would be a method to further regularize the model to make sure that it's not overfitting." P3 high-CV. P5 low-CV also mentioned *"applying some dirty labels (for instance I would apply the label of "kitchen" to "dining room" pictures) to create a positive perturbation and rebalance the number of samples"*.

Other participants mentioned *feature-specific transformations*: adding or removing images with specific features, or obfuscating irrelevant information from images. *"If there are cats only at dining rooms, I should do cat recognition and mask them."* P11 mid-CV. The hypotheses of these participants revolved around the relevance of the model features, and/or the existence of unknown unknowns. Transformations of the *data engineering process* were also mentioned by some experts as simple steps (e.g. increasing image size, changing scaling) along model modifications.

Modifications of the training parameters. Expert participants transformed the *loss function* to penalize classes with higher error rates: *"It is easier that the model learns to base the classification on different things than when you add more data"* P9 high-CV. They also gave more *importance to training samples* erroneously predicted *"It's like the Bootstrap algorithm where you keep re-feeding falsely predicted samples into the model, assigning higher weights for the last computation."* P3 high-CV. This method is used by participants with computer vision experience, as they are more familiar with the functioning of the models. *"It allows me to avoid using a parameter so that the classification of two classes becomes more diverse, and the optimization of the training based on a more relaxed representation."* P14 high-CV. A few participants with some experience also discussed tuning *training hyperparameters. "What I found is that setting the right parameters, especially learning rate or batch size, can help the model avoid certain biases"* P7 mid-CV.

Model transformations. Hyperparameter tuning (e.g., changing the model architecture) was the main solution of high-CV experience participants, which sometimes came hand in hand with simple dataset transformations. *"The network didn't learn the task. It's the famous bias variance. You have to see whether it cannot generalize, which means that it has been overfitted to the training set. If you have a lot of data available, you just throw more data at your model hoping that it can generalize better. If the data is scarce, let's say you are in medical imaging and each MRI is from a patient, you cannot collect more data. You have to change your model and that's more expensive because a machine learning expert needs to work on it. Instead, for data, you can just crowd source it via Amazon Mechanical Turk, it's much cheaper. There are also scientific insights: if the task is simple, adding more complex model doesn't make sense, but usually for computer vision task, it's complex enough that you can have a complex model."* P16 high-CV. Low-CV participants did not engage in such activity as they were not familiar enough with the functioning of computer vision models *"That's where I'm hitting a wall. I would change something about the model. But I need to understand that model a little better."* P4 low-CV.

Changes in the model setup. Certain participants with low-CV experience proposed additional solutions based on their own experience. These solutions are not mentioned in literature, but useful in practice. They would a) build separate models for the most

confused classes, b) create additional classes for the ones that are too diverse in terms of image content, or c) append a rule-based model, to correct inferences with heuristics defined on the content of the images. *"Establishing rules means to modify the model decisions manually. It's not something that you should do, but if it's a requirement, it can be done. Let's say this is towards 60% confidence, it's a weak prediction. The probability of being a dining room is lower than average. So, once you have the combination of low probability of being a dining room and you also have the presence of a metal component intertwined with black glasses, then you can push it to the kitchen classification."* P8 low-CV. d) Others proposed to engineer features based on visual information identified in the images *"Most bathrooms have a mirror, then it's really good if we can classify if there's any mirror. From specific elements that you discover, you arrange other features."* P8 low-CV. e) One participant mentioned deferring difficult cases to humans, or using active learning to fine-tune the model. *"The way to proceed is through the human eye: you leave extreme cases to workers to annotate. The model can learn about the general cases and leave you the extreme ones."* P7 mid-CV.

COLLABORATION BETWEEN STAKEHOLDERS FOR HANDLING FAILURES

Results. As it appeared along the previous subsections, for most participants, failure handling was not a lonely process. Practitioners frequently mentioned communicating with other stakeholders during the process.

- *With other "developers".* The practitioners often need to discuss with other individuals who took part in the model development process, dataset creation, etc. to obtain more information about choices and previous experiments. Especially, expert practitioners implicitly had a list of steps they always perform when developing a model (e.g., training with different architectures and hyperparameters), and a list of necessary operations (e.g., normalization and standardization of the dataset, data augmentation, etc.) (P3 high-CV) *"I suppose that the input has been sufficiently preprocessed? I would normalize, typically by the max value if we are talking about standard RGB images."*

- *With model requesters.* To clarify when the model is satisfying, the practitioners also rely on the model requesters (subsection 5.4.2) who are the final judges of the acceptability of the model (and the requirement providers) (P14 high-CV) *"the final decision on how much you should improve the model is given by somebody else (the client, the model owner, ...) given whether it is a critical situation."*

- *With domain experts.* Domain experts are involved by the practitioners (when reachable) to better understand the target task and potential pitfalls, and to judge how ready the model is, to identify feature expectations, and to reason on the relevance of certain features when searching for model bugs and feature failures (P14 high-CV) *"the part of saying whether it's ok that the model makes a specific mistake, it's not up to me. It's up to the experts."* P7 mid-CV also mentioned questioning the experts who are the end-users of their model to resolve data ambiguities, whether they are inherently ambiguous, or whether one specific class can be attributed to the samples (P7 mid-CV) *"Give it to people who are as close as possible to the end-users and say: what do you think? Is this a bedroom or a living room?"*

- *With potential end-users.* The developers have to convince the model requesters and users (who are often the experts) of the validity of the models. P14 high-CV explained *"You are the person that can communicate the density of information to a specialist like a doctor. When we have a meeting, we show the model understood the class."*

Implications. Our results identify additional communication needs from the developer to non-developers, especially for defining when a model is suitable for deployment, whether specific failures on single samples are acceptable, and which features one should expect [133, 371, 620]. Since the accessibility of domain experts was one of the main problems for the developers, research should investigate how to facilitate collaborations around these specific concepts, potentially with the development of remote, asynchronous tools, and common languages (possibly inspired from existing knowledge elicitation methods [431]), e.g., to indicate relevant features. Existing works that facilitate the cooperation between domain experts or end-users, and a machine learning model, could be adapted to these specific concepts [455, 149, 916].

6

PRACTICES FOR DIAGNOSING & MITIGATING SOCIAL HARMS

6.1. INTRODUCTION

In this chapter, we continue our investigation of the practices of machine learning (ML) developers. Differently from the previous chapter (Chapter 5) where we investigated model development and diagnosis practices broadly, this time we prompt the developers specifically about the social harms their ML models might cause.

As described in Part I, in reaction to algorithmic harms, different research communities have focused on ensuring *distributive fairness* around the outputs of the models. They have focused on developing *algorithmic unfairness* metrics [848], mitigation methods [268], and toolkits [107, 91, 704, 842, 205]. Another line of research has taken a critical and interdisciplinary stance on the concept of algorithmic fairness. It has explored broader *algorithmic harms*, i.e., issues arising from the development or deployment of an ML model, around not only distributive fairness, but also the questionable desirability of using ML for a task, the use of inappropriate training datasets, the negative impact of model training on the environment, or the poor labor conditions of the crowd workers involved in system construction [62, 557]. The complex and negative social and environmental impact of these issues has been argued to be inaccurately (and incompletely) reflected by the proposed algorithmic fairness metrics and consequently inadequately addressed by the mitigation methods [508, 718, 361, 457, 870]. Algorithmic fairness concepts, in particular, that are still nascent are said to represent only a narrow simplification of distributive fairness (cf. section 6.2). Hence, it has been recently argued that while one can employ these concepts towards building non-harmful models, they should maintain a critical attitude to avoid techno-solutionism [543, 254].

In parallel to these theoretical works, the HCI community has adopted a tangential lens. It has begun investigating how ML developers *build (fair) models by relying on the algorithmic fairness concepts*, and what challenges they meet when using the algorithmic fairness tools and toolkits that ought to support these practices [369, 846, 220, 679,

481, 666, 521, 619]. It has also studied a gap between current formalisations into algorithmic fairness metrics and user perceptions of distributive fairness, pointing out the contextual nature of fairness [791, 836, 786, 864, 338, 425]. However, relatively little work has considered how ML developers approach, perceive, and *tackle the broader worrying algorithmic harms caused by ML models and the gap between algorithmic fairness and distributive fairness* —especially since these harms and the gap might not necessarily be visible to the developers via existing toolkits. Developers are often the first, sometimes the only, and always important stakeholders who can act on algorithmic harms through the various design choices they make. It is, therefore, vital to understand how developers perceive downstream harms caused by ML models they deal with, how they choose to handle potential harms, and whether their perceptions and practices are fragmented. Addressing this knowledge gap is a crucial step towards questioning the broad and potentially negative impact of algorithmic fairness solutions shrouded in techno-solutionism. Thus, we frame the following research questions: *How do machine learning developers envision and tackle unfairness issues and other harms that might arise from the models they develop? How does the research/practice gap manifest in this step of the machine learning lifecycle? What are the main limitations of their practices?*

To answer this research question, we conducted a think-aloud study followed by semi-structured interviews with developers ($N = 30$). We recruited developers through a combination of snowball [300] and convenience sampling [352], corresponding to varying demographic and educational backgrounds and varying levels of experience with ML and algorithmic fairness. We first tasked developers with investigating two ML problems that merit various considerations of harms, by providing them access to existing algorithmic fairness toolkits, and observing their practices and reasoning around the harms. Next, we conducted semi-structured interviews with the developers by asking them why they prioritized certain harms, envisioned impact of the activities in their ML lifecycle, and foreseen challenges in the given task. This resulted in transcripts spanning 2207 minutes, which we analyzed using inductive and deductive coding.

We found a new set of activities performed by developers to tackle harms, that had not been reported in prior empirical studies of the ML lifecycle. Across the developers, we observed fragmented conceptions of harms and practices towards algorithmic harms, the way they are prioritized and handled. We identified different misconceptions and various ways in which harms are mishandled. Our results corroborate findings from existing works on the use of algorithmic fairness concepts, while extending their generalisability to another set of domains of application in an effort to methodologically triangulate results. Importantly, our work provides an extensive understanding of the considerations about and approaches for broader algorithmic harms, where developers typically follow a similar, often intuitive but not always substantiated reasoning process. Where some developers are satisfied trading off accuracy with fairness and ticking algorithmic fairness checkboxes that build up a false sense of fairness, others recognize the complexity of the socio-technical issue, and the diverse unsolvable concomitant tensions. This calls for various theoretical and empirical investigations, to guide developers in their design choices. Apart from advancing the current discourse around ML practices to curtail algorithmic harms, our work also has broad implications on the design of fairness toolkits and the fostering of reflexive practices among developers.

6.2. RELATED WORK

6.2.1. CONCEPTUAL UNDERSTANDING OF ALGORITHMIC HARMS

ALGORITHMIC UNFAIRNESS

Each step of the ML lifecycle might create or reinforce distributive unfairness [546, 811]. Unfairness might come from biased training data, and from the choice of ML algorithm and the tuning of its parameters, and it can be reinforced by feedback loops. Works around algorithmic unfairness develop fairness metrics that aim at measuring distributive unfairness in the outputs of the final model or in a dataset, and unfairness mitigation methods that ought to improve the model distributive fairness as defined by the metrics.

To date, there exists three types of fairness metrics [848] (statistical group metrics, similarity-based individual metrics, and causal metrics) characterized by the type of information they require to be computed. Research has shown various impossibility results between metrics, e.g., stipulating that any two of the group metrics cannot be satisfied simultaneously [442]. Trade-offs with other optimization objectives such as differential privacy [646, 893] have also been explored. These tensions constitute a challenge for ML developers to choose the appropriate targets for their task. In an effort to clarify this choice, the fairness metrics have been shown to account for different moral and political philosophy theories and especially a variety of normative egalitarian considerations on which one might align [104, 477, 254].

Unfairness mitigation methods are classified into three types (pre-, in-, and post-processing) depending on the components of the ML pipeline on which they act [268, 66]. They are adapted to specific fairness metrics, and only applicable to specific types of data [268], numbers of protected attributes, and tasks. Practically, these methods bear limitations in terms of performance and especially of their brittleness to small variations in the model training process (e.g., data splits) [268] and data processing activities [726, 256], to imperceptible variations in training frameworks [648], to the application of additional model optimization methods like model pruning [372], or to distribution shifts [767, 759].

BEYOND ALGORITHMIC UNFAIRNESS

A few works have looked beyond algorithmic fairness to identify other harms of ML [62, 557]. We do not aim at a comprehensive account of these harms but present a few (cf. Appendix Figure 6.2), that are highly worthy of consideration according to the literature. To the best of our knowledge, practices around these harms have not been investigated extensively in prior studies.

Some harms reveal by considering the conceptual limitations of algorithmic fairness metrics and methods. Looking at output distributions, algorithmic fairness cannot reflect the contextual factors that influence what is considered fair (distributive justice). For instance, it wrongly assumes that parity is always desired in the model outputs [508], it does not account for the impact one same output has on different receivers of this output [557], and simplifies intersectionality issues [718, 361], while also not accounting for indirect impact on non-data subjects [457]. Besides, looking at the process to reach algorithmic fairness (procedural justice), the metrics and mitigation methods do not ensure that the way in which the unfair situation is addressed is aligned with moral principles

[870]. For instance, a model can reach low disparate accuracy by treating all individuals or groups unjustifiably [571], or differently (e.g., post-processing method allocate different decision thresholds for different groups) which consists in direct discrimination [308]. As a result, algorithmic unfairness can be artificially solved, but the structural causes of the initial unfairness might remain [254, 571].

Three other categories of harms have also been discussed, that rise from the use of machine learning (ML) techniques. ML requires to use *datasets* whose *schemas and sampling* can be harmful. For instance, certain attributes and their values might be offensive [132, 110] or inappropriate [546], e.g., use of non-volitional or privacy-infringing attributes [315, 834]. They might neglect the complexity of the concept they ought to represent (e.g., the race attribute [335]), or force populations in non-adapted categories (e.g., binary gender) [694]. The dataset distribution, despite a correct dataset schema, might present biases [557, 878, 570], e.g., leaving out of consideration certain populations. Research also questions the *desirability of the prediction model* in the first place, its use for potentially undesired applications [558, 376, 432, 557], and how it impacts the current structures in place [257]. Using ML for certain tasks might be questioned, for instance because it means making decisions for people by comparing them to others instead of following the principle of individual justice [105, 254], or because it only allows to reproduce historical, potentially harmful, data patterns [653]. Certain researchers also question the *negative externalities caused by the production process* of ML applications, such as the environmental impact of data centers and model training [132, 93], the poor labor conditions of crowd data workers [690, 909, 948, 887], the privacy-infringing data that are often used for training [660], etc.

6.2.2. STUDIES AROUND ML PRACTICES AND ALGORITHMIC FAIRNESS

Several studies have investigated ML practices around algorithmic fairness. Topics of focus are specifically around general challenges met by developers [369, 846, 666, 521, 564, 719, 619, 878, 612], focusing on obstacles and limitations for the application of algorithmic fairness paradigms in general, and of fairness toolkits more specifically [679, 481, 220]. Most of these studies consisted in asking ML developers to report on their perceived challenges. Instead, similarly to Deng et al. [220], we conduct one of the first task-based studies to observe practices and identify potentially unreported challenges.

Findings typically outline the need to support developers to concretely use algorithmic fairness paradigms, challenging due to their context dependence and the current lack of guidance [369, 521], and due to the need for adapting existing metrics and methods incompatible with targeted tasks [369]. Additional factors such as business incentives, are also further characterized and shown to be obstacles to develop fair models [666, 521, 523]. Interviews [220, 679, 481] also show the beneficial use of fairness toolkits for developing fair models and learning about algorithmic fairness. Yet, they also show their limitations in terms of support provided to developers for designing the right algorithmic fairness evaluation, noting that participants often inappropriately change their modeling task definition to fit existing tools. We discuss how each of our results relates to and corroborates these studies in section 6.4.

While all these studies are important for helping developers build more ethical systems, they do not provide an outline of the concrete steps through which ML developers

build (ideally harmless) ML models, nor do they consider the known limitations of the algorithmic fairness metrics and methods and other harms that ML systems might pose. We fill this void by investigating developers' understanding and practices towards the broader socio-technical harms.

6.3. METHODOLOGY

Interview Procedure. To identify the nuances through which developers understand and handle harms, we adopted an empirical and qualitative approach via 30 semi-structured interviews. When participants were already familiar with ML fairness and fairness toolkits, the interviews lasted around one hour, revolving around one model development Task T1. For the other participants, the interviews lasted around two hours each, involving three stages (Task T1, a tutorial about one fairness toolkit described in more detail in the Appendix), and Task T2). For this second batch of participants, we studied their practices without and then with a brief tutorial on algorithmic fairness tools. This mimics real-world scenarios where they may or may not come across a fairness toolkit, and experiment with it while working on their own harm-sensitive use-cases. In Figure 6.1, we show the workflow of the interviews with the questions asked in each stage, for the two groups of participants. We asked three types of questions: *background experience questions* (demographics, experience with ML and algorithmic fairness); *reflection questions* around algorithmic fairness, harms, or toolkits, and around general comments, wishes, doubts, and challenges the participants might have about their workflow or harms; and *process questions* to understand the reasoning behind each participant's activities during the tasks, especially in relation to harms that might be impacted by these activities. In total, we collected and transcribed 2207 minutes of recording.

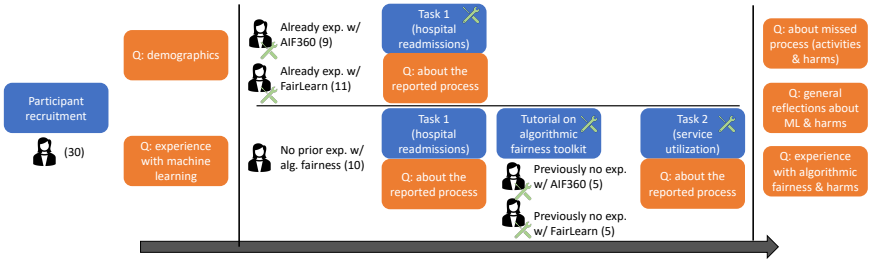


Figure 6.1: Interview procedure for the participants already experienced with one of the two fairness toolkits, and for the participants who did not have any prior practical experience with algorithmic fairness and harms. In blue, we represent **the main steps of the procedure**, and in orange we represent the **questions posed in each step**.

Participants. We recruited our participants between April to June 2022, by means of personal networks, targeted requests on social media (35% of positive responses), calls for participation on the official Discord or Slack communication channels of fairness toolkits, and snowball sampling. The participants received no financial compensation, and their contributions were voluntary. Our institution's ethics committee approved the

study. All participants signed an informed consent form acknowledging the risks involved with participating, as well as agreeing to the interview being recorded (all interviews were conducted online), transcribed, anonymized, destroyed, and consented to the results being used in scientific publications. 30 participants were recruited across research and industry institutions, and across application domains such as healthcare, finance, and predictive maintenance. Manual sampling was performed to make sure that all participants have (a) responsibilities in ML model development, deployment, or evaluation; (b) varying levels of prior experience with ML, ranging from 2 to 15 years; and (c) varying experience with algorithmic fairness. The participants differ in terms of demographic (nationality, gender, and age) and educational background.

Materials. We chose two use-cases, the first one involving the prediction of *hospital readmissions* within 30 days for individual patients [798], referred to as Task `T1`, and the other involving the prediction of low or high *medical services utilization* [342], referred to as Task `T2`. We pre-processed the two corresponding datasets in order for them to have similar characteristics (similar, tractable number of attributes, and number of records), and to be prone to similar kinds of harms. Table 6.1 illustrates harms in the two use-cases. We chose the domain of healthcare because it is prone to various harms, requires expertise to be handled correctly (we could check whether the participants mentioned the limits of their knowledge), several corresponding datasets were available, and these are not the most frequent use-cases in the algorithmic fairness literature which should allow each participant to investigate them for the first time. These design choices represent realistic scenarios where ML developers often have to develop or deploy models without having an extensive expertise in the domain of application (only 4 out of the 30 participants reported having some healthcare knowledge, among which only one had more extensive, practical experience).

For each task, we shared a Google Colab notebook with the participants, which included a design brief with one of the two datasets pre-loaded. The design brief mentioned that a hospital (or an insurance company) wanted to optimize their cost and services (or their prices), and therefore wanted to investigate whether ML could help them predict readmissions (or utilization, respectively). The institution tasked the participant to investigate this feasibility possibly using the dataset they had collected, and to report on their findings by speaking outloud. For the interviews with developers who had used a fairness toolkit in the past or with the ones whom we introduced to a toolkit, we loaded a specific toolkit (FairLearn [107], or IBM AIF360 [91]) into the notebook, that they were most familiar with.

Analysis of the Transcripts. We analysed the transcripts using a combination of inductive and deductive coding. The first author identified the segments reporting on the main themes we wished to discuss (e.g., harm conceptions, identification, and handling), and coded any other emerging themes (e.g., factors that developers trade-off when developing ML models) in close collaboration with four other researchers. In a second round of analysis, this author studied each higher-level theme in detail, and identified the response declinations of each participant (e.g., choice of fairness metrics based on expert advice, or applying all of them). In a third round of analysis, the author, in dis-

Table 6.1: Examples of potential harms introduced in the two use-cases presented to participants.

Category	Task 1: Hospital readmissions	Task 2: Medical services utilization
Desirability of the ML model		
Task encoding desirability	Over-simplified and potentially irrelevant target labels (unjustified threshold of 30 days).	Potentially unethical task where insurance prices would be computed based on estimation of medical services utilization.
Distributive unfairness		
Biased dataset causing unfairness	High imbalance for various potentially sensitive attributes (e.g., `race`: 74% Caucasian, 20% African American and the rest divided in 4 other categories).	High unbalance of `race` (white at 80%, others at 20%).
Sensitive attributes	"Classic" sensitive attributes (e.g., `gender`), and rarer, potentially sensitive ones (e.g., `marital status, weight`). Proxies (`region` synthesized to be highly correlated with race).	Same with `race, sex, age`, and question of `marital status, military service`. Proxies (e.g., `race` highly correlated with `poverty status`).
Conceptual limitations of metrics	Consequences of the model output not only for the patients but also for their family, not measurable.	Consequences of the model output not only for the insured but also for their family, not measurable.
Harmful datasets (focus on attributes and their potentially inappropriate nature)		
Attribute information	Utility and ethics of using the `marital status` to predict hospital readmissions.	Same for `marital status`, and `military service status`.
Encoding	`Gender` encoded as binary, `age` encoded into three categories.	`Race` encoded as binary (white, non-white).
Impact of various core technical ML activities onto these harms (especially onto algorithmic unfairness)		
Missing data	Synthetically introduced to correlate with specific values of the `weight` and `medical speciality` attributes.	21% of synthetically introduced missing values for the `weight` attributes with primarily values corresponding to `gender` female, which would lead to gender imbalance if dropped.
Outliers	Synthetic injection of outliers in the number of `lab procedures` attribute	Outliers introduced within one synthetic attribute corresponding to an aggregation of several other attributes.
Duplicates	No visible duplicates.	20% of synthetically introduced duplicates, that would cause target label imbalance if dropped.

cussion with the other authors, reconciled redundant codes, reviewed the consistency of codes across granularity levels, and identified additional transversal themes from existing codes (e.g., prioritization of harms or requirements). This process resulted in 276 codes. Based on our preliminary analysis of the literature about algorithmic harms, we critically reflected on the codes. We identified participants' perceptions or approaches that are incomplete or invalid while accounting for the recency and subjectivity of the knowledge built on the topic, where such information was available. Note that we do not believe there is a single, correct, approach to perceive or tackle harms. Further details about the interview participants, materials and questions, and the resulting codes, are included in Appendix. All our materials, resulting data, code and analysis will be shared publicly for the benefit of the community and in the spirit of open science.[1]

[1] https://osf.io/dmr82/?view_only=a00e68796f494fbb9776cf9a95fb7051

6.4. RESULTS

We present the findings of our study following the stages of the ML developers' process. We first discuss how ML developers recognized potential harms, how they analysed them to decide whether to tackle them, and finally which approach they adopted for mitigation. We contrast the identified practices with those reported in literature on algorithmic fairness and harms to identify alignments between the practices and the research, and the corresponding limitations.

6.4.1. RECOGNIZING ALGORITHMIC HARMS: DISPARATE REFLECTIONS AND (MIS-)CONCEPTIONS

A RICH SET OF HARMS ENVISIONED BY DEVELOPERS

Our analysis of the interviews resulted in the identification of three conceptual layers of considerations about harms. The first layer corresponds to macro-categories of harms, where the four macro-categories identified in the literature (cf. subsection 6.2.1) match with those discussed by developers (we color-code the considerations based on their macro-category, and underline them based on the layer they belong to). *P28 "We need to look at the bigger picture to see if our work is ethical. That can go for the carbon footprint, the sustainability, the impact this may have on the labour market, and in warfare."* The second layer corresponds to the sub-categories of harms identified per macro-category. There, we identified a number of harms that haven't been discussed in-depth in the literature. For instance, during the development process of an ML system, P6 discussed their concern for equally sharing resources (e.g., GPU clusters) across those of an organisation who might need them. About the desirability of an ML system, multiple developers discussed the modes of human-ML collaboration that the system should be designed for to be considered acceptable. They suggested that although ML can serve to remove human biases, one should remain cautious when using the outputs of an ML system, and ensure human oversight –this shared control is often not discussed in the context of harms but solely accuracy [74]. The third layer corresponds to the complementary and opposing conceptions of specific harms. We identify a diversity of considerations, that are not all discussed within the literature. For instance, while several works argue that one should consider the ethicality of the goal an (ML) system is built for [558, 432], and then reflect on whether using ML for this goal is appropriate –considering what ML entails, e.g., in terms of repeating previous patterns [718], or explainability of the outputs [145]–, it appears that research has not considered the practical concerns of P3. P3 argued that one should not employ ML in a system in contexts where the functioning and outputs of the system need to be updated at a fast pace to avoid certain harms. Indeed, P3 explained that ML-based systems are not considered flexible enough for quick updates, as ML developers shy away from modifying them. *P3 "Everybody is afraid of changing something "if you change this, it breaks this". So we usually start with: what is the problem that you are trying to solve? could it be solved by simple query, by business rules, or statistical model? If not, by machine learning? It's not about amplifying the buzz and having AI everywhere. It's about the real value of using it."* The different conceptions of harms across layers are exhaustively listed in Table 6.4 and Table 6.5 in the Appendix.

A HIGH DIVERSITY IN THE BREADTH AND DEPTH OF REFLECTIONS

During the interviews, each developer touched upon different categories and sub-categories of harms. For example, in terms of categories, certain developers did not mention any harm at all before being introduced to the fairness toolkits, while others reflected on a large diversity of them; only 3 developers discussed crowd workers' labor conditions. Many participants mentioned concerns around privacy infringement in training data, yet, at the conceptualisation level, most of them envisioned issues specifically with either consent for data use or with data anonymisation, but not both. Similarly, several participants engaged in critical reflections about the appropriateness of the data schema, but not all of them discussed both the completeness of the set of attributes, the meaningfulness of each attribute, and of their encoding. This highlights the importance of delving into the layers of harms and fostering concrete discussions, since developers may stop at the first or second level in their considerations, and may not foresee the deeper issues.

DISAGREEMENT AND (MIS-)CONCEPTIONS

Disagreement on what constitutes a harm is natural due to the inherent subjectivity in the perception of harms. Disagreement is clearly illustrated by the reflections of our participants around the third layer, where we identified potentially opposing considerations. For instance, in terms of the goals of the system, not all participants reflected on all stakeholders, but instead adopted the lens of a single stakeholder, e.g., declaring the system desirable as soon as it benefits the organization that deploys it, or considering the morality of the goal towards society, which might not always lead to the same conclusion. *P16 "It's appropriate and relevant for the business. They want to save money or to reduce time of the workers."* In terms of feature sensitivity, developers disagreed on the exceptions making a sensitive feature not harmful, e.g., exception as soon as the feature is related to the target label, or if it is volitional *and* related to it. Even when developers agreed on the sensitive features, they did not envision the same ideal use of these features for the system to not be harmful. Some mentioned that such features should not be used in any case, whereas others proposed exceptions, e.g., when the model does not attribute high-importance weights or when its output does not display disparities across them. These opposing considerations also surface across the macro-category layer. For instance, certain participants' conceptions of harms were found to be contextual and extremely relative, as they considered the environmental impact of model training non-harmful as long as the ML system was desirable for society or that it would somehow allow to save some energy somewhere, while others solely saw the potential for harm. We also found subjectivity around questions of distributive fairness. Different developers mentioned different conceptions of the ideal output distribution, that can be attributed to different moral assumptions and theories in political philosophy [104]. For instance, they referred either to notions of predictive parity or to notions of statistical parity that reflect different cases of equality of opportunity [347].

Certain considerations might be considered questionable according to existing re-

search and regulations. For example, certain sensitive features are protected by law in certain contexts and certain output distributions are demanded, yet developers discussed neither these nor questions of intersectionality. Research [361, 508] has shown the limitations of considerations of parity in output distributions, that were only envisioned by three developers. Besides, while 63% of developers discussed the need for changing current data distributions to reflect algorithmic fairness, 30% of developers incorrectly assumed that a distribution representative of the real world will always lead to train a fair, non-harmful, model (and that "debiasing" a dataset is not desirable) as one should not distort the way the world is (WYSIWYG –What You See Is What You Get [267]). Yet, literature [546] has shown that for building fair models, one should account for existing historical biases (WAE –We Are All Equal) in data.

VARIABILITY IN THE UNDERSTANDING OF THE ACTIVITIES THAT MIGHT CAUSE HARMS

Prior work [381, 714, 256, 726, 878] has highlighted a wide spectrum of challenges surrounding some of the data and model activities of the ML lifecycle, that can impact algorithmic unfairness and other data-related harms. In the interviews, developers discussed such activities and others that they perform — data processing, data cleaning, crowdsourcing-based data labeling, dataset splitting, and model building. However, most developers did not envision any harm that these activities might cause or reinforce. They also did not discuss the potential negative implications of more-well known issues such as distribution shifts between deployment and training, be it in terms of accuracy (more familiar) or algorithmic unfairness [674]. Only 3% to 10% of the developers acknowledged potential harms from these activities (e.g., P5 for data outliers, P21 for missing values, and P1, P29, P30 for other preprocessing activities), mentioning skews to the datasets that the activities might cause, which would lead to algorithmic unfairness in the outputs and/or silencing certain populations in the dataset.

Some envisioned connections between the activities of the ML lifecycle and harms went beyond what is discussed in the literature. For instance, prior work [726] has discussed **processing of data errors** as an activity that can impact algorithmic fairness. Yet, P29 suggested thinking beyond the technological handling of the errors, to their meaning for the data subjects and the design of the system beyond the algorithm. *"In Southern California where there's a large Hispanic population, when testing a model to allocate poverty benefits to low-income individuals, they found that Hispanic applicants were rejected at higher rates, just because these applicants aren't fluent in English [mentions **data outliers**]. They're having trouble with the application form. So the solution to make this system fair was just to **offer the form in Spanish**, you don't do anything with the model."* Cf. Table 6.6, Table 6.7 for the ML activities and envisioned harms.

6.4.2. ANALYZING HARMS & SETTING GOALS: GOAL DIVERSITY BASED ON VARIOUS ENVISIONED TENSIONS

ENVISIONING TENSIONS & IMPOSSIBILITIES

A recurring theme along the developer's process is tension and trade-off. Along the ML lifecycle, factors that developers account for and that trade-off with considerations around harms trickle down. Some of these tensions already emerged when conceptualising when to consider something harmful (subsection 6.4.1). Others are discussed

when deciding whether (e.g., how important the harm is compared to system objectives) and how to handle a harm (e.g., mitigating distributive unfairness by collecting more data might be privacy infringing). We identify five types of such tensions. Developers take into account requirements concerning the ML model capabilities (e.g., accuracy, explainability), the system infrastructure (e.g., computational power for training), and the development process (time). They also bend to external constraints, especially around the data, e.g., feasibility and cost of collection. Finally, solving certain harms is inherently in tension with other harms (e.g., impossibility results about various fairness metrics [442]). Many of these tensions are not accounted for in the literature.

Developers do not envision all these factors that might be at play when making a choice about a harm. For instance, some developers first chose a type of algorithm to build an ML model focusing on explainability power, and only later considered algorithmic fairness without questioning the initial choice, incorrectly assuming independence of explainability and fairness [90]. Besides, the tensions they account for are not all valid. For instance, nine developers wrongly envisioned the acontextual existence of a fairness-accuracy trade-off [192, 239, 526], especially because they did not reflect on data biases that might render measures of accuracy invalid. One developer considered a feature harmful to be used by the model, but argued for not dropping it, incorrectly believing they would not be able to monitor for output bias (incorrect as the training and test set can be different). All identified tensions are listed in Table 6.12.

PRIORITIZING AMIDST TENSIONS

Because of the tensions, developers have to prioritize certain objectives or harms. We do not identify the same priorities across developers. For instance, while some developers reported being ready to use smaller models and datasets resulting in less accurate models in order to reduce environmental or labor impact of model training, others judged model performance as the highest priority to optimize the model for. The thresholds of satisfaction for the different objectives also differed across developers, resulting in accounting for different objectives and harms to different extents. P27: *"In an ideal scenario, you want the system to be fully fair and accurate, but if you increase one, you decrease the other, and conversely. So we want to cut in* ***half*** *the burrito, like an optimal trade-off."* However, we did not find any precise criterion for the developers to judge the satisfaction of a system in relation to harms. Instead, they either relied on the judgement of other stakeholders (e.g., data subjects, model requesters, or domain experts), their own intuition and the amount of effort needed to handle the harms, or on comparisons with prior algorithmic or human baselines. Their prioritization was mainly informed by how important and severe they considered each harm individually, and relatively (when they perceived a tension), the feasibility and effort needed to address the harm, and various cost-benefit trade-offs (e.g., utilitarian view vs. libertarian view).

DEFINING VARIOUS GOALS

Developers who consider important to handle a harm, do not all take upon the same operationalization goals. Most of them undertook to mitigate the harm. Yet, others did not, because of other priorities and potential tensions, or the lack of (awareness of) method for mitigation. Instead, they mentioned keeping track of the harm (e.g., when a population is silenced if the corresponding records are erased from the data) as a memo to

carefully use the system (potentially working around the harm, e.g., by having human-decisions for the non-supported populations), or explicitly and solely deciding pragmatically not to address it. A last solution three developers proposed is not to deploy the system, or making the harm transparent for the decision maker to take such executive decision. *P1 "if you really need the mitigation approaches for the model to be accurate or have a good selection rate, you should question whether ML makes sense to use in this scenario." P6 "I would have this conversation with the hospital. I could say where we're confident and where we're not."* All goals are listed in Table 6.11.

6.4.3. ACTING ON HARMS: PLURALITY OF OPERATIONALISATION PRACTICES

A MORE COMPLEX WORKFLOW FOR HANDLING HARMS

Eight activities that ML developers perform specifically to handle harms emerged from our analysis, in addition to the usual ML lifecycle activities that can impact harms. Works [569, 920] that have identified ML activities do not highlight these activities, and even when they look beyond the inner loop of technical ML work (dataset collection and cleaning, model design and monitoring), they remain more abstract [454, 637]. These are 1) understanding the allocation of responsibilities and power relations within the project and organisation where they work in order to identify potential obstacles and needs for them to tackle harms; 2) envisioning potential harms the project might cause (subsection 6.4.1); 3) identifying invisible tensions and often first uncovering ambiguous objectives and external constraints that constitute these tensions (subsection 6.4.2); 4) prioritizing harms or other factors (subsection 6.4.2), and setting up realistic goals for each harm (subsection 6.4.2); 5) identifying, adapting/developing, and applying algorithmic unfairness identification and mitigation methods (see below); 6) identifying, developing, and applying strategies to account for the other harms ML models foster (see below); 7) actively warning the stakeholders empowered to deploy the ML model about the harms; and 8) working to develop re-usable toolkits and responsible AI processes within their organization (often voluntarily). Not all developers performed each step, e.g., as they would not necessarily realize the existence of tensions, would not feel concerned by harms in their systems, or would not have the desire or opportunity to design harm-related processes for their organisations — increasing the potential for harm of the ML system. Certain activities occur in different orders, = in iterations, e.g., 5) and 6) are often performed simultaneously, and potentially serve to update on 3) and 4). We now discuss 5) and 6) in more depth as they are crucial to ML harm practices.

A DIVERSITY OF APPROACHES FOR HANDLING HARMS

When developers decide on handling a harm outside the distributive fairness category, they either do not deploy the system, bring additional constraints onto the development process (e.g., on the dataset size, schema, or computational power), engage into additional data engineering and model engineering efforts (e.g., deletion or re-collection of data), or envision restructuring the learning task and the broader system design and interactions with users. To the best of our knowledge, such approaches have not been reported and studied in prior works in relation to algorithmic harms. When developers handle a harm related to distributive justice, they employ various approaches to iden-

tify, quantify, and tackle it, resulting in a variable mitigation of these harms. For instance, they considered one or multiple fairness metrics simultaneously, often selected among either group performance or group distribution, but sometimes among individual fairness (causal fairness metrics were only mentioned by one developer). *P2 "because this model will work in hospital with patients where fairness is important, we check all the group fairness metrics of FairLearn."* Similarly, for mitigating unfairness, they either proposed various manual or semi-synthetic transformations of the dataset, or applied different fairness mitigation methods across the three existing categories of methods. While most approaches revolve around data and algorithmic changes related to mitigation methods from the literature, some system design-level transformations are also proposed that are not extensively discussed in the literature (but might be effective). For instance, P28 brought the need to develop a different, more usable, interface for the decision subjects to enter their data (avoiding dataset under-representation from minority individuals not familiar with the technology or input language), five developers proposed to leave out under-represented populations from the dataset and model, and five others modeled a new learning task. *P6 "We actually have enough data that we might be able to train separate models. So you might not even use the normal FairLearn strategy, which is to train one model that works well across populations."* Table 6.9 and Table 6.10 list the ways with which distributive unfairness is identified and mitigated.

MIS-HANDLING?

Some approaches employed are not appropriate, either because they do not have the intended effects, or because they can cause new harms in certain contexts. For instance, in order to reach algorithmic fairness, three developers proposed to simply drop the sensitive attribute that presents unequal distributions, overlooking the limitations of "fairness through unawareness" [241] and especially the existence of proxy attributes that might skew a model. 30% of developers did not realize the (almost unavoidable) need for data sampling transformations to reach algorithmic fairness, not realizing the existence of potential measurement biases, or not envisioning that when the model sees too little data about certain groups, it might not be able to learn to make correct predictions on those groups. *P23 "Some of the bias comes by nature, like the data given the situation happening in the real world. That's not something you can change, it's happening by nature."* Other developers decided to aggregate data of different underrepresented groups to create a more equally-distributed dataset (in comparison to the majority group) without envisioning that relevant differences between these groups might still prevent algorithmic fairness [260]. Finally, other developers filtered out under-represented populations to reach parity across a smaller number of groups. This can lead to an even lower accuracy and harms for the silenced groups — an aspect several developers did not realize.

A DIVERSITY OF CRITICAL REFLECTIONS AROUND HANDLING OF HARMS

We investigated to what extent developers engaged in reflective practices, and observed epistemic or practical limitations of their process. Since most developers did not engage in such reflections, we prompted them directly. Most developers were not able to envision any limitation. When some did, the limitations identified matched those brought up by prior literature. For instance, they talked about conceptual limitations in accounting for individual differences when receiving wrong outputs or accounting for

the impact of the systems on non-decision-subjects stakeholders. For mitigation methods, they discussed that some approaches might not be considered ethical, or that they reflect techno-solutionist trends where the solution allows to reach parity in numbers but does not solve the societal cause of the problem. In the face of such limitations, the developers were often at loss in knowing how to react. They were not aware of better solutions and they reported to be satisfied with their approach, or chose a different fairness metric or mitigation method, without realizing that the alternatives were also limited.

6.5. DISCUSSION & IMPLICATIONS AROUND THE RESEARCH / PRACTICE GAP

6.5.1. POSITIONING THE RESULTS: RENEWED EVIDENCE OF DIVERSE AND FLAWED CONCEPTIONS AND PRACTICES

Our results highlight that there is no standard practice among developers. We identified a multitude of (mis)conceptions of harms, different ways to prioritize harms and other objectives, and to set concrete goals, and various potentially flawed approaches to quantify and handle harms. Our study complements prior works, and unveils novel insights.

CONCEPTUAL RESEARCH ON HARMS

Theoretical works have demonstrated results about algorithmic fairness, tensions [105, 315, 397, 748], and ML lifecycle activities as a cause of unfairness [256, 726, 878]. Others have formalised issues around flawed assumptions made by developers or researchers [846, 731, 630, 313, 492], and have discussed the underlying philosophical theories of different algorithmic fairness tools such as the conceptualised opposition between two visions of the world (WYSIWYG and WAE) reflected in datasets [267, 870]. These works are aligned with our findings: they bring rigorous frameworks to describe (mis)conceptions that our participants fell into. For instance, apparent trade-offs between group and individual fairness metrics [105] or distributive and procedural fairness [315], or between accuracy and algorithmic fairness [397], are in reality nonexistent in certain contexts. Yet most participants believed in a strict opposition between objectives as they did not understand the intricacies of these objectives and relevant approaches.

We could not identify literature to characterize each conception, prioritization, and handling approach we identified, especially around algorithmic harms beyond distributive questions. We suggest to investigate each finding independently, e.g., by conducting empirical studies, theoretical proof-based works, or conceptual reflections, to better understand their ins and outs, and advise developers. Our results outline a multitude of factors that are unspoken in the research community, e.g., conflicting ML performance, infrastructure requirements, or external data constraints (except the conflicting business/developer goals [621, 521, 612], and lack of metrics and mitigation methods for certain contexts [369]). As these factors are inherently in tension with algorithmic harms, they unavoidably have to be accounted for by developers, and further research is required to better understand how to make choices around these seemingly unsolvable tensions.

EMPIRICAL RESEARCH ON PRACTICES AROUND HARMS

Our results validate, corroborate, and extend empirical works that have identified flawed perceptions, practices, and obstacles around algorithmic fairness. We bring further confirmation and hint at generalizability of the results as we performed our study on different use-cases, with different, and larger numbers of participants. For instance, the problematic belief developers have in fairness through unawareness [220], the subjectivity of choices based on personal experiences [220, 666], and misconceptions towards fairness metrics [175] had already been pointed out in different contexts. Besides, Muller et al. [570] have surfaced harmful forgetting practices in a multitude of activities of the ML lifecycle, discussing for instance data silences and the flawed WYSIATI ("What You See Is All There Is") assumption. These literature-based results on biased data work echo our empirical finding that developers routinely conduct data activities that might erase certain populations, and decide not to deploy the ML models for these populations, without deeply reflecting on the consequences for them.

Our results also extend the list of (mis)conceptions and (mis)handlings to broader algorithmic harms, where there is insufficient guidance for developers. We could not find any work that investigated in detail practices towards harms beyond distributive fairness. Yet, we note great similarities in how one approaches the other harms and distributive fairness, or more generally how the data science workflow is approached by developers, via constant negotiations between amorphous objectives and unambiguous problem formulations without explicit normative considerations [619]. Besides, only few works have brushed over practices around the gap between distributive fairness and algorithmic fairness, while our results delved deeper into identifying the traps and good practices participants fall into or follow. These traps were discussed in prior conceptual works [731], and now observed in practice: the formalism trap that directly refers to the gap, and the framing trap that relates to the other harms, differently handled across developers. The portability and ripple effect traps were not envisioned by most developers, yet a few mentioned the possibility for their models to be dangerously repurposed, or discussed how the original system functions in terms of decisions and what kind of shared control should be established between the decision maker and the model. The solutionism trap was solely considered by a few developers when they referred to the technical solutions to algorithmic unfairness not being enough, e.g., they mentioned that identifying biased outputs indicates a deeper structural issue that cannot be solved through an ML model. The sociotechnical system (STS) lens [731] proposed to circumvent the traps should be revisited to better guide developers while accounting for our new insights on their practices.

6.5.2. WHERE DO WE GO FROM HERE?

Our findings corroborate the lack of standardization observed in ML practices around algorithmic harms. It is tempting to argue for more standardization to curtail the perpetration of harmful models, by enforcing specific definitions of concepts, clear harm prioritization and handling approaches, and a uniform source of information for practitioners. These are the advantages brought about by attempts at standardizing ML processes (e.g., MLOps [20, 819]) or algorithmic fairness processes specifically [8]. Yet, is it really possible to standardize processes? The problem is socio-technical and complex

due to its context-dependence and recency. HCI research [136] has argued that standardization cannot account for the issues and constraints imposed by the context of the work. And this has been demonstrated by the mild impact AI ethics standards seem to have [945]. It is not possible to propose definitions or metrics for all factors to account for, nor approaches cognizant of all these factors, but the involvement of various stakeholders is evidently necessary.

SUPPORTING REFLEXIVITY

Instead of standardizing, the research community should support flexibility of practices, and invest efforts into changing the mindsets of developer, and particularly foster contextualisation and reflexivity activities [548, 156], which are not commonplace. Future research can explore how to train developers in performing such activities, or how to guide them specifically around harms and the design choices they have to make. Collaboration with more adapted practitioners should also be encouraged. For instance, when a tension occurs, interdisciplinary insights are needed to understand how to prioritize factors, e.g., by uncovering and negotiating preferences of different stakeholders [836, 146, 484], and when not to deploy models. It is now well-known that collaboration in the ML lifecycle is often needed [220, 920, 637, 454]. In this study, we also found that diverse stakeholders were sometimes involved in the activities we synthesized, e.g., to identify inappropriate attributes and task encoding, or to select fairness metrics and define satisfactory thresholds. *P6: "There's a question of what is an acceptable difference in performance. It's a difficult question to answer, and that's something you talk to all the stakeholders about."* Yet, prior work has shown that tackling questions around algorithmic harms is still predominantly the job of ML practitioners alone [920, 521, 882], a finding that was echoed in our study. For instance, the elicitation of requirements around algorithmic harms and their prioritization were typically left out from stakeholder involvement and the stakeholders themselves did not proactively involve themselves [637, 803, 846, 803] whereas it could be considered their responsibility. We join the recommendations of such prior works that call for facilitating collaboration between stakeholders in these new socio-technical activities.

GUIDING VIA ALGORITHMIC FAIRNESS TOOLKITS

We acknowledge recent debates arguing whether ML developers are the right individuals to address socio-technical problems of ML (myth of ML practitioners as "ethical unicorns" [662]). Recognizing that ML developers are thrown into this role, we emphasize the pressing need for future work to better support activities of these developers. Since fairness toolkits are one of the primary sources of information and tools developers use to handle harms, but considering that we still identified flawed conceptions and practices, we propose to extend existing toolkit rubrics with broader and deeper considerations of algorithmic harms to guide developers better. Aligned with the idea of AutoAI tools as collaborators in organising a work plan [862], existing fairness toolkits could remind developers not to leave out certain harms and to consider the new activities we identified. Toolkits could also aim at facilitating the identification and proper handling of these harms with awareness of the limitations of algorithmic fairness concepts. One could potentially leverage the (mis-)conceptions we collected, and the attached theories, in order to outline anti-patterns to avoid at each step of the ML lifecycle. Enhancing

toolkits will be a challenge as existing warnings in FairLearn [107] are not always considered by the developers. Besides, one should keep in mind that toolkits should augment ML developers but not automate their work [862, 882], and remain cautious not to make the toolkits checkboxes, but instead make them foster critical thinking on topics for which knowledge is still incomplete. There is an opportunity to develop other tools than toolkits, such as guidance frameworks [915].

INVESTIGATING THE DEEPER FACTORS THAT IMPACT PRACTICES

While we identified the landscape of (mis-)conceptions and (dubious) practices around harms, we did not investigate what causes such a diverse landscape. This would be necessary in order to understand where to focus our efforts in the future, e.g., in fostering developers' due diligence through education [220, 719, 454], or enforcing structural incentives via the establishment of organisational processes [666]. Particularly, from the analysis of the interviews, many factors emerged, around developers' perceptions and knowledge of harms and potential solutions (e.g., only P4 and P30 admitted their lack of knowledge P4: "*For hyperparameters like learning rate, I can't see the connection with harm: it just influences accuracy. But I'm hesitant to say it doesn't affect it at all because you never know with these things.*"), and their aptitudes and attitudes for reflecting about them, that are themselves impacted by numerous individual traits and the developers' environments (e.g., incentives from the organization, available mitigation tools, etc.). Despite all these factors being relevant, and considering the diversity of developers we interviewed, not all of these factors have been studied thoroughly in existing literature. Some of our findings also differ from prior works. For instance, Deng et al. [220] asked developers without experience in algorithmic fairness how they would evaluate the fairness of their model given a fairness toolkit and a college admission scenario. They found that these developers recognize the limitations of their knowledge and wish to receive help from domain experts, which was only the case for a few of the developers we interviewed on a different scenario and with varying experiences with fairness. Hence, identifying potential factors and potential impact on harm conceptions and practices, and then quantitatively studying those is a meaningful direction for future work.

6.6. LIMITATIONS & THREATS TO VALIDITY

Despite leading to a large amount of insightful results, our experimental setup bears limitations that might hinder the generalisability of our findings. While we strived for recruiting a diverse set of participants in terms of demographics, experience with ML, and algorithmic fairness, it was not possible to obtain a larger sample for each category. Several of our observations, however, corroborate findings from previous studies, hinting at their validity through methodological triangulation. Yet, focusing on other domains and less-represented segments of population using targeted recruitment methods would be important in the future. In terms of interview sessions, not all developers had the time to answer our entire set of questions for each harm identified prior to the interviews. Hence, we cannot provide quantitative results around unreported harms (due to not having time, forgetting, or simply not considering it an harm). Finally, we acknowledge our own unavoidable subjectivity in identifying and characterizing potential harms and flaws.

6.7. CONCLUSION

Our study represents a testimony of the constant socio-technical negotiations [619] needed to build and deploy a machine learning (ML) model. Our results echo previous studies on algorithmic fairness, and complement them with new evidence of the complex and potentially worrying state of ML practices around broader harms. We contribute to the effort of triangulation of results in HCI research [520] and especially in the relatively recent field of ML practices. Particularly, our results contribute a deeper and more comprehensive understanding of the (mis)conceptions and (mis)handling around algorithmic harms, which calls for theoretical, design, and HCI works to ultimately guide developers in the development of unharmful models. We believe that the FAccT community is uniquely positioned to tackle these challenges with the growing interdisciplinarity of research efforts, an indisputable pre-requisite to make real progress. In the next chapter (Chapter 7), we investigate the factors that might lead to the fragmentation of conceptions and practices identified across ML developers. Knowledge of these factors is extremely important to understand the deeper reasons for the subsistence of certain limitations and challenges identified in this study, and to propose additional remedies to such limitations and challenges.

ADDITIONAL DETAILS AROUND RELATED WORKS

Fairness toolkits. To facilitate the adoption of metrics and mitigation methods, various companies and public institutions have built fairness toolkits. These toolkits are typically code repositories that allow an easier implementation of the metrics and methods. Examples of these toolkits are FairLearn [107], AIF360 [91], Aequitas [704], Lift [842], Themis-ML [77], ML-Fairness Gym [205], TensorFlow Fairness Indicators [892], etc.

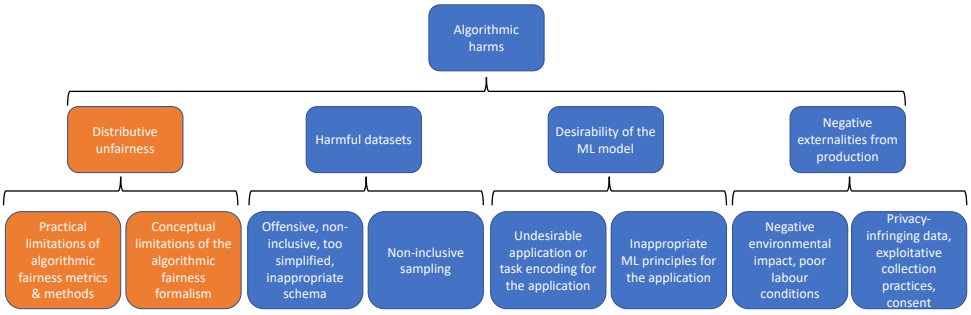


Figure 6.2: Taxonomy of harms investigated in our study. In orange we represent **the limitations of algorithmic fairness**, i.e., the current, flawed, solution to distributive unfairness, and in blue we represent the **other types of harms.**

ADDITIONAL DETAILS ABOUT OUR METHOD

DETAILED DESCRIPTION OF PARTICIPANTS' BACKGROUND

Cf. Table 6.2.

Table 6.2: Background of the participants in our study. Note that some participants reported multiple educational backgrounds.

Dimension	Values (and number)
Demographic information	
Nationality	US (6), Netherlands (6), India (4), Iran (2), Russia (2), Romania (2), Sint Maarten (1), Canada (1), Brazil (1), Slovakia (1), Poland (1), Greece (1), Spain (1), Ukraine (1)
Gender	male (24), female (6)
Highest education	BSc (2), MSc (21), PhD (7)
Experience with machine learning	
Work type	applications (14), research (8), both (8)
Application domain	healthcare (4), finance (3), recommender systems (related to human resources) (3), predictive maintenance (1), others
Education	computer science (25), mechanical engineering (3), business or economics (3), sociology (1), psychology (1), accountant ethics and compliance (1)
Years of experience	2 or less (13); 3 to 5 (15), 15 (2)
Experience with algorithmic fairness	
Years of experience	18 (1), 3 (3), 2 (7); 1 (2), 0.5 (7); 0 (10)
Type of experience	long-term research (6), short-term research (4), frequent use (7), irregular use (3), none (10)
Toolkit	no exp. then FairLearn (5), no exp. then AIF360 (5), exp. with FairLearn (11), exp. with AIF360 (9)

DETAILED DESCRIPTION OF THE QUESTIONS ASKED TO THE PARTICIPANTS DURING THE INTERVIEWS

Questions on background experience. We started the interviews by giving a brief overview of our research to the participants, and by questioning them about their background (demographics and machine learning experience). Once all required tasks were completed by the participants, we asked final questions about their fairness experiences, how they learned and work with algorithmic fairness/harms, and reasons for using a certain toolkit, as well as their broader knowledge of the responsible machine learning field. We made sure not to ask any question related to their algorithmic fairness experience at the beginning of the interviews not to bias them towards thinking of particular topics.

Questions on higher-level reflections. At the end of the interviews, we also asked general reflection questions about any other considerations they might have when building models, any additional harm they could envision, their experiences with the fairness toolkits that we had introduced (for practitioners who previously did not know these toolkits) and potential changes they would like to see in these toolkits, about algorithmic fairness and whether it can be solved as well as on the limits of fairness metrics and mitigation methods (when not mentioned earlier), about their responsibility in considering algorithmic harms, and about any other wish, doubt, or remark.

Questions on the process. While the participants were working on the tasks, we asked them questions about their process, in order to understand the reasons for performing each exploration activity, the thoughts they had when seeing the results of an exploration, and the actions they would take based on these results, as well as to make sure they had not forgotten any activity. We especially questioned them on activities that might have a connection to algorithmic harms (e.g., observing data distributions and rebalancing the dataset based on the target labels). After the two tasks in the case of the participants inexperienced with toolkits (not to bias the participants towards certain reflections when looking at the second task), and after the first task for the other participants, we further questioned them on the algorithmic harms they had not investigated (whether they usually consider them, why or why not, how they would handle them) during their exploration of both tasks, and on the harms that could be resulting from the activities they mentioned. We identified the harms we posed questions on through our analysis of the literature subsection 6.2.1 Table 6.1, and we also coded any other harm they could mention. We made sure to first ask vague questions (e.g., what can be issues with the activity of labeling data with crowd workers), before going onto more specific questions (e.g., what do you think of potentially poor labor conditions of crowd workers), so as to see to what extent the practitioners actively think about these harms.

OTHER MATERIALS

Tutorial. The tutorial consisted in presenting the concept of algorithmic fairness, the ways different fairness definitions are computed and different mitigation methods are applied (concepts of data pre-processing, model in-processing, and output post-processing), as well as illustrating the use of one of the toolkits to apply these definitions and mitigation methods. We gave the tutorial with a third use-case dealing with the prediction of credits default [363, 908]. This use-case was chosen for its popularity within tutorials on algorithmic fairness and toolkits, so as to be as close as possible to what a machine learning practitioner might see first when learning about algorithmic fairness.

To give the tutorial, we shared our screen with the participants, showing a Jupyter notebook we had prepared with these concepts and examples of application of the tools on the credits default dataset. We especially presented the computation of some of the metrics on a simple logistic regression classifier, and on the same classifier to which various mitigation methods (e.g., the threshold optimizer and grid search algorithms of FairLearn, as well as the reweighing and prejudice remover algorithms of AIF360) are applied. We made sure to answer any question the participants had during the tutorial and later when provided with their second task. At the end of the tutorial whose aim was to give the participants a basic introduction to algorithmic fairness and toolkits, we asked for verbal validation from the participants to confirm we achieved our goal.

Notebooks. When working on these tasks, we made sure to reassure the participants that they did not have to code the entire exploration they would perform (only if they wished to), but they could also simply speak out-loud and report on what they would do. We had already prepared additional notebooks with code snippets that the participants might want to use, and we shared these snippets with them whenever they would mention a certain exploration activity that would correspond to the snippet. This allowed to

reduce the complexity of the session for the participants, to accelerate the process, as well as to see them reflect about concrete results of the exploration activities.

Pilot Studies. Before performing the interviews, we performed two pilot studies with practitioners working at our institution. These two studies allowed us to check for the understandablity of the tasks, to refine our questions to prompt about the different harms, to better time each task, and identify relevant reflection questions, as well as to make sure that we had prepared enough code snippets to help the practitioners.

RESULTING THEMES AND CODES

The coding process resulted in 13 high-level code categories (e.g., data schema considerations) with 3 to 6 intermediate levels of codes within each category (e.g., sensitive attributes, inappropriate attributes), and 8 to 34 finer-granularity codes (e.g., automatic or expert-supported identification of attributes) that represent the different response declinations. In total, this represents 276 finer-granularity codes (summarized in Figure 6.3).

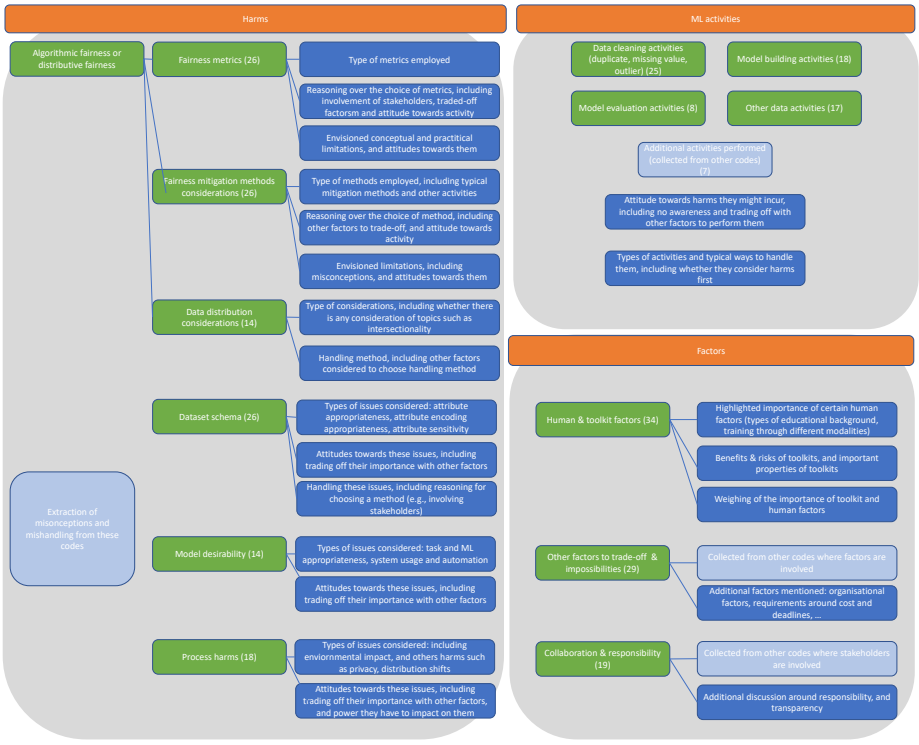


Figure 6.3: Summary of the main themes (in orange) resulting from our study, the main categories of code (in green with the total number of code per category in parenthesis), and the main relevant results (in blue). We show (in light blue) how we surfaced new themes from certain categories of codes.

DETAILED RESULTS FROM THE CORE OF THE MANUSCRIPT

ON ALGORITHMIC HARMS

We list in Table 6.4, Table 6.5, and Table 6.3 the different categories and sub-categories of harms discussed by the developers.

Table 6.3: The various conceptions of one macro-category of harms: around the ideal output distribution (i.e., distributive fairness). We do not include when the developers are not aware of or lacking precise information to discuss the harm, as this applies for each of these harms.

Harm	Conception	Example
Output distribution (distributive fairness)		
Ideal distribution & distributive fairness	No mitigation because the data represents the world (unfair or not)	*P23 "some of them come by nature, like the data given the situation happening in the real world. So you get that bias into data, and that's not something you can change actually, it's by nature happening."*
	Distribution representative of the real population	*P5 " what is the statistical characteristics of the real world scenario and what are the statistical characteristics of the scenario that you see here. When I say statistical characteristics, I'm actually speaking about this set of data across parameters. I focus on protected category variables."*
	Equal accuracy across sensitive features via equal distribution	*P28 "if you want to have the same probability of giving a correct answer for all societal groups, you need to be training with the dataset that is one divided by the number of social groups that are considered."*
	Middle ground: none of the two distributions is feasible to collect	*P11 "For all of these distributions, I would consult either a specialist or literature from medicine to see from all the hospital patients or just diabetes patients: does the distribution look somewhat like that?"*
	Ambiguous judgement of acceptable slack	*P28 " I would say the data static between female and male is quite balanced. You can try to make it 50, 50, but it might be the case that make it 50, 50 doesn't change much in the accuracy of the whole model because it's quite similar the number of data points."*
	Acknowledging historical biases in joint distributions	*P2 "I would also look at the selection rates in historical data. Has it really been unfair in history? And do we have to fix?"* (P2, P11, P12, P20, P21, P23)
	Rare consideration of intersectionality	*P21 "checking whether we have any groups that are specifically underrepresented if we take a look at the combination of the demographic features, that's possibly something to take into account."*

Table 6.4: The various conceptions of two macro-categories of harms: around the desirability of the system and the development process. We do not include when the practitioners are not aware of or lacking precise information to discuss the harm, as this applies for each of these harms.

Harm	Conception	Example
Desirability of the system		
Goal of the system	Broad ethical considerations (society)	*P28 "We need to look at the bigger picture to see if our work is ethical. And that can go for the carbon footprint, the sustainability, the impact this may have in the labour market, and in warfare."*
	Morality (society)	*P17 "That's a big problem. Everybody as they get older, they have more health costs, so that'd be price gauging, the hot button issue of building based on pre-existing conditions. For health insurance, I think that's unethical."*
	Utility for the organization	*P16 "It's appropriate and relevant for the business. They want to save money or to reduce time of the workers."*
	Impact on organization	*P25 "even the organisation where the model was employed might be affected."*
	Impact on society and "silenced" individuals	*P6 "we might ask what are the consequences of some people having access to this model and others not? Some might say this will have knock on effects in a broader scope where there are bigger consequences, where people of some descent might not trust us. So in the overall picture, it's a harm to society for us to deploy it."*
Employing ML	Appropriateness/ethics	*P1 "I would question whether we should be using ML at all? question all the assumptions that are being made."*
	Complexity & flexibility	*P3 "Everybody is afraid of changing something [with deep learning models] because if you change this, it breaks this. So we start with: what was the problem you are trying to solve? could it be solved by simple query or simple statistical model, or by business rules and statistical model? If not, by machine learning? It's not about amplifying the buzz and having AI everywhere. It's about the real value of using it."*
	Right to explanations	*P27 "at least if a computer tells the person you're not getting a loan, explain why."*
Automation mode	Removing human bias collaboratively	*P27 "cause people can also have biases. It should be a doctor and in addition, this model. I don't think we should just believe the output of the model, but things should be used hand in hand with an expert."*
	Suggesting to human decider	*P4 "It's possible to automate, but it's not wise to let the model do all the work. It's important to have another medical professional opinion."*
	First filtering tool	*P29 "Do I think the hospital can fully automate this? No, I think you can use it as a recommendation or triage tool. You don't have unlimited healthcare resources, unlimited doctor availability, so it's sort of a triage."*
Task design	Meaningfulness	*P1 "Think whether the problem was formulated in a way that makes sense, for example why is 30 days the cut off? Was it just chosen out of the data?"*
	Alignment with goal	*P17 "A better way would be pay per probability, so if there's a 0% chance they're getting re-admitted, we're going to pay you more, but as there's like a 50% chance, we're going to pay you a little less, and 100% chance, we'll put the full penalty."*
	Informativeness	*P17 "we're just trying to classify you and say "are you someone that is going to use a lot of health care services or not?" I wouldn't do it this way. You're not going to get a lot of information. I'd rather use a regression."*
Development process		
Labor	Crowd exploitation	*P1 "Crowdsourcing is very important from an exploitative point of view."*
Environmental impact	Only around training	*P8 "You need a big amount of CPU time, GPU time, to train a big model. It's bad energy-wise."*
	Training and inference	*P15 " it is a very big growing problem in the whole computer science community because you have these very big models like GPT 3 which all the big companies are doing. But then you need a whole lot of compute power for them."*
	Only for large deep learning models	*P9 "From my understanding, that only happens at the scale of a really large language model, the things which literally have like trillions of parameters."*
	Balancing with benefits of the application	*P4 "I have thought about this in terms of climate AI. I have read that training a model to tackle AI is actually counterproductive because it harms the environment."*
	Scale: Not relevant as models are beneficial	*P2 "I wouldn't consider that. I think automating anything would make stuff more efficient, so I think it would save energy somewhere else."*
	Other systems are worse	*P8 'There are better ways than reducing model training to improve environment."*
Privacy	Consent for data use	*P18 "You need to make sure that everyone is ok with data being collected and used."P19 "look at whether the Clients are OK with their information being shared."*
	Anonymisation of data subjects	*P7 "Since the data are not publicly available, we need to take care of masking the data set not to release any personal information, not to release any sensitive information within the training."*
Team	Resource sharing	*P6 "This was a university cluster that we shared with others. I didn't want to hog the whole cluster for myself."*

Table 6.5: The various conceptions of one macro-category of harms: around the dataset schema and its population. We do not include when the developers are not aware of or lacking precise information to discuss the harm, as this applies for each of these harms.

Harm	Conception	Example
Dataset schema		
Feature desirability	Relevance through causal relation or correlation	*P5 "I would primarily try and understand what's the merit in using these numbers. Without a specification on the positive correlation to, or the causality link to the outcome, it may not merit directly being used."*
	Use-case dependence	*P1 "This is tricky because it may or not make sense depending on what you're using this model for."*
	Acceptability as proxy	*P1 "it would be better to have a feature for your socioeconomic status. But race could be a proxy."*
	Completeness	*P13 "My first thought would be that the dataset doesn't have a bunch of information regarding the patient exams. I think it would be cool to include it to be more precise regarding the target feature."*
Feature sensitivity	*Sensitivity based on:* * Regulations	*P7 "In the credit adjudication use-case [..], one of the regulations was that the sensitive features should not be used as a predictor in the training of the model."*
	* Ethicality (sensitivity, relevance, offensiveness, privacy)	*P13 "If I use gender to try to predict something that is not related to gender, for example whether this person would be a good employee, the sensitive features to predict these labels, that would be bad."*
	* Exception if causally related to target label	*P13 "I don't know if race or gender is important to predict the diabetes. If this feature would be important for this problem, it wouldn't be a sensitive feature.".*
	* Exception if causally related to target label and volitional	*P17 [looking at dataset features: e.g., demographic, military service, employment, poverty status, heart diseases, etc.] I wouldn't want to be biased on any of them. The only one that society has said it's OK to be biased on is smoking because it is probably the only one on which you can make a conscious decision."*
	Confusion with privacy infringement	*P15 "I would think that there are personal information. I mean their history, their age, gender and all those things apart from the things that hospital needs to note down."*
	Confusion with a parameter of a tool that would (magically) avoid discrimination	*P30 "Marital status and region: those are things that could be removed. And protected that would be more the tricky ones like sex, employment status. I'm curious to see if there will be a difference between protecting a sample and removing it."*
	Forbidden to: * use	*P7 "The sensitive features should not be used as a predictor in the training of the model."*
	* receive high model feature importance	*P2 "I would check which coefficients have the highest weight. Just to see on what attributes is the model predicting on, And those shouldn't be the sensitive attributes."*
	* display model output disparity	*P12 "your boss just asks you to make a classifier that works fairly for some feature."*
Encoding meaningfulness	Sensitive proxy: attribute correlated to a sensitive one	*P3 "Getting back to the financial use-case, if you know the ZIP codes, it could be really sensitive features as well because ZIP code could predict for example your economic status."*
	Sensitive proxy: not accounted due to impossibility to "unbias" the model for all attributes	*P21 "We are going into territory where fairness becomes almost impossible, because it could well be that Medicare and Medicaid are a proxy for demographic features: whether minorities are, for example more likely to take Medicare and Medicaid."*
	Silenced "values" (i.e., individuals)	*P15 "You would also have other races, there's not just two races. Then those kind of communities, for instance. Also for gender, I would say that to include more other genders."*
	Doubtfully aggregated values (incorrect representation)	*P20 "It's white and non white here. From the start, it's a bad feature. The people that are not white also are different between them. This should have been a category feature with all the races that are here."*
	Informativeness of values	*P27 ""Other" isn't really informative here. You see, ideally you don't want other and missing and all that. Those kind of values in your data. This is really not informative."*
	Correctness of values	*P1 "Let's look at the race column. We have mostly Caucasians, a bit of African American, unknown, Hispanic, other, Asian. Always interesting to see how race is Hispanic: that's not a race, it's just false."*
	Concept representation & measurement errors	*P1 "I would want to know how this data was collected. Like who determines the race and gender columns?" P24 "I will try to understand what each column means, and whether or not there have been mistakes in encoding the data and maybe reach out to the people responsible and say hey, what's up?"*

ON THE ACTIVITIES OF THE MACHINE LEARNING LIFECYCLE

Table 6.6: Summary (part 1) of the ways the activities performed during the machine learning lifecycle are conceived in relation to harms (in green) and other trade-off (in grey), and handled (in red).

Activity	Conception	Example
Data duplicates	No envisioned harm	*P10 "I would delete one or the other, because I don't think it would make any effect."*
	Percentage of duplicates within dataset	*P4 "It's important to have them because they represent the distribution. But it depends: if there's a lot of the same occasions, you might want to trim it down a bit."*
	Removing duplicates in any case	No awareness of the different natures of duplicates (real or apparent) *P10 "I would delete one or the other, because I don't think it would make any effect or any changes."*
	Understanding the nature	*P2 "it depends also on the use case. Why are there duplicates? How do those duplicates get into the data? It could be really similar people and then you would leave them ."*
Data outliers	Cause of dataset biases and algorithmic unfairness (only P5)	*P5 "I would be cautious of eliminating outliers as it can cause bias. I would focus on statistical characteristics to know what's the proportion of outliers. If the outliers are related to one of the variables, I would consider whether to eliminate it."*
	Cause of population silences (only P21)	*P21 "I would look at whether we have any important outliers in the data. What could be a problem is say you know five people in this big dataset of 100,000 records spent in hospital 100 days and you know all the others spent less than 20. Then you know the question would be whether the model that I built is at all applicable to such people. I would say probably not so maybe it's best to remove records which seem to have very strong outliers. And have that caveat that you know the model shouldn't be applied in some very rare cases."*
	Indirect sign of deployment issues, in turn causing potential algorithmic unfairness (only P6)	*P6 " it is useful to see if there are outliers, as a way to detect if there is input issues. If someone is listed as being 10 pounds, then you know that's an issue where someone entered it wrong and then I would look at why was this entered in wrong? Is there a manual process somewhere that this is the result of? Now that I've been confronted with this fact that there's manually entered data, then I'd have to go back and think about what are the consequences of that at inference time?"*
	Dataset size, impact of removing outliers on model accuracy with or without experiment	*P28 "deleting points just because they are outliers, that's not the right approach, because those outliers could be those that have the most information, while the ones that are located in the median in this case, or the mean, they are more common and provide less information."*
	Understanding provenance to handle outliers	*P2 "If you have weird outliers, I would look at those rows because they're often something parsed wrongly. Then you can remove those. If there's enough data and there are some outliers, they could just be outliers, so we would keep them in."*
	Adopting one of the three default approaches in any case	*P18 "If we're talking about use cases where the outliers are really purely of an anomalous nature, you can just get rid of them. For example, having a person in our data set being 400 years old. Well, that's to my estimate, at least unlikely."*
Missing values	Causes dataset biases and algorithmic unfairness (only P21)	*P21 "I wouldn't drop them. People from specific backgrounds are less willing to answer some demographic questions. For instance, people from some minority group would be less willing to admit that they are using state insurance. If not dropping, I would say imputation. That depends how much time we have."*
	Silences populations	Only P29.
	Depends on dataset size	*P2 "Depends on how much is missing. I would impute it if there's not a lot of data missing."*
	Stakes of the system	*P15 "If this problem is critical, I would not introduce averaging or some interpolation for imputating the missing data, because it has to be as accurate as possible."*
	Handling by dropping records or imputing them or dropping attribute, depending on other factors, or taking one default approach	*P11 "I would look at which columns have excessive amount of missing values like one third, then I would remove this variable from the dataset. After removing columns that have a lot of missing values, I would remove all rows that have missing values so that this dataset has no missing values. The data is quite big (over 100,000 records), so if we have to remove two or three variables with missing values and then we will remove all other rows that contain any NaN, we still have quite large datasets."*
Data distribution shifts	Ensuring the populations seen in deployment are represented in training	*P15 "Is this really representational of the general situation of diabetes? For instance, sometimes these things are taken from very specific hospitals, very specific region, and that region might have very specific distribution of diabetes. It's not representative of the entire country."*
	Ensuring the model is adapted to any distribution shift happening after deployment	*P3 "Usually, the biggest problem is a huge difference between production and training data. When you get more sensitive medical devices, the way the data is distributed also changes, because the bad quality medical devices will have much more noisy data and if you optimise everything and re-calibrate to make sure that this data will be processed in this way, then you will be literally fucked up if the quality of medical devices will be better."*

Table 6.7: Summary (part 2) of the ways the activities performed during the machine learning lifecycle are conceived in relation to harms (in green) and other trade-off (in grey), and handled (in red), potentially influenced by other factors. Overall, participants do not envision harms from these activities.

Activity	Conception	Example
Preprocessing	Cause of dataset biases and algorithmic unfairness (only for data splitting, data label rebalancing –P1, P29, P30–, and data annotations)	*P5 " Training-test split, I would prefer to make it absolute, looking at it in terms of proportion. The split is going to be random and the split may not be an unbiased split, so that is something that I would standardize." P11 "if we have this re-admit that is a false negative committed by the humans that decided. That's exactly what you want to avoid that the model repeats this behavior. If this proportion fits with what medical experts say, then it might be fine. It's like a cognitive bias, so I would look at these kinds of variables. And make sure that it's all representative and makes sense to experts."*
	Accuracy and data-model compatibility	*P25 "There are algorithms which take both. You can input the range value and then feed categorical data. Otherwise, these range values need to be converted into categorical manually."*
Data labeling	Impact on model accuracy /no harm	*P15 "Labels are very important: the source of annotation can be noisy. The label itself can be noisy, so there can be misinterpretation of: OK I am a labeler and how do I interpret this?"*
	Cause of dataset biases and algorithmic unfairness (label unavoidable subjectivity)	*P20 "This is a very important source of bias, because if it's not something objective like doctors looking at X rays but something like insurance, and people manually label this based on their experience, they're 100% introducing bias. Maybe someone which is a minority would take into account bias more. But anything that is subjectively labeled is inherently biased. Because I think all the people are inherently biased."*
	Label "quality" vs quantity	*P9 "There is a very large graph of everywhere that you can have a fairness issue in a machine learning pipeline and labeling was one of them. So you have to decide for yourself whether the possible biases of the people labeling your data are more important."*
	Improving "quality" with the labelers	*P24 "I acknowledge that there can be labeling bias. And this is again Specific on the case. in the hospital, I think I would reach out to the doctors who actually labeled the patients ."*
	No action due to unavoidable subjectivity	*P5 "I need a comfort on the quality of data. Once I have a reasonable comfort, I'll go ahead because there's no end point to trying to understand data labeling or data annotation, there will always be bias in it."*
Model building	No envisioned harm	*P25 "In terms of building the model, considering fairness? Didn't we consider all of these things already? like we removed all the features, stuff like that."*
	Harms only come from data	*P2 "I don't think that giving a parameter a certain value can lead to harmful implications. I think it's mostly caused by the data, not really by the model."*
	Cause of algorithmic unfairness	*P5 "there may be models where you choose hyperparameters. And the choice may induce bias. I would do a grid search for all combinations of my dataset/model. And run them to know which has a higher propensity of bias. There may be impact caused by multiple other factors including the batch size, the epochs, the learning rate".*
	No awareness but benefice of the doubt	*P4 "For hyperparameters like learning rate, I can't see the connection with how it might harm people because it just influences accuracy. But I'm also hesitant to say it doesn't affect it at all because I feel you never know with these things, so you should always be cautious."*
	Accuracy, model explainability, privacy, expected output type, cost of training, easiness of maintenance	*P3 "For me, the simpler is the model, the easier it will be to deploy, the easier it will be to monitor, and the easier will be to retrain. So if there is a choice between doing something with deep learning and doing something with logistic regression with properly engineered features. I'm gonna go with logistic regression, because it will be just easier and less expensive to run in prod."*
	Algorithmic fairness as the second stage of model building	*P9 "The first iteration will always be to investigate even the feasibility of the accuracy, 'cause the second you start trying to incorporate other things like privacy or fairness into your models, you will immediately start making accuracy tradeoffs like in privacy."*
Model evaluation	Meaningfulness of the learned features	*P2 " I would check which Coefficients have the highest weight. Just to see on what attributes is the model predicting on? And those shouldn't be the sensitive attributes."*
	Algorithmic fairness when the use-case is sensitive	*P9 "when we talk about automating a task, you can create an arguably false dichotomy between sensitive tasks and insensitive. For example, you're going to pay far more attention if you're trying to automate something in college admissions, versus trying to use machine learning to automate the protocol for handwriting recognition."*
	Fairness when people involved	*P2 " when the use case is about making decisions for people, and especially when it's for demographic of people. Fairness issues can really disturb groups in society."*
	No fairness	Algorithmic fairness not mentioned during the evaluation.
	Accounting for algorithmic fairness implicitly without knowing the concept	*P28 "accuracy is only a certain perspective. The performance of the model can say it's 99%, but it's not telling you how accurate it is for different groups of society. Perhaps, for instance, it could be very inaccurate for African Americans, very accurate for caucasian, and that's not reflected only in accuracy."*
	Representativity of the test set	*P6 "When we evaluate accuracy on subgroups: do we have enough data to say that we have that accuracy? False confidence is a big danger."*

ON ALGORITHMIC FAIRNESS

A comprehensive analysis of the concepts related to algorithmic fairness can be found here, with summaries of their practices related to fairness metrics in Table 6.9, and fairness mitigation methods in Table 6.10, as well as how they handle via (simpler) approaches sensitive features and data distributions in Table 6.8.

Table 6.8: Practices around data issues towards algorithmic fairness (sensitive features and data distributions): simple approaches to identify and to handle them (in grey explicit trade-offs).

Conception	Example
Identification of sensitive features	
Mandatory according to external entity (guidelines, regulations, client, model owner)	*P11 "I know that these are legally defined. So, the EU for example, has a guideline on what are sensitive attributes. I will look at that as a baseline. Anything that's in there is protected or sensitive.*
Based on the existence of human discrimination on certain attributes	*P11 "Weight: obesity is common among people that have diabetes, so people are misjudged by doctors if they are comparatively thin."*
Based on intuition	*P16 "I would say, the most obvious sensitive features are race and sex. But also status of veteran is important for me. It can also be kind of sensitive."*
Based on experience	*P3 " I already see the alarms such as race, gender and age as well."*
Based on personal reflection	*P21 "What is for me important to consider is just thinking where that data comes from, or trying to imagine what could have influenced the initial fairness of the data."*
Based on information collected from other stakeholders or from the literature	*P8 "With the help of someone having domain knowledge because even though it could be that an expert has some unknown bias thinking "oh, we should probably look into this group", it is also domain knowledge."*
Identification of proxies based on intuition	*P16 " Pregnant status would be very sensitive because it's related to the sex".*
Identification of proxies based on statistical tests	*P28 "I will check what is the correlation of each variable to each other. Basically, having a correlation matrix and checking if there is a higher local relation to those that we have protected.*
Identification of proxies: ambiguous correlation threshold definition	*P28 "Marital status. It's quite a big negative correlation. Age, there's a decent correlation. I would consider something as positive or negative correlated when it's magnitude is higher than 0.25. That's a value that I take from personal experience with my own research."*
Handling of sensitive features	
Dropping attributes: because they are forbidden/sensitive	*P7 "We had to remove the sensitive features in the training set, and then feed the training set into the modeling and model training."*
Dropping attributes: to train "unbiased" model	*P3 "I also make sure that if even I decide to drop these sensitive features, there is no more of this information ingrained somewhere in the data.*
Dropping attributes: not appropriate due to proxies	*P17 "You could argue you get rid of race and sex and just make your models blind to this sort of stuff. But it might not be truly blind because you can have like satellite features. Or like indirectly related features."*
Dropping attributes: not appropriate when they are informative of the target label	*P16 "I see the correlation between these attributes and target columns. I expect to see some correlation between some of them. We could keep it as it is, and we will understand the importance of different features later."*
Dropping attributes: not appropriate in order to monitor algorithmic fairness	*P10 "These are my sensitive attributes. it's important to leave those in. I keep it just to check if it has a weird distribution."*
Handling undesired data distribution	
Grouping the values that are too underrepresented into a larger group (P2, P8, P28)	*P8 "other groups, for instance, these bottom four are really low in number, so in order to get some insightful results, you might want to group them."*
Leaving out under-represented populations (P2, P6, P15, P21, P25)	*P15 "if I have to make a model out of this, then you have to account that the dataset itself has very few points for this category. I would leave out some percentage of data set which is not representational in a way."*
Dropping the attributes which display problematic distributions	*P23 "For example for some variables, if it's very biased, you should avoid using those."*
Transforming set of samples: Collecting additional data, artificially augmenting data, undersampling (P20, P25)	Naturally, all practitioners discussed the possibility to collect more samples, and some mentioned avoiding undersampling not to lose information.
Strategy depends on amount of data	*P2 "If there's only 3 Asians in the whole dataset, it wouldn't make sense to make up for that: it is not enough data to equalise over this. So I would only equalise over Caucasian and African American. Or maybe even combine others as the minority group and have Caucasian as the majority group."*

Table 6.9: Conceptions and practices around algorithmic fairness metrics.

Conception	Example
Used notions	
Group accuracy (e.g., equalized odds)	*P28 "I would compare accuracy for the races "0" and "1", and see whether the results are similar."*
Group output distributions	*P22 "I look for statistical parity and disparate impact because those are not dependent on the target."*
Individual fairness	*P21 "We can have fairness between groups, not necessarily meaning that similar individuals will get the same outcome."*
Reasoning for selecting metrics	
All available metrics (P2, P9, P10, P11, P14, P16, P18, P19, P26, P27, P28)	*P2 "because this model will work in hospital with patients where fairness is important, we check all the group fairness metrics of FairLearn."*
Metrics applicable for both data and outputs (output distribution based)	*P13 "I chose disparate impact ratio because it is a metric that can be applied before and after the training of a model."*
Prioritizing group accuracy or group output distribution metrics based on data correctness	*P15 "demographic fairness is very important. But sometimes, you pick a very obscure data set, then demographic fairness is not the answer if your dataset or representation is fundamentally not correct."*
Prioritizing group accuracy or output distribution metrics based on existence of causal relations between sensitive and target attributes	*P6 "Demographic parity wouldn't be used because it's possible that because of many factors, Caucasian people should be discharged at a higher or lower rate than African American, and so we don't want those to be set to be equal. We want the error rates to be roughly the same, not the selection rates."*
Prioritizing group accuracy or group output distribution metrics based on use-case type (e.g., distribution of resources, hiring) (8 participants)	*P1 "It's important that the model is accurate if resources are being distributed, like whether you actually receive care. So it really depends. In some cases, you really care about whether the model is accurate. In some cases you care more about whether the same proportion of people get a particular resource."*
Prioritizing specific group accuracy metrics based on the weighing of different errors (9 participants)	*P6 "False negatives and false positives are both damaging. I'd have to really think of the costs of those two sides, that informs what fairness criteria you would choose."*
Involving external information (experts or laws) (P1, P4, P6, P8, P12, P19, P22, P28, P29)	*P8 "Depending on domain knowledge, you want to know what metric you want to look at. Just by myself, I wouldn't really have an idea what would be in this case the best metric. A doctor would know. This is either some legal stuff or just some ethical stuff that we want to make sure that's OK. "*
Using their own intuition	*P11 "I know there are a million different metrics. I would compute statistical parity for sure. And then I would probably go down the list."*
Mentioned limitations of the metrics	
No limitation envisioned	*P19 "I think for fairness these metrics work well."*
Limitations of certain metrics said to be fulfilled by others (P8, P10, P21, P24)	When asked whether one metric such as demographic parity is enough, they answer no but instead they can use another metric like equalised odds.
Limited to reflect underlying injustice (P1, P2, P3, P9, P18)	*P9 "In the college admission example, due to historical factors, we see correlations between certain races, socioeconomic classes, and education. Should people of different races be given equivalent outcomes? I don't think so. You have to consider and fix the underlying factors first. You can't just fix it at this top level and expect it to be done. So I can't call demographic parity enough."*
Limited to reflect certain notions of fairness	*P6 "If we look at the broad range of people, people have views on fairness that are defined on very different criteria than the ones we can see in these numbers."*
Limited to account for the impact on other stakeholders	*P19 "it depends on the situation, but mostly it's not only me who could be affected, but people around me can also be indirectly affected by whatever it is. In the case of health, if I was to be discharged without being supposed to, I would be directly affected, but also my family or people that I'm surrounded by."*
Limited to account for individual outcomes (impact of outputs on each individual)	*P18 "If I don't get a credit score, it's no problem because I'm young, I have a lot of opportunities ahead for myself, but then if I were 50 and I have 4 kids and I know I'm gonna be homeless, then maybe it's worthwhile giving me the credit."*
Limited to account for exploitation of outputs by decision-makers	*P3 "it reminds me of this famous child benefit scandal, when the problem was not a model, but the people who were using these predictions. They were literally doing this manual post processing of predictions according to their beliefs."*
Dangers of fairness metrics to be used as checkboxes (P3, P6, P9, P13, P29)	*P6 "It's easy to think: we checked the fairness box because we implemented this specific library, or this constraint when really fairness is a much broader topic."*
Dangers of fairness metrics to remove critical attitude (P3, P6, P9, P13, P29)	*P13 "Responsible AI is an AI built with high quality processes, not only regarding fairness, but regarding using the best metrics, not doing something like "My metric is good, so my model is good". No. Have a critical point of view."*

Table 6.10: Conceptions around algorithmic fairness mitigation (in grey explicit trade-offs).

Conception	Example
Used methods	
Data balancing, attribute dropping	
Scoping out populations (P2, P9, P15, P25)	*P9 "You can choose to limit the scope of your classifier and use this one on people who are over the age of 60. That's one way of making sure that you're not having false positives or false negatives on these underrepresented data."*
Modeling a new task (P4, P6, P15, P17, P28)	*P6 "we have enough data to train separate models. So you might not even use the normal FairLearn strategy –training one model that works well across populations."*
Data preprocessing method	*P22 "we would use some of this re-weighing or adversarial debiasing kind of techniques."* (reweighing 10 participants, correlation remover 5 participants)
In-processing method (12 participants)	*P2 "After [computing fairness metrics], I would do some in-processing mitigation."* (e.g., grid search and Lagrangian classifier)
Post-processing method (P1, P2, P3, P12, P21, P29)	*P3 "You have threshold optimizer. So for example, for logistic regression, the decision threshold by default is 0.5, and you also can play a little bit with the threshold that defines whether this data point belongs to this class or to that class."*
Reduction method	*P6 "what we've done internally, it is doing this reductions approach in FairLearn."*
Selection	
Based on speed	*P6 "the major downside to the reduction approach is that it can take a long time."*
Based on amount of available data	*P6 "we actually have enough data that we might be able to train separate models."*
Based on applicability to specific model	*P12 "the cons are that they are not model agnostic: it depends on each kind of model you apply. You'll need to know all of them where they can be applied."*
Based on compatibility with deployment constraints	*P12 "When you are in production, in some cases, you won't be able to do a lot of changes. So post processing is good, you're just changing the labels and given a minimal loss of accuracy, you may just make it fair."*
Based on image it brings to the company	*P13 "[talking about post-processing methods that flip certain model outputs] They kind of imply a bias in the process. It would be a problem for the company to say that they are doing this: if I am a company and I am saying publicly that I am imputing bias on my model, how would society react to it?"*
By experimenting	*P21 "try out a few of those algorithms which are still applicable."*
Preference for not simulating new data	*P22 "if possible, we want to re-sample the data instead of simulating data. I typically prefer if they can get the data from the source corrected, as much as we can."*
Preference for changing the data (P9, P15, P16, P19, P20, P24)	*P9 "if you can get fair data, that is the best way to make sure that your classifier is going to be accurate on all representations of people. More data has always been the best way to make a machine learning model more accurate."*
Admitting not knowing	*P11 "I would just like read up on it so that I know about this strategy is better."*
Mentioned limitations of the mitigation methods	
Non-applicability to certain types of tasks / algorithms	*P7 "we needed to somehow mix up some approaches in order to customize them and modify them. In some cases, there is absolutely no methodologies to tackle individual fairness mitigation, that can be applied on the loan adjudication use case."*
Impact of one method on different fairness metrics	*P21 "Optimizing for one type of fairness will make another type of fairness worse. If I optimize for fairness between individuals, fairness between groups will suffer."*
Does not fix structural causes of injustice	*P2 "I think about demographic parity, about making the decisions equal for everyone in population. It depends a lot on the way you do this, because you can also positively discriminate to get these outcomes, and it differs by use case if this would be fair. Or you can get a population fair by making the model work less good for the majority group and then it would be demographic parity. I wouldn't consider that fair."*
Approach might not be ethical	*P1 "One thing that people very commonly do is use different decision thresholds for different groups, and that's a very easy way to get different selection rates, but what does it imply in practice? You literally put people to a different standard. Whether that's justifiable or not, it depends on the scenario."*
Inadapted solution to the cause of the unfairness	*P29 "When they were trying to test out a model to allocate poverty benefits to low income individuals, especially for food banks, Hispanic applicants were being rejected at a higher rate, and that's just because these applicants actually aren't fluent in English. They're having trouble with the application form, and so the solution to make this system more fair: just offer the form in Spanish."*
Biases users to take technical mitigation approaches when they might need to be structural	*P29 "If you find some disparity, what does that mean in the real world? Then what is the intervention you take? If you don't understand the harm, you can't take an intervention to stop the harm. That part is very important because there are plenty of cases where there's an intervention that isn't technical.' '*

6.7.1. ON GOALS AND ENVISIONED FACTORS IN TENSION WITH HARMS

Table 6.11: Goals formulated by practitioners along the interview sessions.

Type	Example
Modes of handling harms and potential impossibilities	
Not deploying the system (P1, P29, P17)	*P1 "if you really need the mitigation approaches for the model to be accurate or have a good selection rate, you should always question whether machine learning makes sense to use in this scenario."*
Transparency for the decision makers to make the informed choice to deploy	*P6 "That would be a conversation I would have with the hospital. I could say where we're confident, and where we're not confident."*
Transparency for the decision makers to account for it in deployment	*P20 "I would certainly voice my concerns towards the Fairness of a problem and how people plan to solve it"*
Not accounting for the specific issue	*P6 "There's a question of what is the current performance. We're comfortable deploying something if it improves the baseline performance, maybe it's OK if the data is not perfect."*
Mitigating this issue instead of prioritizing another objective	*P17 "I think they could automate it. But it's just those other concerns that I've addressed. You need to understand how it's affecting people and what you could do if you were getting really poor performance on one of our smaller subsets."*
Examples of rationales for prioritization of harms and other objectives	
Making the least-bad choice around impossibility (with intuition or external inputs)	*P30 " if I decide to optimise for demographic parity or equalised odds, it's impossible to optimise for everything, so I need to pick up specific metric that I'm going to look." P21 "This boils down to being able to make a rational, reasonable choice of what are we actually trying to optimize at the early stages? And then you know, keeping in mind that making some sort of fairness metric better, even a lot better, it can still negatively influence other metrics."*
Compromising on certain aspects hoping to solve other issues	P2 mention that an attribute is sensitive when it should not be used for decision making, but considers that one can train a model with it as long as the model does not learn to rely too extensively on it. Some practitioners recognize that one cannot aim for equal data distributions across groups and that a middle ground is acceptable.
Neglecting the issue to focus on other objectives such as model performance	*P18 "This would not really be of my concern as in having to include, for sex, I don't know, 20 categorical options. Because I feel like at the end of the day, we're not doing politics here, but we're trying to solve a problem. But if the results that we obtain are really poor because of the fact that we did not take into account these attributes or variables, then we should include them. "*
Not accounting for (impossible?) limitations of fairness metrics because they are better than nothing	*P8 "if you don't depend on metrics then how are you going to evaluate your model? You need to have at least some metrics to be able to say a) my model is fine, and b) my model doesn't have any harmful applications."*
Judging when the metrics values are satisfying	
Ambiguous	*P2 "the difference between African American and Caucasian, their balanced accuracy is pretty equal. False negative rate is also pretty good. So, I think this model is for them equal. So I would not be worried about these numbers."*
Value higher than (human) baseline	*P6 "We're comfortable deploying if it improves the baseline performance."*
When one has tried mitigating as much as possible	*P20 "I strongly believe that there is no way we could achieve absolute fairness because we are biased by nature. You should try your best, and you stop when you run out of ideas and after you've done your best."*
Acceptability for the data subjects	*P29 "Absolute fairness is not possible to achieve. It could be: yes, there is some disparity, but the impacted communities sort of feel fine about that."*
Acceptability for the model requesters	*P19 "I don't think it's possible to remove the entire unfairness. But that depends on the people that they're making the model for, and how they react to it."*
Acceptability for experts	*P6 "There's a question of what is an acceptable difference in performance and I think it's a difficult question to answer, and that's something you want to talk to all the stakeholders about."*

Table 6.12: Other factors that might impact harms (in grey the ones that are accurately envisioned).

Type	Example
Requirements on model objectives	
Accuracy, type of output, inference time // impact algorithm	*P15 " do I want the probability of hospital readmissions? —I would guess that is what I want then probability-based classifiers are good."*
Model explainability for decision-maker	*P8 "For the algorithm, like in such a hospital case, you would prefer a non black box algorithm so you can have a look at: how does every feature influence my results?"*
Rare consideration of model explainability for data subject	*P27 "You should not base the output only on the model. It should also be an expert, so that's not a black box who tells the person "you're not getting a loan" and that person would be really confused of why."*
Necessity to trade-off these requirements	*P2 "I would first check different classification models and which one has the highest AUC value. If there is a more explainable model that just lacks a bit of accuracy or AUC, then I would choose that one over the bigger models that are not explainable. "*
Typically no requirement on algorithmic fairness and other harms	*P7 "We had a company involved in paper recycling. In that case, we definitely need to make sure that the amount of data that we are requesting or any other request that we have from the client wouldn't have any side effect on the environment."*
Requirements on system infrastructure	
Deployment requirements such as easiness of deployment, easiness of update, *and easiness of monitoring*, and running time	*P29 "do you want it to be a simple model so that you could retrain it properly? Do you want something that's very small, so you can deploy it on like a AWS or on Azure" P3 "The simpler is the model, the easier it will be to deploy, the easier it will be to monitor, and the easier will be to retrain"*
Computational power and cost for deployment	Impact algorithmic choice, dataset size, and trade-off with model accuracy *P29 "Do you want something that's very small, so you can deploy it on like a AWS or on Azure?"*
Computational power in relation to environmental impact (only 2 practitioners)	*P15 "We have 20,000 GPUs and it gives a very high accuracy like human level. On the flip side, you have this much power and then how do you obtain this same accuracy within any alternative algorithm with much less compute power?"*
Requirements on the development processes	
Time pressure	*P22 "Everybody has deadlines and this is going to add to the work."*
Data constraints	
Availability of data, feasibility of collecting data samples & attributes // impact dataset, algorithm, model performance	*P5 "One of the first things I would do is to see whether this dataset is sufficient for running a model. Sufficiency comes from 2 perspectives. One is what kind of model I want to use. If the dataset is not large enough, I cannot use a neural network, I would end up using a linear model which would basically have its own limitations.*
Data types impact choice of algorithm	*P25 "There are algorithms which take both (continuous and categorical). You can input the range value and feed categorical data and the model will work."*
Features for higher accuracy/fairness models // feasibility and practicality constraints	*P6 "Right now, we have 100,000 records. If we decide that we want another feature, we have to wait a long time before we get all the data on that feature again. So we always try our best and see if it's good enough."*
Trading-off the appropriateness of the target label with the above data constraints	*P1 "In ML, people choose a target label based on what's easy to get rather than when you think about more statistical inference, then it's typically much more well thought out. Many of the issues with fairness can come from mismeasurement."*
Inherent statistical and theoretically clashing impossibility around algorithmic fairness and absence of harms	
Inherent statistical impossibility in reaching algorithmic fairness if considering all sensitive proxies	*P21 "Fairness becomes almost impossible, because it could well be that Medicare and Medicaid are a proxy for demographic features: whether minorities are, for example more likely to take Medicare and Medicaid."*
Inherent statistical impossibility in reaching fairness because all attributes are possibly sensitive	*P17 "The only one that society has said it's OK to be biased on is smoking because it is probably the one that you have conscious decision about although you could argue that depending on where you're born, it is probably different probabilities."*
Inherent statistical impossibility in reaching algorithmic fairness simultaneously for multiple metrics	*P21 "optimizing for one type of fairness will suddenly make another type of fairness worse. if I optimize for fairness between individuals, it's possible that the fairness between groups will suffer, but also even one level lower, if I optimize for predictive parity, it's possible that the disparate impact will suffer."*
Theoretically clashing objectives around algorithmic fairness and absence of harms (e.g., privacy around data attributes and their encoding, fairness, and accuracy)	Impossibility in reaching or measuring algorithmic fairness without accessing sensitive attributes traded off with the law forbidding to exploit these attributes *P9 "Is the dataset collected in a way that had the informed consent of people in the data set? Or are we collecting hospital records and using that data to do something that patients were not made aware of? You're under health care data constraints like HIPAA."*
Theoretically clashing objectives around the use of ML and the absence of harms	Employing ML itself might be the subject of trade-off, as it might be useful for various stakeholders to deploy an ML model, but this model would require privacy-infringing data (P19), or might negatively impact the environment (P28).

7

FACTORS IMPACTING PRACTICES TOWARDS ROBUSTNESS & HARMS

7.1. INTRODUCTION

Following on the previous chapter (Chapter 6) where we identified a fragmentation of considerations and practices around algorithmic harms from machine learning (ML) developers, we now investigate the causes of this fragmentation. Particularly, considering that fairness toolkits are becoming a defacto standard means of tackling questions pertaining to algorithmic fairness[1] and potentially of teaching "ethical ML" to developers [117, 536], it is important to understand the extent to which developers rely on such toolkits, and whether and how toolkits shape their practices. Addressing this knowledge gap is a crucial step towards questioning the broad impact of fairness toolkits. A majority of past studies [369, 846, 220, 679, 481, 665, 522, 619] that have focused on the practices and challenges of developers in using the fairness toolkits have already identified a number of limitations of the toolkits in terms of design and technical specifications, that might hinder their adoption. However, such studies fall short in two major ways.

Fairness toolkits allow to implement algorithmic methods for handling algorithmic unfairness. Yet, it is now well understood that these methods bear conceptual limitations [62, 557, 508, 718, 361, 457, 870]. Algorithmic unfairness is only a simplified representation of distributive unfairness (what the metrics aim at quantifying), mitigation methods might themselves cause harm or not address the root causes of distributive unfairness, and other harms (beyond distributive unfairness) caused or reinforced by the use of ML systems are not accounted for by this framework (e.g., the purpose of the system itself might considered harmful, independently of the system's outputs being fair or not)[2]. None of the studies around practices and toolkits has however investigated how ML developers might conceive and overcome these limitations. It is especially unclear

[1] https://www.borealisai.com/research-blogs/industry-analysis-ai-fairness-toolkits-landscape/; https://www2.deloitte.com/de/de/pages/risk/solutions/ai-fairness-with-model-guardian.html

[2] In the remaining of the paper, we use *algorithmic harms* to refer to any harm that ML systems might cause

whether the toolkits narrow down developers' activities towards algorithmic unfairness and broader harms. These insights are necessary to envision where to focus future research efforts in terms of algorithmic harms beyond algorithmic fairness.

Besides, prior studies do not report on differences of practices and challenges across developers, and the factors that cause these differences. Yet, identifying these differences, and grounding these differences into the *factors* that impact the fragmentation would allow to identify the root causes of potential flawed practices and of certain challenges. This would allow to envision more appropriate future solutions. In other words, explicitly looking into factors would allow to answer the following questions: should fairness toolkits be our object of study to foster practices for handling algorithmic harms, i.e. are toolkits really the most important factor that supports and impacts practices around algorithmic harms (they would be if we would find a coherent set of practices across developers using a toolkit in comparison to those who do not)? Or are they only technical mediators of practices, that are impacted by deeper factors beyond the availability and design of the tool?

Hence, in this study, we ask: *what are the main underlying factors that impact the attitudes and practices of machine learning developers, and that might represent challenges leading to the persistence of harms?*. More specifically, we divide the question into two sub-questions: 1) How effective are toolkits in enabling developers to reflect about algorithmic harms and to handle them? 2) Which are the factors that affect the (in)effectiveness of toolkits in shaping developers' practices around algorithmic harms?

In order to answer these questions, we conduct 30 semi-structured interviews[3] with developers of various backgrounds. We compare practices before and after a developer is introduced to a fairness toolkit (within-subject experiment), and practices between developers who do not use a fairness toolkit to those who do (between-subject experiment), in order to understand the potential role of toolkits in shaping up practices. Besides, we further analyse qualitatively the interviews, and compare practices across developers, and across the two toolkits selected for this study, in order to identify potential additional factors that might impact practices.

We find that toolkits do increase awareness and use of algorithmic methods towards algorithmic fairness, do not impact considerations of algorithmic harms, yet can foster a checkbox culture with absence of reflexivity around the limitations of algorithmic fairness. More than solely toolkits, we also find that various human factors, such as types of training, and psychological and socio-demographic traits, as well as contextual factors, and especially organisational incentives, interact to shape up how developers make use of the toolkit, how reflexive they are around the limitations, and whether they conceive and tackle broader algorithmic harms. These factors, while they have been mentioned scatteredly across research publications that deal with perceptions of algorithmic harms [405] or the governance models of organizations around algorithmic fairness [665], had not been analyzed in detail in terms of their impact on the practices for the development

or reinforce, among which are *distributive unfairness* harms (related to the unfair ways in which resources are allocated following the recommendations made by the outputs of an ML system). We use *algorithmic unfairness* to refer to the limited conceptualisation of distributive unfairness in the lens of algorithmic metrics and methods developed by the scientific community.

[3] All our materials, resulting data, code and analysis will be shared publicly. https://osf.io/dmr82/?view_only=a00e68796f494fbb9776cf9a95fb7051

of ML systems (with harms in mind). We then further discuss the implications that our findings bear when fostering reflexivity among developers towards avoiding algorithmic harms, e.g., in the form of design guidelines for fairness toolkits, as well as educational programs, and for further enforcing policy efforts towards making algorithmic systems less harmful.

7.2. RELATED WORK

7.2.1. FAIRNESS TOOLKITS FOR DEALING WITH ALGORITHMIC UNFAIRNESS

ALGORITHMIC UNFAIRNESS

Each step of the machine learning (ML) lifecycle might create or reinforce *distributive unfairness* [546, 811]. Theoretical works have primarily developed *algorithmic fairness* metrics [848] that aim at measuring distributive unfairness in the outputs of the final model or in a dataset. These works also propose algorithmic unfairness mitigation methods [268, 66] that ought to improve the model's algorithmic fairness as defined by the metrics. Facing the diversity of metrics, the challenge for a developer is to choose the appropriate one for their task.

Several studies have investigated how ML developers work with algorithmic fairness metrics and mitigation methods. Topics of focus revolve around general challenges met by developers [369, 846, 665, 522, 564, 719, 619, 878, 612], and obstacles and limitations for the application of algorithmic fairness methods. Findings outline the need to support developers to concretely use fairness methods, as this use is challenging due to the context dependence of methods, the current lack of guidance [369, 522], and the need for adapting methods that are incompatible with targeted tasks [369].

EFFECTIVENESS OF FAIRNESS TOOLKITS

To facilitate the adoption of algorithmic fairness metrics and mitigation methods, various companies and public institutions have built fairness toolkits. These toolkits are typically code repositories that allow an easier implementation of the metrics and methods. Examples of these toolkits are FairLearn [107], AIF360 [91], Aequitas [704], Themis-ML [77], ML-Fairness Gym [205], TensorFlow Fairness Indicators [892], etc.

Various works [220, 679, 481] have shown through interviews the beneficial use of toolkits by developers for developing fair models and learning about algorithmic fairness. Yet, they also show their limitations in terms of support provided to developers for designing the right algorithmic fairness evaluation, noting that participants often inappropriately change their modeling task definition to fit existing tools. These works also identify obstacles to the application of the toolkits in terms of compatibility with other ML frameworks and usability, summarized into toolkit checklists that should inform the design of future toolkits. We will show that our results corroborate and complement these insights. Indeed, to the best of our knowledge, our work is the first to investigate (or report) whether the toolkits do impact practices contrary to a situation where no toolkit would be available, whether there are differences in practices of different developers using a same toolkit, or whether different toolkits lead to different practices.

7.2.2. FAIRNESS TOOLKITS FOR REFLECTING ON HARMS BEYOND ALGORITHMIC UNFAIRNESS

ALGORITHMIC HARMS

A few theoretical works have looked beyond algorithmic fairness to identify other harms of ML [62, 557]. We now present a few of these harms that are highly worthy of consideration according to the literature. Algorithmic fairness metrics and methods bear conceptual limitations, that do not allow to comprehensively gauge the distributive unfairness they are aimed at addressing. By limiting harms to the frame of output distributions (distributive justice), algorithmic fairness cannot reflect the contextual factors that influence what is considered fair. For instance, it assumes that parity is always desired in the model outputs [508], it does not account for the impact one same output has on different receivers of this output [557], nor for the indirect impact on non-data subjects [457]. Looking at the process to reach algorithmic fairness (procedural justice), the metrics and mitigation methods do not make sure that the way in which the unfair situation is addressed is aligned with moral principles [870]. For instance, individuals or groups might see low disparate accuracy by all receiving unjustified treatment [571], or by all being treated differently (e.g., post-processing methods allocate different decision thresholds for different groups) which consists in direct discrimination [308].

Three other categories of harms have also been discussed. First, ML requires to use *datasets* whose schemas and sampling can be harmful. For instance, certain attributes and their values might be offensive [901, 110] or inappropriate [546], e.g., use of non-volitional or privacy-infringing attributes [315, 834]. Second, research questions the *desirability of the ML model* in the first place, its use for undesired applications [558, 376, 432, 557], and how it impacts structures in place [257]. Using ML for certain tasks might be questioned, for instance because it means making decisions for people by comparing them to others instead of following the principle of individual justice [105, 254], or because it reproduces historical, potentially harmful, data patterns [653]. Third, certain researchers question the *negative externalities caused by the production process* of ML applications, such as the environmental impact of data centers and model training [132, 93], the poor labor conditions of crowd workers [689], the privacy-infringing training data [660], etc.

EFFECTIVENESS OF FAIRNESS TOOLKITS

Besides investigating the effectiveness of toolkits in enabling reflexivity around algorithmic unfairness, it is important to acknowledge the known limitations of the algorithmic fairness methods and the existence of other algorithmic harms that ML systems might pose. To the best of our knowledge, no work has investigated practices in relation to these limitations. We do not know to what extent the use of fairness toolkits —that foster the use of the algorithmic fairness methods— impacts considerations of algorithmic harms and of the limitations of algorithmic fairness (that are typically obfuscated from the toolkits). It is unclear whether fairness toolkits, that do not deal with these harms, might lead developers to “forget” them.

7.2.3. FACTORS AFFECTING THE USAGE OF TOOLKITS

The effectiveness of fairness toolkits in enabling reflexive practices among ML developers around algorithmic unfairness and harms is conditioned by factors that shape the usage of these toolkits. Research into the characterization of these factors is still scarce. It is important to understand which factors make developers choose one metric or the other, and more broadly, to identify the factors that impact the decision of developers to try quantify unfairness, and later to mitigate it. The factors that lead a developer to handle broader algorithmic harms have also not been investigated in the past. Knowledge of these factors could allow to better understand the deeper nature of the challenges faced by developers, and to provide more personalised support to these developers.

Up to now, studies have solely identified organisational factors, that are further shown to be obstacles for developers to develop fair models [665, 522, 523, 846]. Contrary to our work, previous studies had not accounted for human factors in their study design or in their result analysis, such as Deng et al. [220] who only reported on coarser-grain practices (e.g., they reported that the developers they interviewed recognize the limitations of their knowledge and wish to receive help from domain experts, but do not specify any difference across these developers). In our study, we find such factors, and also investigate the existence of technical ones.

7.3. METHODOLOGY

To characterize the effectiveness of fairness toolkits in enabling reflexive practices, and to identify the factors that might impact and fragment those practices, we adopted an empirical and qualitative approach via 30 semi-structured interviews with ML developers. By comparing practices within-subjects (participants are observed before and after receiving an introduction to fairness toolkits), we observe the extent to which toolkits enable or hinder reflexivity. Additionally, by comparing practices in-between subjects who bear different characteristics (e.g., background and prior experiences) and who use different toolkits, we characterize the fragmentation and delve further into the contributing factors.

7.3.1. PARTICIPANTS

We recruited our participants in the period of April-June 2022, by means of personal networks, targeted requests on social media, calls for participation on the official Discord or Slack communication channels of the toolkits, LinkedIn, and snowball sampling. The participants received no financial compensation, and their contributions were voluntary. Our institution's ethics committee approved the study. All participants signed an informed consent form acknowledging the risks involved with participating, as well as agreeing to the interview being recorded (all interviews were conducted online), transcribed, anonymized, destroyed, and consented to the results being used in scientific publications.

A total of 30 participants were recruited across research and industry institutions, and across application domains such as healthcare, finance, and predictive maintenance (cf. supplementary material). Manual sampling was performed to make sure that all participants have responsibilities in ML model development, deployment, or evalua-

tion; varying levels of prior experience with ML, ranging from 2 to 15 years; and varying practical experience with algorithmic fairness and fairness toolkits (11 participants already had experience with FairLearn, and 9 with AIF360). The resulting participants differ in terms of demographic background (nationality, gender, and age), level of highest education, educational background, and type of training received around ML. Besides, participants already experienced with algorithmic fairness presented variations in terms of how they learned about the topic, the kind of experience they have had, and for how long they have worked with these issues (from 0 to 18 years).

7.3.2. INTERVIEW PROCEDURE

The interviews with participants already familiar with a toolkit lasted one hour each, going through Task T1. The interviews with the other participants lasted around two hours each, through three stages (Task T1, a tutorial about one fairness toolkit, and Task T2). These three stages were designed to identify how the use of toolkits might impact practices around algorithmic harms. Comparing practices between participant groups with or without prior familiarity with the toolkits allowed us to unveil other influential factors, such as the type of training received around harms. In total, we collected 2207 minutes of recording. In Figure 7.1, we show the workflow of the interviews with the questions asked in each stage, for the two kinds of participants. We asked three types of questions: background experience questions (demographics, experience with ML and algorithmic fairness); reflection questions around algorithmic fairness, harms, or toolkits, and around general comments, wishes, doubts, and challenges the participants might have about their workflow or harms; and process questions to understand the reasoning behind each participant's activities during the tasks (cf. supplementary material for details on tutorial and questions).

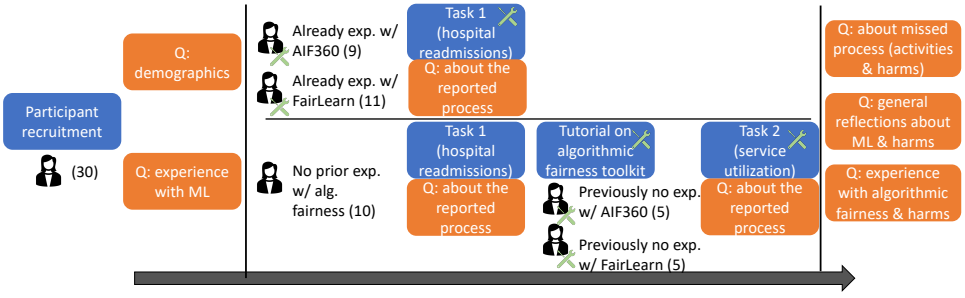


Figure 7.1: Interview procedure for the participants already experienced with a fairness toolkit, and for the participants who did not have any prior practical experience with algorithmic fairness. In blue: **the main steps of the procedure**; in orange: the **questions posed in each step**.

7.3.3. MATERIALS

Use-Cases. We chose two use-cases, the first one involving the prediction of *hospital readmissions* within 30 days for individual patients [798], referred to as Task T1, and the other involving the prediction of low or high *medical services utilization* [342], referred

to as Task T2. We pre-processed the two corresponding datasets for them to have similar characteristics (number of attributes and of records), and to be prone to similar harms (cf. supplementary material). By employing comparable domains and datasets without re-using the exact same use-case for the two tasks of the interviews, we aimed to minimize learning effects. We chose the domain of healthcare because it is prone to various harms, requires expertise to be handled correctly (i.e., we could check whether the participants mentioned the limits of their knowledge [220]), several corresponding datasets were available, and these are not the most frequent use-cases in the algorithmic fairness literature which allows us to minimize the confounding effect of familiarity with the domain of application. Our choice also allows to mimic a realistic situation, where oftentimes, developers have to develop or deploy models without having extensive expertise in the domain of application. In such cases, developers' decisions might lead to harms, that fairness toolkits are meant to empower developers to reflect about.

Tasks, Toolkits, and Notebooks. For each task, we shared a Google Colab notebook with the participants, which included a design brief with one of the two datasets preloaded. The design brief mentioned that a hospital (or an insurance company) wanted to optimize their cost and services (or their prices), and therefore wanted to investigate whether ML could help them predict readmissions (or utilization, respectively). The institution tasked the participant to investigate this feasibility possibly using the dataset they had collected, and to report on their findings by speaking outloud. Along the investigation, when participants mentioned some code-based exploration, we shared corresponding code snippets prepared before the interviews to speed up the process.

For the interviews with developers who had used a fairness toolkit in the past or with the ones we introduced to a toolkit, we loaded a specific toolkit (FairLearn [107], or IBM AIF360 [91]) into the notebook, that they were most familiar with. We consider these toolkits because they contain a large number of functionalities around algorithmic fairness; they are the most studied toolkits in research [481, 220] and appear to be popular among developers. Cf. Appendix for details about our interview materials.

Analysis of the Transcripts. We analysed the transcripts using a combination of inductive and deductive coding. The first author identified the segments discussing the main themes we wished to discuss (e.g., the harms, their conceptions, identification, and handling, and toolkit use), and coded any other emerging themes (e.g., other factors that developers trade-off when developing ML models) in collaboration with four other researchers. Then, the author in discussion with the other authors, reconciled redundant codes. Finally, this first author studied each of these codes based on their associated participants. While we cannot certainly identify which factors cause observed variations in terms of conceptions and practices based on our qualitative study, certain developers explicitly mentioned potential factors that we report. We also explore quantitative differences based on the background information we have about the developers (yet, all the factors are impacting practices in different ways, that we cannot explore within our study).

7.4. RESULTS

7.4.1. ON THE EFFECTIVENESS OF TOOLKITS

In terms of algorithmic unfairness, developers reported the toolkits to be extremely useful for them to quantify and mitigate unfairness, what was confirmed by our observations. Yet, we also identify drawbacks of the toolkits for distributive unfairness, that we describe next. In terms of algorithmic harms beyond distributive unfairness, we did not note any evidence of positive or negative impact of the toolkits on developers' considerations and practices.

EFFECTIVENESS OF TOOLKITS

Among toolkit-inexperienced developers, toolkits fostered a positive shift in practices around algorithmic fairness between task T1 and their introduction in task T2. Before being introduced to the toolkits (T1), it was not natural for the developers to reflect about algorithmic fairness. After our tutorial (T2), they began discussing potential unfairness caused by the outputs of their models and trade-offs between different fairness metrics and with accuracy, to judge which model is satisfactory (even if superficially on occasion). They also started envisioning approaches to mitigate the potential issues with the outputs. Hence, toolkits, for these developers, represent a means to foster awareness around distributive unfairness and its causes. *P19: "Just seeing how it worked, made me realize that it's not only about the dataset, but there's bias everywhere."* It also represents a means to learn about existing solutions to mitigate unfairness, and a prompt to start actively tackling the issue (being readily-available code repositories, toolkits lower the entry-barrier to the problem). *P17: "If it's quick and easy, run a quick check. 'Oh, there is something there I didn't think of. I need to explore that.' I could see that happening."*

As for toolkit-experienced developers, they primarily use toolkits to speed-up their processes around algorithmic fairness, and to foster communication with other stakeholders. *P11: "I talk to business people and this is how they can connect to this topic from the technical side because they can't code or anything."*

UNDESIRABLE CONSEQUENCES OF TOOLKITS: REDUCING HARMS TO ALGORITHMIC FAIRNESS

Despite their perceived utility, toolkits can be misleading, and create a gateway to a narrow view on distributive justice. 6 out of 10 participants who were inexperienced with fairness, 4 out of 9 relatively experienced ones, and 2 out of 11 experienced ones took the toolkits at face value. They applied all fairness metrics available through the toolkits without considering their meaning and appropriateness, declared a model satisfying if certain values of (often arbitrarily picked) fairness metrics were reached (sometimes operating a non-informed balance between accuracy and fairness metrics) without reflecting on their limitations. *P13: "With the use of toolkit, I don't think my view changed. [Before having the toolkit,] I already believed in what the techniques could do. So if the toolkit correctly implements techniques, I have faith in it."*

55% of developers who were more experienced with fairness explicitly expressed concerns surrounding the toolkits. Toolkits might narrow down critical thinking around what is measured in relation to distributive fairness and be misleading, limit reflections

on broader socio-technical concepts, and foster techno-solutionism triggered by the development of unfairness mitigation methods. *P22: "You cannot rely on the toolkit. You need to understand the problem and the domain knowledge. I can easily see these toolkits like before metrics like precision, recall were just thrown at random without knowing the actual meaning. Things like statistical parity difference, as they become more common, I can see them being misused because a lot of people don't even know their definitions. It's easy for people to misinterpret them."* developers also felt that toolkits encode biases in their setup. *P23: "These libraries can introduce some biases that you are not aware of, so you don't need to put all the chances on those libraries, you should look into data yourself to see what type of bias data contains."* All in all, toolkits might illegitimately serve as a checkbox. *P3: "Fairness for many companies is just a small checkbox, and sometimes people put their mark without any question. I hope there will be a time when they understand that fairness is not about code and just picking up one toolbox. [..] The toolkits would constrain your view if you're using them blindly."* This is in direct contradiction with the way a few participants perceive the toolkit as an opportunity to realize and convey the complexity of the distributive justice problem *P21: "The recurring theme of our conversation is that fairness is difficult, and this realisation is what the toolkits achieve. They give a large variety of options to make fair models, but their biggest positive impact is helping developers realize that this is not a topic where we just do the same five steps and we have a fair model, but it's something that requires a lot of consideration."* This is evidence that beyond the toolkit itself, there are additional factors that impact practices –we discuss them next.

TECHNICAL FACTORS: DIFFERENCES ACROSS TOOLKITS

We do not find any notable difference in the conceptions of harms between developers who used different toolkits, irrespective of their experience with fairness. While in practice some functionalities (metrics and mitigation methods) are only supported by one of the toolkits, this did not appear to be a major obstacle to the developers, who seemed to use other methods when needed (some developers also mentioned having to design novel methods to tackle their problems). This could however potentially be dangerous for beginner developers who learn about algorithmic fairness solely through the toolkits, and may revert to sub-optimal metrics and methods.

developers did mention factors that impact the adoption of toolkits: compatibility with existing frameworks and code, frequency of maintenance and open source nature, ease of adoption and learning curve, transparent implementation and documentation, amount of functionalities and adaptability to various use-cases, and socio-technical questions the toolkits foster (cf. supplementary material for details about these factors and the others we identify). Interestingly, these mainly refer to non-functional requirements. While developers agree on these requirements, the evaluation of the satisfaction of a requirement for a toolkit was sometimes contradictory across developers when choosing one toolkit over the other (oftentimes, developers did not know both toolkits, but used similar arguments for explaining the choice of one over the other), e.g., they mentioned choosing AIF360 or FairLearn both because of their compatibility with existing coding frameworks.

7.4.2. HUMAN FACTORS

Finding out that the toolkits are not the only factor that substantially fragments practices, we turn to the human factors and the specificities of each developer to understand observed variations.

EXPERIENCE IN ALGORITHMIC HARMS

As already mentioned, the amount of prior experience with algorithmic fairness (which includes experience with fairness toolkits) seem to impact practices on average. Relatively experienced developers typically think of less harms and reflect on issues with less critical attitude, and more often solely relying on their intuition, than the more experienced developers. Most participants who are just entering the realm of distributive fairness through a toolkit are not very critical about algorithmic fairness. *P20: "Using it this way seems to be one of the best ways, taking into account what I knew before, and what I learned today about the toolkit."* They become more critical if they accumulate more practical experience and knowledge by further exploring the toolkits' guidelines. Hence, more than the mere amount of experience, the type of prior experience with algorithmic fairness is a factor that seems to strongly impact practices. For instance, practices among the most experienced developers do vary, with some also relying solely on sometimes flawed intuitions, while others systematically involved external sources of information and rigorous computations (e.g., other stakeholders, laws, guidelines, business) and potentially make use of statistical tests.

WAYS OF LEARNING ABOUT ALGORITHMIC HARMS

Types of Interactions with Others. The developers who displayed a more critical attitude discussed having learned about distributive fairness through interactions with various stakeholders. For instance, half of the participants who have learned about the metrics primarily through the code and 70% of the inexperienced participants who only briefly learned about the metrics during our interview discussed observing all metrics without reflecting on their meaning, while all the ones who have had more interactions with the research community (7 participants) or other interdisciplinary teams (3 participants) judged choices based on use-cases. These interactions (discussions, workshops, and conferences) often involve colleagues, clients, or researchers in AI ethics that highlight potential limitations and critical attitude to keep, or illustrate the subjectivity of the topic. *P3: "We invited one developer of FairLearn to run workshops. Her message was clear: you can ingrain fairness in code, but if you don't understand what you're doing, you will be in the world where we are already."* Similarly to previous results showing that discussions can positively impact fairness considerations [636, 544], the participants we introduced to the toolkits also mentioned the benefits of our discussion (to make them conscious of potential harms and of the limitations of their own, often non-critical practices), more than the one of the toolkits. *P20: "[Do you feel like your perspective on algorithmic harms changed after after seeing the toolkit?] Yes, I mean more after this discussion altogether. I personally wouldn't have taken some of them into account myself if I weren't pointed in the right direction by your questions."* Our participants reflected about the choice of fairness metrics and mitigation methods, once we explicitly prompted them about specific use-cases and actual meaning of different choices. *P28:*

"You also mentioned proxy. And I realized that just protecting some variables doesn't mean that you have removed completely that bias."

Types of Courses. Other developers learn about various harms and algorithmic fairness by reading literature (e.g., P9 mentions the diagram from the Algorithmic Justice League) or by following courses on ML in general, on AI ethics, or on ethics of technology. The way the course is taught seems to impact practices, as one developer discussed having been trained through use-cases and was able to identify a number of harms, while four others mentioned a few ML ethics courses with toolkits introduced during the courses but did not reflect on any harm during the interview.

Importance of the Design of the Learning Material. While developers learn and develop their experience with ML and algorithmic harms via various means, leading to various practices, they also seem to interpret differently the same material, sometimes leading to misconceptions. While we discuss in a later subsection relevant human factors, we emphasize here the importance of the framing of the materials around harms. For instance, certain initiatives, although having a legitimate aim —warning against issues or proposing relevant approaches— sometimes had the inverse effects, and narrowed down the view of the developers towards related harms. This was especially the case for the recent "data first" approach advertised by different research communities [42], that led certain developers not to understand that model design might also create algorithmic unfairness; *P22 "I talk about the data quality first like Dr. Andrew Ng says. Data-driven ML is becoming very prominent."* Similarly, P9, P16, P23 learned about model energy-consumption issues by reading the "Stochastic Parrot" paper [93], leading them to acknowledge these issues solely for large language models, but not for other types of simpler ML models.

Next to the framing of harms, the vocabulary employed (e.g., "bias", "sensitive feature", "protected attribute") also revealed to be a source of confusion and flawed practices. For instance, certain fairness-inexperienced developers only conceived "biases" as statistical skews without relations to, e.g., sensitive attributes or harms *P30 "with medical instruments, for a specific machine, there is some specific noise in the data. If you know which machine measured the blood pressure, then you know the bias in the data."* Some expert developers even warned about issues with loaded terms.

DISCIPLINARY EXPERIENCE

ML Experiences. The amount of experience with ML also seem to be an impacting factor for practices around algorithmic harms. We observed that developers who have longer experience with ML (independently of having experience or not with algorithmic harms) reflect about more harms, more in-depth, and often envision more diverse mitigation methods than less experienced developers. For instance, three of those developers without experience around fairness were able to envision potential harms from the model design, and naturally evaluated the model based on subgroups of population without knowing the concept of equalized odd, whereas developers relatively inexperienced in ML with some algorithmic fairness training often did not account for this.

Three participants who had extensive experience with data science but were inexperienced with fairness and three mildly experienced ones were also more critical about the toolkits. *P18: "You always need to question existing tools and practices to be able to improve and innovate."*

Experiences with other Fields. Three developers who have not only studied ML or data science emphasized the potential benefits of their background: a participant trained as an ethicist; another trained in industrial design *P1: "This is my industrial engineering background talking. Let's map out the process to see, if we would be using a model, where it would fit in the current process and what requirements might be there? Is this supposed to be a fully automated system? How are people going to use this system? [..] For that, I talk to people. Can you imagine yourself saying that? [sarcastic remark about computer scientists]"*; and a last one in sociology *P29: "that's why they hired me: someone who's both good on the computer science side and on this sociology side."* These participants indeed identified more relevant harms and presented a more critical attitude towards their own activities, reinforcing the importance of involving multiple stakeholders with a diversity of backgrounds when the ML developers themselves do not have the relevant education.

PERSONAL FACTORS

As we hinted at earlier, developers might behave differently even when presenting similar prior training and experience, within similar contexts. This hints at the existence of additional human factors that impact practices. Especially, non-volitional, sociodemographic factors were explicitly reported by developers as drivers of certain practices, such as gender, nationality, and culture that impact their ways of perceiving harms. Belonging to a minority might also change the lived experiences and efforts put onto harm mitigation. *P13: "I felt my obligation because I participate in many unprivileged classes. So I would like another person to do it for me."*

Although not always directly observable via our interviews, other factors (e.g., psychology traits, abilities, and the resulting personal interests) appeared to be at play. For instance, when asking the developers to envision potential limitations of fairness metrics and mitigation methods, many of them could neither envision any conceptual one, nor see the potential risks of distribution shifts (that is a more technical and well-known topic –mentioned by only 20% of the participants). Similarly, when we prompted the participants to reflect broadly about their approaches, many did not envision or acknowledge any potential limitation. Yet, some participants showed more reflexivity, accurately recognized being biased and having to make subjective, uninformed choices, and acknowledged the complexity and subjectivity of the choices they make. *P20: "I'm sure that there is a possibility to create bias if I create features based on my interpretation of the data or what I think in my subconscious about people that get ill."* A few (also recognized not really knowing the potential impact but potentially keeping the benefice of the doubt. *P4: "For hyperparameters like learning rate, I can't see the connection with how it might harm people because it just influences accuracy. But I'm hesitant to say it doesn't affect it at all because you never know with these things, so you should always be cautious."*

7.4.3. CONTEXTUAL FACTORS

Along the interviews, developers also mentioned a number of organisational factors that represent obstacles or impetus towards handling questions of algorithmic harms.

INCENTIVES AND SUPPORT

Several participants discussed monetary incentives (financial compensation) and non-monetary incentives and opportunities (possibility to get dedicated time for investigating harms), or the lack thereof, provided by their organization, that impact their considerations and actions. *P14: "the challenge is that, from a legality compliance and the organization perspectives, the appreciation should be there for you to spend the time."* Several participants mentioned engaging in volunteer work in their organization, in order to setup trainings and tools for tackling harms, or directly investigate harms for their own ML projects.

Others also reported on the material support (or the lack thereof) provided to them to facilitate tackling algorithmic harms. They especially mentioned the access to convenient tools (such as the fairness toolkits), and education around the topic (e.g., via the participation to workshops and seminars ordered by the organisation). Human support was also reported, especially the facilitation of the access to various relevant stakeholders (e.g., domain experts, decision-subjects, researchers) who might be able to give indication on the existence of potential harms and the way to solve them.

PROCEDURAL OBLIGATIONS

Procedural obligations were also reported by participants, as wishes to foster algorithmic harm considerations. In terms of requirements or guidelines for the ML system to be built, they reported that, oftentimes, the organisation did not specify any harm-related requirement, and that certain requirements would come in opposition to the mitigation of harms (due to existing impossibility results; limited access to data, e.g., due to cost, etc.) —a clear hindrance towards harm mitigation. For instance, P16 and P19 described that their decision to develop a system is based primarily on the system's usefulness (time and cost saved) for the business that requires it, leaving out questions about harms towards data subjects *P16: "It's appropriate and relevant for the business. They want to save money or to reduce time of work."* Subjective norms (the vision that the society might have on the organisation, or the belief that the organisation has on the way of handling harms of other organisations) also played a role in the establishment of requirements by the organisation. In certain cases, it made the organisation push the developers towards investigating harms, while in other cases it refrained them to do so —for instance, P13 mentioned that if the public knew about a certain harm mitigation approach, they would not accept the ML system deployment *P13: "[talking about post-processing methods that flip certain model outputs] They imply a bias in the process. It would be a problem for the company to say that they are doing this: if I am a company and I am saying publicly that I am imputing bias on my model, how would society react to it?"*

Next to inexistent, ambiguous, or contradictory requirements, the allocation of responsibilities towards harms was described as structurally unclear for the developers. Very few developers mentioned clear allocation of responsibilities by their organisation (e.g., existence of an ethics committee). This represented one more challenge for the

developers, as that did not necessarily provide them with the needed power to make choices towards harm mitigation. Particularly, participants often discussed that they can strive to make harms transparent within their projects, but that the model requesters have the final say in deployment decisions.

7.4.4. INTERACTIONS BETWEEN FACTORS

Here, we provide a short description of the main interactions we identified between factors, that reveal the importance of psychological traits and other human factors, and reinforce the need to account for the entangled nature of these factors.

PERCEIVED OR ACTUAL RESPONSIBILITY

We described that organizational factors might leave responsibility around harms ambiguous. In such situation, different developers react differently (hinting again at the importance of human factors): they perceive their responsibility differently, and engage to different extents in activities that are not promoted by the organizations in order to tackle harms. Certain developers argued that as data scientists that know the most about the system, they are the ones responsible for identifying and reporting harms (if not also for making decisions on system requirements and deployment) *P17: "It needs to be the responsibility of the developer, or have a developer that is some sort of fairness compliance person, that's doing some peer reviews of code, because once you get to the developers' boss, they don't know code."*; that the model requesters are the ones deciding for any requirement; that the C-level and managers should be responsible to incentivise the engineers and to make choices where developers do not have knowledge *P19: "As much as I would probably want to, I don't think I have all the necessary background for that."*; or that a committee within the organization should be responsible as it would gather more diverse expertise *P16: "We have a committee of ethics. If we have any questions, we can go there to understand their opinion, it will not be the decision of one person but a collective decision."*

OBSTACLES AND EFFORTS

We mentioned that developers might lack resources (e.g., access to relevant stakeholders) and knowledge to tackle harms. In such cases, we identify different attitudes towards the challenge. While it is well-known that collaboration in the ML lifecycle is often needed for the developers [220, 920, 637, 454], prior work and our study both show that tackling questions around algorithmic harms is still predominantly the job of ML developers alone. Except for certain highly-ML experienced developers, most of them did not mention putting proactive extensive effort into reaching out to relevant stakeholders. In terms of knowledge, many of the participants who admitted lacking knowledge to identify or mitigate harms, concluded by reporting that they consequently do not put effort into acting on harms. *P10: "I am slightly aware of it but I wouldn't be able to say how to make changes towards that. I don't have any experience."* Instead, others mentioned searching into research papers to identify appropriate methods. For instance, P15, P18, P24, P27 proposed to look into research that trades-off model size (assuming a smaller model would be less energy-consuming) and accuracy performance to reduce environmental impact. Some developers explained potentially having a higher propensity to put

effort onto fairness challenges because they have research experience, and hence can search within publications for relevant methods *P7: "I'm interested in research. When you try to apply these tools, that is connecting the academic world to the business side."* Similarly, when participants mentioned that no method exists yet to tackle a harm, certain would attempt to create a new one, while others would wait for research to progress.

7.5. DISCUSSION & IMPLICATIONS AROUND THE RESEARCH / PRACTICE GAP

7.5.1. THE RENEWED IMPORTANCE OF FACTORS

SUMMARY OF OUR FINDINGS

In our study, we found that a complex set of interdependent human and organisational factors interact, and result in diverse practices of machine learning (ML) developers around algorithmic harms. For instance, we identified that, overall, developers who have little experience with ML and have not received practical and critical training around algorithmic fairness often stop at the application of a few fairness metrics and mitigation methods. The more experienced developers and those with an interdisciplinary background present a more critical attitude, attempt to go beyond what fairness toolkits permit (e.g., by envisioning non-algorithmic ways to avoid algorithmic unfairness), especially when they had opportunities to discuss these topics with experts. Next to these prior experiences, organizational constraints and incentives also represent drivers or obstacles towards deeply tackling harms, that, in interaction with psychological and socio-demographic traits, result in a diversity of trade-offs made between algorithmic harms and other business considerations.

While it is natural that such types of factors impact practices in the context of ML model development and algorithmic harms, no investigation of such factors had been performed. This study provides a first qualitative investigation that bear broad implications, and whose output validity should be later investigated through quantitative studies. As toolkits cannot serve as straightforward recipes for the developers, developers should also be supported in exercising due diligence. We argue that this should go through the development of better means for knowledge dissemination and training, the design of supportive materials and new organizational processes, and the consideration of organizational factors.

A LUKEWARM PERSPECTIVE ON TOOLKITS

Our results bring evidence confirming the results of prior works on the use of various documentation and code toolkits, that have shown that these toolkits can indeed support ML developers in finding more algorithmic harms than without a toolkit [125, 220]. Yet, our results also bring more nuance to the benefits of toolkits, and show the risks of using those. These nuances had not been demonstrated in prior, empirical works on toolkit practices, as they did not focus on the impact of toolkits on algorithmic harms, but only on the correct implementation of algorithmic fairness methods. Our results also provide empirical evidence for prior broader works that argued against the techno-solutionism of algorithmic fairness [309], demonstrated the potential dangers of ethics washing [102], and more broadly warned against automating ML processes, e.g., through

AutoML [890].

Prior work [220] had not discussed major differences in usage of different fairness toolkits. We corroborate such findings. Besides, the factors we find developers mentioning as important for selecting a toolkit are well aligned with the insights of prior works on the use of these toolkits [220, 481, 679]. These works have developed, among others, rubrics for the design of better toolkits, including similar functionalities (compatibility with various models, inclusion of diverse fairness metrics, guidance along the entire ML lifecycle, facilitating interdisciplinary conversations, etc.) and non-functional requirements (e.g., learning curve, compatibility with common coding frameworks, etc.). We especially echo the recommendations they make to better guide developers along socio-technical considerations [882], in order to avoid the pitfalls emphasized by our participants. These prior works however had not discussed the contradictory evaluation of toolkits by developers, that we found in our interviews, and that would merit further investigation.

THE IMPORTANCE OF HUMAN FACTORS

Although prior works have sparsely investigated human factors that impact attitudes towards algorithmic fairness, we find a number of prior results that align with ours, and hint at the validity of our results. While these studies do not investigate ML developers specifically (but computer science students, or decision subjects), they are still relatable, as perceptions of fairness impact follow-up practices towards harms. Besides, our work expands on these prior results in that it looks at a broader range of harms, and at different types of individuals.

- *Toolkit.* A few works [220, 481] show the potential usefulness of toolkits and their current practical limitations. No study mentions potential negative impact that we identified.

- *Experience.* Kleanthous et al. [441] identified the impact that the level of computer science education has in understanding fairness issues along an ML pipeline, that we also identified. Yet, no study reveals the importance of the type of educational background and the type of prior ML experience and fairness training.

- *Socio-demographic factors.* Quantitative studies [405, 636] have shown the impact of gender on students' considerations of ML fairness, privacy, and non-maleficence. Prior work has also shown the effect of gender and race on judgements of fairness metrics [314, 338]. While this is not a result we could explore due to the imbalanced distribution of participants we had, all our female participants also displayed a critical attitude towards their practices and acknowledged various harms, whereas the results were more disparate across male participants.

- *Non-volitional factors.* Others [314, 544] found that non-volitional factors, e.g., political views and experiences with identity-based vulnerability, are relevant. Our results also hinted at the importance of non-volitional factors, as multiple developers referred to their personal interest in the topic, or being part of discriminated minorities, as motivating factors.

While the studies above align with our work, other studies seem contradicting. Some studies have not found impact of socio-demographic or other human factors on the perception of different fairness metrics [786, 314], and the results of other studies are contradicting each other in terms of fairness perceptions, as detailed in [338]. For example, Wang et al. [864] identified that people with higher computer literacy perceive algorithmic decision-making fairer than people with lower levels of literacy, and that age, gender, race, and education level do not have a significant impact. Contrary to these findings, others [405, 636] pointed to the impact of gender, and our work showed the variability in perceptions of fairness among all our participants who were highly computer literate. We argue that these contradictions are due to the absence of detailed investigation of the impact of the human factors we identified, or to the lack of relevant intersectional considerations across factors.

CONTEXTUAL FACTORS: OBSTACLES OR VECTORS

Our study identified various clashing constraints and objectives that developers have to take into account during the ML lifecycle. Some of these points have already been highlighted in previous empirical works, such as the conflict between business goals (e.g., the system should work for a majority of cases but not necessarily for edge cases to have a competitive advantage) and developers' goals (making sure to have high accuracy on all kinds of population) [621, 521, 612], or the lack of organisational support [665] (time and cost allocated, development of tools and guidelines, etc.), that result in individual efforts instead of organizational processes. Other factors had not been discussed until now to the best of our knowledge, in the context of practices for handling algorithmic harms.

7.5.2. REFLEXIVITY VIA RENEWED EXPERIENCES

Facing the importance of various factors, one should take those into account in the future development of support structures for ML developers to tackle algorithmic harms. Support should be personalised to the relevant types of developers we identified.

GUIDELINES FOR THE DESIGN OF TOOLKITS

While fairness toolkits mildly contribute to enacting reflexive practices around algorithmic harms, they still represent an almost inevitable medium for algorithmic fairness. They appear as double-edge swords according to our results. This is where the danger of breeding a "*Checkbox Culture*" can manifest among developers with respect to handling algorithmic harms. Our work especially shows the need for pointers to relevant activities and resources within toolkits [480], while emphasizing the complexity of the problem and its context-dependence. Toolkits should also be adapted to the type of stakeholders that use them, based on their prior training, experiences, and other human factors, showing pop-up warnings, enforcing attention checks towards harms, allowing for different functionalities, or proposing trainings before using the toolkits. This will be a challenge as existing warnings in FairLearn [107] do not seem to always be considered by the developers. Besides, we need to make sure the toolkits do not become new checkboxes, but instead foster critical thinking.

DUE DILIGENCE THROUGH EDUCATION

Topical Education. Since our results highlighted the importance of the type of training and experience developers have received about ML and harms, we join prior studies in advocating for more education of ML developers [220, 719, 454]. Many works [121, 269, 281, 662, 436, 117, 147, 362, 536] have discussed ways to provide a responsible AI education to developers, and we recommend to refer to their insights (e.g., modular approaches to responsible AI education for easy integration into courses, including events reported in news articles). We also recommend to rely on insights from farther domains such as data science teaching [474, 292, 788] (perhaps even more worrying than our results, low-ML-experienced developers also failed into well-known, non-harm-related traps, such as not reflecting on the limitation of accuracy as a performance metric), ethics and HCI [270, 249], or even ethics of long-established fields such as medicine [157], which have tackled tangential questions. We emphasize the importance of accounting for the breadth of the topic (only Garrett et al. [281] noticed the absence of certain harms like environmental impact from existing courses), its complexity, and the importance to raise awareness about the issues and to train on tackling them.

Change of Attitudes. Next to teaching about algorithmic harms, it is important to develop the moral sensitivity [121], the critical attitude, and the reflexivity of future developers, in this highly-subjective context (Green et al. [311] talk about an algorithmic realism approach, acknowledging the contextual, porous, and political nature of these harms and objectives) where no easy solution to algorithmic harm can be prescribed. Three concrete mediums of good practices surfaced from our interviews: discussions with diverse stakeholders to develop awareness around the subjectivity of the problem, warnings to develop a critical attitude towards existing theories and tools, and use-cases to experience potential challenges in the responsible use of tools. These should be incorporated in the trainings. We envision that trainings using close-to-real-world use-cases, starting from the beginning of the ML lifecycle (problem formulation) to the end (deployment and monitoring), with various stakeholders to interact with, and varying degrees of challenges (e.g., having all harm-related and other constraints explicit or proactively identifying them), could be beneficial. Markus and al. [535] insist on accounting for organisational dynamics in such trainings.

Terminological Considerations in Education Material. The terminological confusions we identified align with prior works [571] that highlight disciplinary confusions in the task of making a model fair, and works that studied the impact of terminological choices [467] on one's perceptions of an ML system. Mulligan et al. [571] promote the value of shared vocabularies and reconciling taxonomies that facilitate discussions. We echo these recommendations and the ones of P29 who suggested to move away from loaded terms towards more specific words, e.g., characterizing the type of bias in relation to the harm it creates, arguing that these materials should not only contain definitions such as it is currently done [518], but should also make concepts clear to the extent of pointing out to the different related theories behind them.

ACKNOWLEDGING CONTEXTUAL FACTORS

While these factors are often unspoken in the research community, they have to be accounted for by developers, as they are inherently in tension with handling algorithmic harms, but most developers currently face the dilemmas alone. We argue that the research community and policy makers should account for these factors further, and support —sometimes empower— developers in the decisions they have to make along the ML pipeline. Interdisciplinary research is needed to understand how to prioritize tackling the different harms (beyond distributive fairness), accounting for realistic trade-offs that have to be made across stakeholders and acknowledging practical constraints. Relevant directions are the understanding of preferences of stakeholders beyond well-studied preferences across fairness metrics [314, 338], the development of frameworks to uncover and negotiate preferences between stakeholders [836, 146, 484], and the creation of guidance for developers to navigate the trade-offs.

Knowledge and due diligence are not enough when developers do not receive structural incentives. P18 mentioned *"Practice is different from the ethical goals of the world. I had an interview. I said it's important to recommend people music that is worthwhile listening to. The manager told me these are idealistic thoughts, not how the real world operates, this company is all about revenue. So fairness at a company level, it depends on the culture and ethics of the people."* Hence, we join [665] in the idea of developing organizational processes to foster the development of good practices: the design of guidelines [523], e.g., for identifying responsibilities and appropriate requirements, the facilitation of interdisciplinary collaborations [882, 662], and the establishment of structural incentives and principles such as slowness [615]. Development of regulations, that explicitly account for organisational obstacles (e.g., making sure some employees of an organization are well-equipped to investigate algorithmic harms, have time dedicated for it) could also incentivise these organizations [299, 772, 845].

7.5.3. RIGOROUSLY INVESTIGATING THE FACTORS

The factors we identified should be quantitatively explored in the future to validate our results (identified conceptions for each harm could serve as dependent variables). This would inform the design of trainings and supportive tools (e.g., the categories of individuals to tailor them to), and the constitution of ML development teams, accounting for the perceptions and abilities of each member. We foresee challenges in the design of a rigorous experimental setup: difficulties to quantify human factors, need to account for interactions between them, and need for specific scales around each harm, their different perceptions, and mitigation approaches. Apparent contradictions among results of prior works seem to be due to subtle differences in what is measured, who is the experiment subject, and potential interactions between multiple factors, which are differences that one should aim at controlling in future studies.

Existing research could be used to overcome these challenges. A measurement has been developed to quantitatively measure undergraduate student's attitudes towards the ethics of AI [405], that could be useful to evaluate how these factors are impactful. Yet, one should first complete this instrument to account for the types of harms that are currently left out from the instrument and for which we identified a variability of conceptions, and not only for attitudes towards harms but also towards their mitigation. The

insights and methods from social psychology studies about human processes of taking actions, such as the theory of reasoned action or the theory of planned behavior [14, 333], could also be adapted to further analyse results, as they hint at a diversity of factors and their co-existence, for action taking. We already see correspondences, for instance in the subjective norms and perceived control mentioned by these theories, and that our interviewed developers also discussed, e.g., when mentioning the image ML ethics give to an organization.

7.6. LIMITATIONS & THREATS TO VALIDITY

While we strived for recruiting a diversity of participants in terms of demographics, experience with ML and fairness, we could not obtain a significant sample for combined categories. Impossibility came from the relatively small amount of developers tackling these issues in the world (e.g., few developers could be found working regularly with the AIF360 toolkit), the duration of our interviews, and the controversial character of the topic. Yet, since several of our observations are corroborated with previous studies, one can suppose some generalisability of our results. This also indicates future challenges in quantitatively investigating the factors.

Due to time considerations, developers could not extensively explore the toolkits beyond our tutorial. Letting them familiarize themselves further with algorithmic fairness before conducting task T2, would possibly provide a few different results on the impact of experience and toolkits on practices as practices evolve long-term. For instance, FairLearn provides warnings about algorithmic harms that the participants did not see during the interviews, but that could change their attitudes. Yet, the interviews with developers experienced with toolkits allowed us to somewhat control for this, and did not show related differences.

Finally, our participants were not placed into a specific organization and did not have access to different stakeholders. While this was useful for us to fairly compare practices across participants, we foresee the importance of further studies, e.g., with the developers' own projects, to identify additional factors.

7.7. CONCLUSION

Our study led to an extended characterization of the complex, intertwined, factors (technical, toolkits, human, and organizational) impacting the differences of conceptions and practices about algorithmic harms that surface across ML developers. These results do not only align with prior works that surfaced a few factors in relation to algorithmic fairness, but also extend and complement these works with information around a more comprehensive consideration of algorithmic harms. Particularly, we found that the use of fairness toolkits does not necessarily lead to its envisioned impact, and can at times promote a checkbox culture, if it is not accompanied by a distinction of the background and prior training the user of the toolkit received, as well as of the pressures their organisations puts on them. In summary, our study constitutes a strong testimony that ML developers are not as much "ethical unicorns" [662] (i.e., developers who ensure a comprehensive handling of algorithmic harms of the ML systems they work on), than *subjective unicorns encaged in an organization.* Such findings bear strong implications for

future research opportunities around the refinement of the toolkits and of educational programs, accounting for these human factors, and for potential regulations to address organizational concerns. In the next Part (Part III) of this thesis, we will develop initial technical solutions to support ML developers in diagnosing some of the hazardous failures in the outputs of their ML models. We leave it to future work to tackle the other factors that we identified in this Chapter.

APPENDIX

DETAILS ON THE METHODOLOGY

INTERVIEW PARTICIPANTS

Table 7.1 introduces the distribution of participants to our interviews.

Table 7.1: Background of the participants in our study. Note that some participants reported multiple educational backgrounds.

Dimension	Values (and number)
Demographic information	
Nationality	US (6), Netherlands (6), India (4), Iran (2), Russia (2), Romania (2), Sint Maarten (1), Canada (1), Brazil (1), Slovakia (1), Poland (1), Greece (1), Spain (1), Ukraine (1)
Gender	male (24), female (6)
Highest education	BSc (2), MSc (21), PhD (7)
Experience with machine learning	
Work type	applications (14), research (8), both (8)
Application domain	healthcare (4), finance (3), recommender systems (related to human resources) (3), predictive maintenance (1), others
Education	computer science (25), mechanical engineering (3), business or economics (3), sociology (1), psychology (1), accountant ethics and compliance (1)
Years of experience	2 or less (13); 3 to 5 (15), 15 (2)
Experience with algorithmic fairness	
Years of experience	18 (1), 3 (3), 2 (7); 1 (2), 0.5 (7); 0 (10)
Type of experience	long-term research (6), short-term research (4), frequent use (7), irregular use (3), none (10)
Toolkit	no exp. then FairLearn (5), no exp. then AIF360 (5), exp. with FairLearn (11), exp. with AIF360 (9)

INTERVIEW USE-CASES

Table 7.2 introduces the harms we included in the two use-cases.

QUESTIONS ASKED TO THE PARTICIPANTS DURING THE INTERVIEWS

Questions on background experience. We started the interviews by giving a brief overview of our research to the participants, and by questioning them about their background (demographics and machine learning experience). Once all required tasks were completed by the participants, we asked final questions about their fairness experiences, how they learned and work with algorithmic fairness/harms, and reasons for using a certain toolkit, as well as their broader knowledge of the responsible machine learning field. We made sure not to ask any question related to their algorithmic fairness experience at the beginning of the interviews not to bias them towards thinking of particular topics.

Table 7.2: Examples of potential harms introduced in the two use-cases presented to participants.

Category	Task 1: Hospital readmissions	Task 2: Medical services utilization
Distributive unfairness		
Biased dataset causing unfairness	High imbalance for various potentially sensitive attributes (e.g., `race`: 74% Caucasian, 20% African American and the rest divided in 4 other categories).	High unbalance of `race` (white at 80%, others at 20%).
Sensitive attributes	"Classic" sensitive attributes (e.g., `gender, race`), and other, rarer, potentially sensitive ones (e.g., `marital status, weight`). Proxies (`region` was synthesized to be highly correlated with race).	Same with `race, sex, age`, and question of `marital status, military service`. Proxies (e.g., `race` highly correlated with `poverty status`).
Conceptual limitations of metrics	Consequences of the model output not only for the patients but also for their family, not measurable.	Consequences of the model output not only for the insured but also for their family, not measurable.
Harmful datasets		
Inappropriate attributes	Utility and ethics of using the `marital status` to predict hospital readmissions.	Same for `marital status`, and `military service status`.
Inappropriate attribute encoding	`Gender` encoded as binary, `age` encoded into three categories.	`Race` encoded as binary (white, non-white).
Desirability of the ML model		
Task encoding desirability	Over-simplified and potentially irrelevant target labels (unjustified threshold of 30 days).	Potentially unethical task where insurance prices would be computed based on estimation of medical services utilization.
Impact of technical ML activities onto harms (especially unfairness)		
Missing data	Synthetically introduced to correlate with specific values of the `weight` and `medical speciality` attributes.	21% of synthetically introduced missing values for the `weight` attributes with primarily values corresponding to `gender` female, which would lead to gender imbalance if the corresponding records were droppped.
Outliers	Synthetic injection of outliers in the number of `lab procedures` attribute	Outliers introduced within one synthetic attribute corresponding to an aggregation of several other attributes.
Duplicates	No visible duplicates.	20% of synthetically introduced duplicates, that would decrease dataset size consequently as well as create certain target label inbalance if dropped.

Questions on higher-level reflections. At the end of the interviews, we also asked general reflection questions about any other considerations they might have when building models, any additional harm they could envision, their experiences with the fairness toolkits that we had introduced (for practitioners who previously did not know these toolkits) and potential changes they would like to see in these toolkits, about algorithmic fairness and whether it can be solved as well as on the limits of fairness metrics and mitigation methods (when not mentioned earlier), about their responsibility in considering algorithmic harms, and about any other wish, doubt, or remark.

Questions on the process. While the participants were working on the tasks, we asked them questions about their process, in order to understand the reasons for performing each exploration activity, the thoughts they had when seeing the results of an exploration, and the actions they would take based on these results, as well as to make sure they had not forgotten any activity. We especially questioned them on activities that might have a connection to algorithmic harms (e.g., observing data distributions and rebalancing the dataset based on the target labels). After the two tasks in the case of the participants inexperienced with toolkits (not to bias the participants towards certain reflections when looking at the second task), and after the first task for the other participants, we further questioned them on the algorithmic harms they had not investigated (whether they usually consider them, why or why not, how they would handle them) during their exploration of both tasks, and on the harms that could be resulting from the activities they mentioned. We identified the harms we posed questions on through our analysis of the literature, and we also coded any other harm they could mention. We made sure to first ask vague questions (e.g., what can be issues with the activity of labeling data with crowd workers), before going onto more specific questions (e.g., what do you think of potentially poor labor conditions of crowd workers), so as to see to what extent the practitioners actively think about these harms.

OTHER MATERIALS

Tutorial. The tutorial consisted in presenting the concept of algorithmic fairness, the ways different fairness definitions are computed and different mitigation methods are applied (concepts of data pre-processing, model in-processing, and output post-processing), as well as illustrating the use of one of the toolkits to apply these definitions and mitigation methods. We gave the tutorial with a third use-case dealing with the prediction of credits default [363, 908]. This use-case was chosen for its popularity within tutorials on algorithmic fairness and toolkits, so as to be as close as possible to what a machine learning practitioner might see first when learning about algorithmic fairness.

To give the tutorial, we shared our screen with the participants, showing a Jupyter notebook we had prepared with these concepts and examples of application of the tools on the credits default dataset. We especially presented the computation of some of the metrics on a simple logistic regression classifier, and on the same classifier to which various mitigation methods (e.g., the threshold optimizer and grid search algorithms of FairLearn, as well as the reweighing and prejudice remover algorithms of AIF360) are applied. We made sure to answer any question the participants had during the tutorial and later when provided with their second task. At the end of the tutorial whose aim was to give the participants a basic introduction to algorithmic fairness and toolkits, we asked for verbal validation from the participants to confirm we achieved our goal.

Notebooks. When working on these tasks, we made sure to reassure the participants that they did not have to code the entire exploration they would perform (only if they wished to), but they could also simply speak out-loud and report on what they would do. We had already prepared additional notebooks with code snippets that the participants might want to use, and we shared these snippets with them whenever they would mention a certain exploration activity that would correspond to the snippet. This allowed to

reduce the complexity of the session for the participants, to accelerate the process, as well as to see them reflect about concrete results of the exploration activities.

Pilot Studies. Before performing the interviews, we performed two pilot studies with practitioners working at our institution. These two studies allowed us to check for the understandablity of the tasks, to refine our questions to prompt about the different harms, to better time each task, and identify relevant reflection questions, as well as to make sure that we had prepared enough code snippets to help the practitioners.

ADDITIONAL RESULTS

RESULTS ON THE FAIRNESS TOOLKITS

Table 7.3 introduces the properties of the fairness toolkits (functional and non-functional requirements) that practitioners reported as important when choosing which of the available toolkits to adopt.

RESULTS ON PRACTICES AND VARIABLE RATIONALES AND FACTORS

Table 7.4 describes the types of rationales our participants express when handling algorithmic harms. These rationales hint at different factors that impact the practices. Tables 7.5, 7.6 describes the types of challenges and impossibilities our participants envision when handling algorithmic fairness, showing the diversity practitioners have in the way of thinking about these problems.

Table 7.3: Properties of toolkits highlighted by the practitioners.

Property	Example	Comparison and contradictions
Compatibility with coding frameworks	*P3 "FairLearn is natural to use for those who work with scikit learn because it is the same API. But a lot of models work with a huge amount of data, using MLLeap, SparkMLleaP, here FairLearn will be much harder to implement."*	*P13 "AIF is a really good library because it has Scikit learn. This library has this kind of compatibility with the pipelines that I already use."*
Compatibility with production	*P12 "it is not being updated often, it has the dependencies of older versions of Scikit where something was changed. So it is not perfectly maintained. So every time you add something to your production, you'll want something that will be updated often or don't have many dependencies."*	-
Maintenance	*P3 "if I want to use something, I look in which stage it is. Although AIF360 has many stars, the amount of issues shows it is less handled than FairLearn. So I prefer FairLearn: if there is a bug, it will be fixed earlier." P13 "First, it's taken care of by other people, not by the company that I am."*	-
Open source	*P28 "In my next models that I will train, if there is a free (open source) tool, I will check it out and try to apply it to get more insight about how the tool works."*	-
Ease of use /extension	*P1 "AIF360 uses this ridiculous data structure that Doesn't allow you to. I mean, have you tried to put in your own data set in F 360? How easy was it?"*	AIF360 is mentioned as more complex to apply than FairLearn.
Functionalities	*P7 "AIF360 is more complete because it has most of the FairLearn functionalities and a few more mitigation algorithms for the group fairness and individual fairness, that is very new" P21 "FairLearn is somewhat more limited in terms of fairness enhancement because it doesn't have anything that affect the model during training"*	FairLearn is often mentioned to have less metrics and mitigation methods available, yet one practitioner mentions its advantage in presenting disaggregated metrics.
Adaptability to algorithms and tasks	*P2 "it's designed for tabular data mostly so there are a lot of different types of data, it's a work in progress." P7 "in the financial industry, some of those techniques that are published as a paper or announced in some standard packages and libraries, may not be very applicable for your problem"*	Mentioned for both toolkits.
Learning curve	*P21 " for fairness, you'll have a lot of problems during your job and you can't have someone who you're hiring and they will need a week or two to learn the toolkit. The learning curve is quite high."*	-
Transparent implementation	*P3 " Fairlearn was more natural because it's simpler and the majority of the things there are not black boxes. With AIF360, there are lots of things based on threshold optimizer or things that are machine learning models, biased as well. So I would prefer to work with somethingmore transparent."*	FairLearn would be more transparent (only one practitioner discussed this point).
Documentation	*P6 " they invested a lot in their tutorials and and all the other their guides and that that was really nice to see. and they made it very easy to use."*	FairLearn tutorials are often mentioned *P29 "An issue I have with AIF360, they don't have a lot of documentation on how to do this."*, but one participant mentions preferring AIF360. *P21 "AIF is definitely better with a lot more guidance materials."*
Socio-technical considerations	*P29 "Our choices are more deliberate about what we encourage or not, because there is this danger of giving people many tools and not educating them about what they mean. That's a big limitation of AIF360: if you use this tool with some definitions of fairness, then you will be able to solve your problems with very business solutions."*	FairLearn argued to provide more socio-technical information.

Table 7.4: Conceptions around algorithmic fairness metrics and mitigation methods, and their handling. These conceptions and practices reveal the fragmentation that takes place across practitioners around harms.

Conception	Example
Rationales for selecting metrics	
All available metrics (P2, P9, P10, P11, P14, P16, P18, P19, P26, P27, P28)	*P2 "because this model will work in hospital with patients where fairness is important, we check all the group fairness metrics of FairLearn."*
Prioritizing group accuracy or group output distribution metrics based on use-case type (e.g., distribution of resources, hiring) (8 participants)	*P1 "It's quite important that the model is accurate if resources are being distributed, like whether you receive care. So it depends. In some cases, you care about whether the model is accurate. In some cases you care more about whether the same proportion of people get a resource."*
Prioritizing specific group accuracy metrics by weighing different errors	*P6 "False negatives and false positives are both damaging. I'd have to think of the costs of those two sides, that informs what fairness criteria you would choose."*
Involving external information (experts or laws) (P1, P4, P6, P8, P12, P19, P22, P28, P29)	*P8 "Depending on domain knowledge, you want to know what metric you want to look at. Just by myself, I wouldn't really have an idea. This is either some legal stuff or just some ethical stuff that we want to make sure that's OK. "*
Using their own intuition	*P11 "There are a million different metrics. I would probably go down the list."*
Judging when the metrics values are satisfying	
Acceptability for the data subjects	*P29 "Absolute fairness is not possible to achieve. So it could be: yes, there is some disparity, but let's say the impacted communities sort of feels fine about that."*
Acceptability for the model requesters	*P19 "I don't think it's possible to remove the entire unfairness. But I think that's dependent on the people that they're making the model for, and how they react."*
Acceptability for experts	*P6 "There's a question of what is an acceptable difference in performance, it's difficult to answer, that's something you talk to all the stakeholders about."*
Rationales for selecting mitigation methods	
No mitigation can/should be done because the data represents the world	*P23 "some biases come by nature, like the data given the situation happening in the real world. That's not something you can change, it's by nature happening."*
Based on image it brings to the company	*P13 "[talking about post-processing methods that flip certain model outputs] They imply a bias in the process. If I am a company saying publicly that I am imputing bias on my model, how would society react?"*
By experimenting	*P21 "try out a few of those algorithms, see if they maybe work better."*
Preference for not simulating new data	*P22 "if possible, we want to re-sample the data instead of simulating data. I prefer if they can get the data from the source corrected, as much as we can."*
Preference for changing the data (P9, P15, P16, P19, P20, P24)	*P9 "if you can get fair data, that is one of the best ways to make sure that your classifier is accurate on all types and all representations of people. More data has always been the best way to make an ML model more accurate."*
Admitting not knowing how to choose	*P11 "I would just like read up on it so that I know about this strategy is better."*
Mentioned limitations of the metrics	
No limitation envisioned	*P19 "I think for fairness these metrics work well."*
Limitations of certain metrics said to be fulfilled by others (P8, P10, P21, P24)	When asked whether one metric such as demographic parity is enough, they answer no but instead they can use another metric like equalised odds.
Limited to account for exploitation of outputs by decision-makers	*P3 "it reminds me of this famous child benefit scandal, when the problem was not a model, but the people who were using these predictions. They were doing this manual post processing of predictions according to their beliefs."*
Dangers of fairness metrics to be used as checkboxes (P3, P6, P9, P13, P29)	*P6 "It's easy to think: we checked the fairness box because we implemented this specific library, or this constraint when really fairness is a much broader topic."*
Dangers of fairness metrics to remove critical attitude (P3, P6, P9, P13, P29)	*P13 "Responsible AI is an AI built with high-quality processes, not only regarding fairness, but regarding using the best metrics. Have a critical point of view."*
Mentioned limitations of the mitigation methods	
Non-applicability to certain types of tasks / algorithms	*P7 "we needed to mix up some approaches in order to customize them and modify them. In some cases, there is absolutely no methodologies to tackle individual fairness mitigation, that can be applied on the loan adjudication use case."*
Impact of one method on fairness	*P21 "Optimizing for one fairness will make another type of fairness worse."*
Does not fix structural causes of injustice	*P2 "About demographic parity, you can positively discriminate to get these outcomes, or you can make the model work less good for the majority group. I wouldn't consider that fair."*
Approach might not be ethical	*P1 "One thing that people commonly do is use different decision thresholds for different groups. What this means is that you literally put people to a different standard. And then whether that's justifiable or not, it depends on the scenario."*
Biases users to take technical mitigation approaches when they might need to be structural	*P29 "If you find some disparity, what does that mean in the real world? what is the intervention you take? If you don't understand the harm, you can't take an intervention to stop it. Often there's an intervention that isn't technical."*

Table 7.5: Examples of impossibilities mentioned by practitioners along their process, that reveal fragmentation of practices across practitioners, and various types of factors impacting practices.

Type	Example
Inherent statistical and theoretically clashing impossibility around algorithmic fairness and absence of harms	
if considering all sensitive proxies	*P21 "We are going into territory where fairness becomes almost impossible, because it could be that Medicare and Medicaid are a proxy for demographic features: whether minorities are more likely to take Medicare and Medicaid."*
because of all attributes being possibly sensitive	*P17 "I guess the only one that society has said it's OK to be biased on is smoking because it is probably the only one that you have conscious decision you can make about although you could argue that depending on where you're born, it is probably different probabilities."*
simultaneously for multiple metrics	*P21 "optimizing for one type of fairness will suddenly make another type of fairness worse. if I optimize for fairness between individuals, it's possible that the fairness between groups will suffer, but also even one level lower, if I optimize for predictive parity, it's possible that the disparate impact will suffer."*
theoretically clashing objectives around algorithmic fairness and absence of harms (e.g., privacy around data attributes and their encoding, fairness, and accuracy)	*P9 "Is the dataset collected in a way that had the informed consent of people in the data set? Or are we collecting hospital records and using that data to do something that patients were not made aware of? This healthcare case is sort of limited with what you can do because you're under health care data constraints like HIPAA."*
Theoretically clashing objectives around the use of ML and the absence of harms	Employing ML itself might be the subject of trade-off, as it might be useful for various stakeholders to deploy an ML model, but this model would require privacy-infringing data (P19), or might negatively impact the environment (P28).
Objectives clashing with harms	See below for these objectives.
Requirements on model objectives: Typically no requirement on algorithmic fairness and other harms	*P7 "For example, we had a company involved in paper recycling. In that case, we definitely need to make sure that the amount of data that we are requesting or any other request that we have from the client wouldn't have any side effect on the environment."*
Requirements on system infrastructure: Deployment requirements, e.g., easiness of deployment, of update, and of monitoring, and running time	*P29 "do you want it to be a simple model so that you could retrain it properly? Do you want something that's very small, so you can deploy it on like a AWS or on Azure" P3 "The simpler is the model, the easier it will be to deploy, the easier it will be to monitor, and the easier will be to retrain"*
Computational power in relation to environmental impact (only 2 practitioners)	*P15 "We have like 20,000 GPUs and it gives a very high accuracy like human level. On the flip side, you have this much power budget and then how do you obtain this same accuracy within any alternative algorithm? Can you achieve the same with much less compute power?"*
Data constraints: Availability of data samples/attributes, feasibility of collecting new data records, feasibility of collecting new data attributes / / impact training dataset, choice of algorithmic, resulting model performance	*P5 "after I do this, one of the first things that I would consider doing is to see whether This data set is sufficient enough For running a model. sufficiency test comes from 2 perspectives. One is What kind of Choice of model that I want to use. if the data set is not large enough, I cannot use a neural network, I would End up using a Linear kind of a model which would basically have its own limitations. I would want to be Clear of that.*

Table 7.6: Examples of impossibilities mentioned by practitioners along their process, that reveal fragmentation of practices across practitioners, and various types of factors impacting practices.

Type	Example
Impossibility due to the complexity of the concept of fairness	
complexity of the concept of fairness	*P6 "I don't think you can reach a fair model because it's hard to measure."*
complexity in accounting for the impact on other stakeholders	*P25 "[would you consider how different people might be Affected by the same output?] Should be considered, but I don't have a way to consider it in terms of improving the model."*
complexity in accounting for the impact on individuals	*P18 "This is something that we should take into account. I'm not really sure how to take those into account. Maybe we could add the number of children or add more features in the data to make sure that these decisions actually. To account for those specific differences, I think that's really hard and really subjective."*
Impossibility due to the subjectivity of the concept of fairness	
Some practitioners seem to think that despite their subjectivity, there is in theory one appropriate solution that could be defined for a certain context or at a certain level (e.g., a single country)	*P28 "I wouldn't say that someone has to have a different insurance premium when we talk about sex or race. So we would make those variables as protected. I would also say potentially age since at the end of the day, if you make it a constant that will make lives for people easier. But I think our society accepts the fact that there are different premiums if you are older. If you are in your working years or if you are young adult or you were just recently born."*
As fairness is subjective (e.g., on the culture-level or individual-level), it is difficulty or impossible to envision a one-size-fits-all approach at any level	*P6 "I think you can ever say that you're absolutely fair. And I don't think you can ever agree between two people what their definition of fairness is. So I don't think you can reach it and I think it's because it's hard to measure and it's hard to agree what the criterion should be."*
As interests are clashing across types of stakeholders, it is impossible for all to be satisfied simultaneously	*P21 "Ultimately, everyone cares for a model that performs well. The problem is that a model which performs well for the hospital is not necessarily a model that will perform well for the Asian people who go to that hospital."*
Subjectivity not only for algorithmic fairness but also for harms like feature encoding	*P16 "[talking about gender being binary in our dataset] I believe that everyone can be whatever they want to be. So the data itself should respond on this society request. So I mean it is a science request and we have very complex society. And if we have an issue with describing ourselves, we need to somehow mitigate it."*
Impossibility due to the "limits" of the practitioners (assuming algorithmic fairness is reachable in theory)	
due to limited knowledge of the practitioners and lack of guidance / regulations	*P21 " a person who would like to learn how to build a model and is confronted with a choice of 17 different mitigation techniques will know which one to choose? Probably not" P29 "This healthcare case is sort of limited with what you can do because you're under health care data constraints like HIPAA, but I think there's a lot of other use cases where there is no regulation about what companies can do with the data they collect, and that led to a lot of issues."*
due to biases of the practitioners and domain experts	*P8 "most of the time with the help of someone having domain knowledge because even though it could be that an expert has some unknown bias thinking "oh, we should probably look into this group", it is also domain knowledge.".*
due to biases of the tool developers	*P16 "someone decided that we'll go this way with these metrics. Because of different cultures, let's say a group of people who decide that equality between men and female is irrelevant, what we will do with this toolkit?"*
due to lack of tools available for the practitioners	*P11 "I would make weights protected. It's a bit tricky 'cause it's continuous, and I don't know if there are fairness metrics for that."*
(Process) Impossibility due to the lack of incentives and time given to the practitioners from their company or model requesters	*P14 "the other challenges is that, as I told you, from a legality compliance and from the organization perspective, the appreciation should be there for you to spend the time. I don't feel like it's still there." P22 "Everybody has deadlines and this is going to add to the work. But it is important in the long run."*
Handling impossibilities	
Making the least-bad choice (with intuition or external inputs)	*P30 "It's impossible to optimise for everything, so I need to pick a specific metric that I'm going to look." P21 "This boils down to making a rational choice of what are we trying to optimize at the early stages? And then keeping in mind that making some fairness metric better, even a lot better, it can still negatively influence other metrics."*
Neglecting the issue and focusing on model performance	*P18 "This is not of my concern as in having to include, for sex, 20 categorical options. At the end of the day, we're not doing politics, we're trying to solve a problem. But if the results that we obtain are poor because we did not take into account these attributes or variables, then we should include them."*
Not accounting for limitations of fairness metrics because they are better than nothing	*P8 "if you don't depend on metrics then how are you going to evaluate your model? You need to have at least some metrics to be able to say a) my model is fine, and b) my model doesn't have any harmful applications."*

III

Proposing Solutions For Hazardous Failure Diagnosis

After our broad inspection of the research/practice gap in Part I and II, we now propose solutions to this gap, to support developers in developing less harmful models. In Part II, among others, we found that explainability methods partially align with the needs that the machine learning developers have when debugging their models, but that they do not provide all the information needed (e.g., developers need domain expertise to understand when a model is failing), or do so in a non-interpretable manner (e.g., while developers need global explanations about the models' behavior, local explanations could potentially provide such explanations but necessitate too-high cognitive load for the developers to extract them). This calls for algorithmic and design research for the development of novel, more adapted, methods to extract the required information and present it in a usable manner to the developers.

In Part III, we address the development of supportive methods and tools for debugging models in terms of robustness issues. Specifically, we tackle one of the most urgent challenges identified. Developers cannot accurately estimate the brittleness of their model during development as they do not have access to production data —instead, they can only evaluate the potential for output failures of their model on the training data, that might suffer from distribution shifts in comparison to the production data—, but explainability methods seem to have potential to provide them with some relevant information towards this objective. This challenge bears repercussions both for output failures that can translate into physical harms, and for biased output failures that translate into unfairness and other social harms. This challenge calls for technical research enquiry and for human-computer interaction research, types of research we aimed at performing in this thesis. Note that we do not deal with evaluating the harmful impact that different types of output failures can have, but with evaluating the potential the model has to output any type of failure. We leave for future work to bridge the gap between the failures and the harm they cause. We also leave out for future work the other issues constituting the research/practice gap identified in Part II, such as the lack of awareness of developers around relevant technical solutions or the potential for harm of their choices, or the lack of incentives to tackle the harms due to, e.g., business pressures, unclear responsibilities, etc. These issues call for the development of education programs to teach developers how to make appropriate choices in relation to harms along the machine learning lifecycle, and the creation of policies and regulations to overcome the organisational obstacles, that would be the subject of a different thesis.

As a start, we tackle the challenge in the context of deep-learning based computer vision models, for which numerous explainability methods have already been developed. Our main proposition consists in shifting from using accuracy metrics on test datasets to estimate the harmfulness of models —limited because the test accuracy might not reflect production accuracy due to distribution shifts between test and production data—, to exploiting information about the mechanisms the model uses to make predictions on test data. We use the term *mechanism* to refer to the association between the various *concepts* present in an input sample, that a model uses to output a label *prediction* for this sample. We argue that mechanism accuracy is more informative than test (prediction) accuracy in terms of potential harms a model might cause in production. Indeed, a model might employ a same (wrong) mechanism for making predictions on multiple samples, and while the predictions about test samples might be correct using this mech-

anism (high prediction accuracy), they might be incorrect when dealing with production samples (low prediction accuracy). Hence, identifying mechanisms based on the test set, and assessing their correctness from the test set, can already prevent more prediction issues, that looking only at the prediction themselves that would present a high prediction accuracy. Note that focusing on the prediction mechanisms of a model also allows to estimate additional harms beyond the ones related to a model's outputs. These additional harms are the ones related to improper features (e.g., offensive, inappropriate, non-volitional, illegal features) that might be used by a model to make predictions, as a model's mechanisms reflects the features of a computer vision model. To achieve our vision of mechanism-based harm estimation, multiple challenges need to be overcome, that we investigate in the next chapters.

When investigating literature for potential mechanism identification methods, we have identified a plethora of works that propose explainability methods, that should allow to identify a model's mechanisms. Yet, when listing the requirements for the nature of mechanisms that would be useful for harm estimation, we do not find any explanability method that fulfills these requirements. Hence, in Chapter 8, we ask:

> ***R7:*** *How can one collect easily-interpretable mechanisms learned by a model for making predictions?*

We propose a new approach to collect such mechanisms, that builds on top of existing explanability methods and extend them, to make them fit our requirements. Especially, the explanations we provide are textual and global about the model mechanisms, instead of typical visual, local explanations, allowing for this higher interpretability. This chapter is based on a publication at the Web Conference 2021 [69].

In Chapter 9, we realize that in order to estimate the harmful power of a model, one needs to reflect on the appropriateness of the mechanisms the model has learned. In order to do so, we should contrast the model mechanisms with mechanisms a human would expect the model to learn. That is why we ask:

> ***RQ8:*** *How can one collect the mechanisms the model is expected to learn according to human reasoning?*

To answer this question, we propose a Game with a Purpose that allows to efficiently collect tacit knowledge, that could later on be translated into expected mechanisms. This chapter is based on a publication at the Web Conference 2022 (nomination for best paper award) [64], and a demo publication at HCOMP 2021 (best demo award) [63].

Finally, in Chapter 10, we need to evaluate the usability and usefulness of our proposed approach for model harm estimation. We ask:

> ***RQ9:*** *How can a developer use our mechanism information to diagnose a model's harmful power? How useful is this information in comparison to the one provided by existing explanability methods?*

To answer this question, we adopt a research through design approach. After a formative study, we co-create with 20 developers a user-interface that presents relevant information useful to diagnose a model's harmful power, and especially our mechanism information (both expected and learned mechanisms) next to explanations collected from other explainability methods. We then exploit this user-interface as a probe, in order to understand how developers can use this information, what challenges they face, and what other information would be needed for improving their harm estimation process. This study shows the utility of the mechanism information we provide to the developers. This study also allows us to collect new insights for the development of more usable user-interfaces to support developers in debugging their models –insights that we list for future work. This chapter is based on a publication at CHI 2022 [67]. We do not make any modification to the publications used in this Part, except in terms of reconciliation of vocabulary across publications, and small changes to the introductions and conclusions.

All in all, we propose a novel approach to estimate the robustness of a model, and allow for its instantiation by contributing a novel, human-in-the-loop, cost-efficient, method for identifying a model learned mechanisms [69], a game with a purpose for identifying a model expected mechanisms [63, 64], and a user-study to investigate to what extent and how developers can make use of this novel information in practice [67].

8

OBTAINING LEARNED MECHANISMS

8.1. INTRODUCTION

In Part II, we identified the need for machine learning developers to have different types of explanations of the machine learning models they develop and especially of their learned mechanisms, in order to better understand where these might fail, and for which reasons, so as to correct their models later on. We also noted that several types of explanations the developers were asking for have not been developed until now. This is the need that we address in this chapter.

To be effective, explainability methods must: (**1**) present interpretations that match humans' mental representations of *concepts* [6, 668, 534] as humans understand the world through concepts associated with observable properties. Human brains process visual information from low-level concepts such as color, contrast, to mid-level ones such as shapes, textures, and to more abstract semantic representations of an object. For example, an *ambulance* is "a car-shaped object that has a red cross or blue star symbol on it". And, (**2**) allow for the satisfaction of *interpretation needs* aimed at both model behavior *validation* and *exploration*.

A typical validation scenario occurs when a model developer (or auditor) tests precise hypotheses on the workings of automated decision making to ensure the system behaves as intended. In an ambulance recognition example (Figure 8.1), an auditor could ask "In the classification of ambulances, does the model focus on the `red cross and the flash lights`; or does it focus on unrelated background concepts like the `blue sky`?". In an exploratory scenario, the developer would be interested in understanding the classification behaviour of the model, but without a precise hypothesis to test. To support both scenarios, an interpretability method should be able to test for the *presence*, *combination*, or *absence* of *multiple concepts* with varying granularity –e.g. a model might learn to use an ambulance's `overall shape` (coarser granularity), or the `sign` on the frame *`and`* `the flash light` (finer granularity).

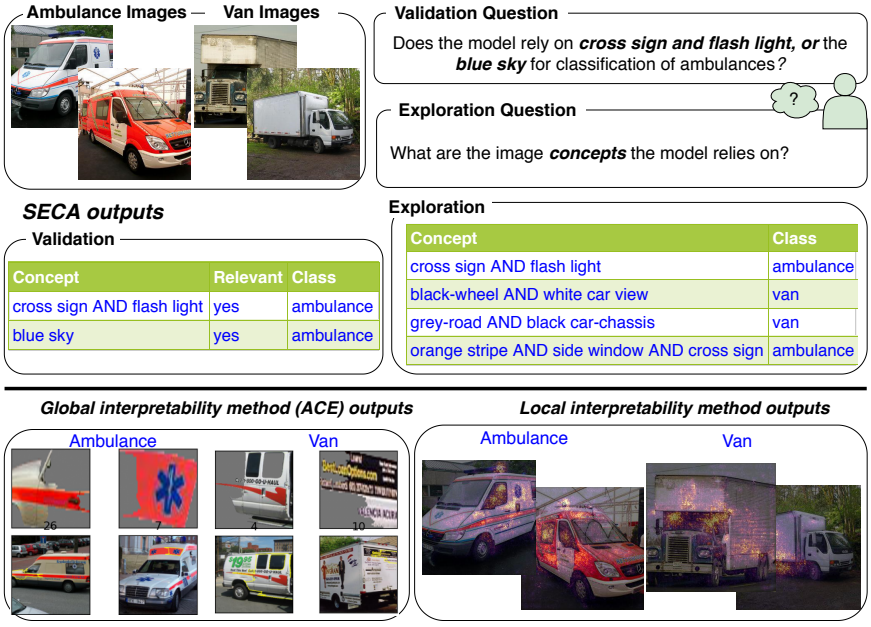


Figure 8.1: SECA generates (multi-concept) interpretations for both model behavior validation and exploration. In contrast, state-of-the-art *global* (e.g., ACE [289]) or *local* interpretability methods do not support multi-concept interpetability need, and generate image patches or saliency maps (for exploration only) that require manual interpretation.

Despite the recent advances in explainable machine learning [437, 289, 898, 677, 55], existing methods addressing image classification fall short in meeting the above requirements. We focus on *post-hoc* explainability methods, which, in contrast to *inherent* explainability methods (see a detailed discussion in Section 8.2), can be applied to any existing classification model. Among post-hoc methods, *global explainability* methods [437, 289] support exploration needs by automatically producing "patches" from multiple images in the dataset (ACE [289]) that should represent one visual concept inferred to be important for classification; or, for validation purposes, require users to provide a set of images (patches) as examples of visual concepts (TCAV [437]). Both approaches have shortcomings. First, they require manual analysis and interpretation to associate image patches with understandable concepts and properties [289], or require an input set of example images that cleanly capture the interpretation hypothesis the user wants to verify (e.g., images of ambulance with a cross sign but without a sky background) [437]. Besides, such methods do not easily support the validation and exploration of *multi-concept* interpretation. On the other hand, *local interpretability* methods analyse individual images [754, 918] and produce image-specific *saliency maps*, i.e. a highlight of the most important pixels for the classification of a given image. Local methods can be adopted for global interpretability, but with significant cognitive demand on users, both for validation and exploratory interpretation needs: multiple images must be individually analysed to associate image regions with intelligible concepts, and the respective concepts need to be reconciled globally and interpreted against the

classification behaviour of the model.

Arguably, a better explainability method would combine the ability to analyse classes of images and support multi-concept interpretation for both model validation and exploration purposes without imposing high cognitive load to its users to make sense of interpretation outputs. With this in mind, we designed SECA, a human-in-the-loop SEmantic Concept extraction and Analysis framework that supports global analysis of machine behavior for multi-concept questions. SECA generates interpretation with a rich set of semantic concepts easily comprehensible by users. It fuses local interpretability methods to identify image patches that are relevant to the prediction for individual images, with human computation to annotate those patches with *semantic concepts*, i.e., visual entities with types and attributes. Using the entities, it then builds a model-agnostic structured representation of dataset images, on which statistical analysis techniques can be applied to answer both validation and exploratory interpretability questions. The combination of local interpretability methods, crowdsourcing, and statistical analysis techniques allows for scalable extraction and analysis of relevant concepts from a large number of images to facilitate validation and exploration of a model's behavior.

We demonstrate the *correctness*, *informativeness*, and *effectiveness* of SECA through several interpretability scenarios and evaluation protocols. To deal with the lack of ground truth of model behavior (a common issue in interpretability literature [233]), we design controlled experiments where several types of pre-defined model biases are induced, ranging from simple visual entities to complex ones related to image scene understanding. We further conduct empirical studies to understand the cost/effectiveness trade-off with varying number of images, granularity of annotations, and crowd involvement. In summary, we make the following key contributions:

- A novel human-in-the-loop explainability framework that allows for statistical analysis of global model behavior through rich multi-concept explainability questions.
- A benchmark for evaluating global explainability methods for multi-concept questions, including explainability scenarios across three image classification tasks with different types of biases.
- An extensive evaluation of the framework, demonstrating its effectiveness for both model validation and exploration, and analyzing its configurations for optimal cost/effectiveness trade-off.

A replication package containing code, datasets, and unabridged experimental results is available on the companion page[1].

8.2. RELATED WORK

We first provide an overview of existing explainability methods, then focus on approaches specific to image classification, and finally discuss works on human-in-the-loop machine learning.

[1] https://sites.google.com/view/webconf21-whatdoyoumean-balayn

8.2.1. MACHINE LEARNING EXPLAINABILITY

Existing explainability methods can be categorized in two ways: i) *local* vs. *global*, depending on the scope of data instances interpreted being individual instances or class of instances; or ii) *post-hoc* vs. *inherent* explainability methods, depending on whether the goal is to provide interpretations for an existing model or constructing self-explanatory models. Inherent explainability is achieved by adding explainability constraints in model learning to enforce feature sparsity [266], representation disentanglement [928], or sensitivity towards input features [809]. Another popular approach is attention mechanisms, which identify parts of the input that are attended by the model for specific predictions [894, 57]. Turning an existing model into an inherently interpretable model might be costly for users and might lead to a drop of model performance. In contrast, post-hoc explainability methods can be applied without model modification or retraining, and have therefore attracted growing attention. Our `SECA` is a post-hoc explainability method.

A key challenge in post-hoc explainability is interpretation *fidelity*, i.e., ensuring that the generated interpretation accurately describes model behavior. This can be achieved in several ways. Koh and Liang [447] propose a perturbation-based method that identifies training instances most responsible for a given prediction through influence functions, which estimate changes in model parameters as an effect of changes in the training instances. Gradient-based methods calculate the gradient of the output with respect to the input to derive the contribution of features [754, 688, 29]. Ribeiro et al. [677] fit a simpler model (with interpretable features) around the test instance to ensure local consistency between the interpretation and model prediction. A simple interpretable surrogate model can be learned to approximate the original model's predictions on a representative sample of the data [800]. Our approach is inspired from this last idea, as it generates interpretations using statistical tools such as association rule mining and decision trees (on human intelligible concepts) that are self-explanatory.

8

8.2.2. INTERPRETING IMAGE CLASSIFICATION

The most extensively studied explainability approach for image classification is *saliency*, a local explainability post-hoc method that highlights the most important pixels of an image for model decisions in what is called a saliency map [754]. "Importance" is defined as the sensitivity of decisions to the pixels with respect to a specific class. It is measured either by computing the gradient of the activation function for that class with respect to every image pixel [754, 732], or by passing the activated features of each layer of the model backwards into a reverse neural network model until the activations are mapped to the actual inputs of the model [55, 750]. Those approaches are likely to generate noisy results highlighting irrelevant pixels. To deal with that, methods such as SmoothGrad [769] and the Integrated Gradient [809] have been proposed.

Due to the intrinsic lack of semantics in pixels, global explainability is challenging in image classification. Kim et al. [437] introduce TCAV on top of their notion of Concept Activation Vectors (CAVs), which represents the translation from the internal states of a model to human-understandable concepts. The importance of a concept for model predictions is measured by calculating the directional derivative w.r.t. the corresponding CAV, i.e., the sensitivity of model predictions to changes in inputs towards the direction

of the concept. A main disadvantage of such an approach is that CAVs are obtained by training a linear classifier between a concept's examples and counterexamples; as a requirement, users need to provide sets of (50-150) example images for the training. Such a process is not only expensive, but sometimes also infeasible when the concept for testing comprises multiple concepts: users need to prepare a number of example images that each cleanly captures the multiple concepts that the user wants to verify. Moreover, the method is designed for model behavior validation; exploratory analysis is possible, but clearly expensive.

Ghorbani et al. [289] introduce ACE to automatically extract visual concepts, by aggregating related local image segments across the data. It relies on automatic image segmentation and clustering to obtain image patches potentially representing the same concept, and then uses TCAV to test for its importance. The quality of generated interpretations is highly dependent on the effectiveness of image segmentation and clustering: our experiment shows that ACE is prone to identify patches representing a concept related to low-level visual information (e.g., color), and that it fails at identifying patches of concepts comprising multiple concepts (Section 8.5.2 and 8.5.3). What is more, image patches generated by TCAV are not self-explanatory, and need to be analysed and interpreted by users.

By a combination of local explainability and crowdsourcing techniques, the SECA framework can address both issues of fidelity and cognitive load by 1) relying on human annotations to present semantic concepts at different conceptual granularities, and 2) by enabling multi-concept model validation and exploration.

8.2.3. HUMAN-IN-THE-LOOP MACHINE LEARNING

Human-in-the-loop machine learning [843] has been traditionally concerned with crowdsourced training data annotation [219] and crowd-collected samples [80]. A closely related line of work is "learning from crowds", where researchers study models that can learn from noisy crowd labels [672]. Unlike the conventional learning setting, these models are concerned with learning parameters of the annotation process (e.g., annotator expertise, task difficulty) and inferring true labels from noisy ones, possibly by incorporating (deep) active learning to reduce annotation efforts [897, 900].

Recent works address the use of human computation to debug machine learning systems. Nushi et al. [590] use crowdsourcing to identify weakest components of a machine learning pipeline and to propose targeted fixes. Yang et al. [899] introduce a human-in-the-loop system for debugging noisy training data using an automatic method for inferring true labels and crowdsourcing for manual correction of wrong labels. Hu et al. [377] introduce a crowdsourcing workflow for detecting sampling biases in image datasets.

The use of human intelligence for interpreting machine learning models has been limited to involving humans as users for evaluating the explainability methods, e.g., by observing if the interpretations help users choose a better model [677, 233]. Unlike those methods, SECA involves human computation as an integral component to identify relevant concepts, which is of crucial importance to make interpretations intelligible and to support multi-concept queries.

8.3. DESIGN PRINCIPLES AND CHOICES

We design SECA with the following key requirements in mind: (**1**) *Intelligibility*, the generated interpretation output should be comprehensible by its users; (**2**) *Effortlessness*, the cognitive load imposed on users should be minimal; (**3**) *Utility*, the framework should support both confirmatory or exploratory questions for model validation and exploration; (**4**) *Fidelity*, the generated interpretation should correctly and comprehensively describe the model behavior; (**5**) *Scalability* and *cost-effectiveness*, the framework should be scalable and effective under reasonable cost. In the following, we describe our design choices following from each of the above requirements.

8.3.1. INTELLIGIBILITY

To cater for *intelligibility*, we draw inspirations from the cognitive psychology literature on human reasoning and concept creation. Aerts [6] considers that *concepts* can be associated with observable properties, and the degree of association, called *typicality*, can be measured, by asking humans to rate it on a Likert scale. For instance, the concept `ambulance` can be associated with the property `cross sign`. Clearly, a property could be a concept itself, or be composed of multiple concepts [5]. The Representational Theory of Mind proposes a compositional semantic [534], where two or more "noun" concepts, or "noun" and "adjective" concepts can be combined using syntactic rules.

In this work, we consider *interpretability needs* aimed at analysing the degree of association (*typicality scores*) between *concepts* appearing in images (e.g. `cross sign`) and the classification *labels* –also concepts (e.g. `ambulance`)– that a machine learning model assigns to them. Those concepts correspond to *entity types* (nouns, e.g. `cross sign`) or *entity attributes* (adjectives, e.g. `red`) drawn from a vocabulary. *Interpretability needs* are expressed as *textual* queries over concepts, possibly using logical operations –conjunction (`AND`), disjunction (`OR`), and negation (`NOT`). An example of query (section 8.5) is: "`orange-stripe AND light AND NOT chassis`".

8

8.3.2. UTILITY AND EFFORTLESSNESS

We represent images and classification labels through the list of concepts, i.e. *entity types* and *attributes* they contain. Without loss of generality, we consider only classification labels related to a single concept (e.g. `male`/ `female`). We only consider a binary representation of a concept's relation to an image (presence/absence of a concept); a weighted representation (e.g. a value between 0 and 1) is an extension that we leave to future work. By explicitly identifying concepts on a per-image basis, we can apply a set of statistical analysis tools to identify the importance of concepts (individual or combined) across images in relevance to model predictions. This lessens the users cognitive load –many other global interpretation approaches rely on human user to identify relevant concepts across several images–, and allows to investigate more diverse model behavior.

8.3.3. FIDELITY AND SCALABILITY

To ensure interpretation fidelity, we use only *relevant* concepts. To do so, we rely on existing local interpretability methods: we compute the saliency maps for (a subset of) images on which a model makes predictions, and create semantic descriptions of the

entity types and attributes in the areas highlighted in the maps.

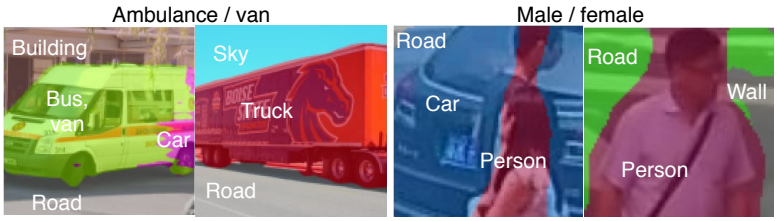


Figure 8.2: Automatic semantic segmentation with DeepLabv3. The `truck` and `ambulance` (left) appear as single segments while more specific entities like `stripes` and `flash light` are probably used by the model. The silhouettes of the individuals (right) form a single segment and the background another, whereas a model likely uses finer-grain entities (e.g. `hair length`, `face shape`).

This annotation process cannot currently be automated, as state-of-the art segmentation and object recognition methods are not accurate enough to uncover entities or attributes relevant to a model's decisions. In Figure 8.2 examples, the granularity of the segmented entities is large and the annotations vague. For instance, an `ambulance` is segmented as one entity and annotated as `bus`.

Hence, SECA adopts a crowdsourcing approach, where crowd annotators are asked to identify and describe with a textual annotation each entity in the salient image areas. Such approach can provide high fidelity and, while incurring some unavoidable costs, be scalable. Section 8.4 describes how SECA tackles obvious issues of annotation coherency across images. In the experiments of section 8.5 and section 8.6, we empirically study fidelity and cost-effectiveness, showing the quality and feasibility of the approach.

8.4. PROPOSITION: THE SECA FRAMEWORK

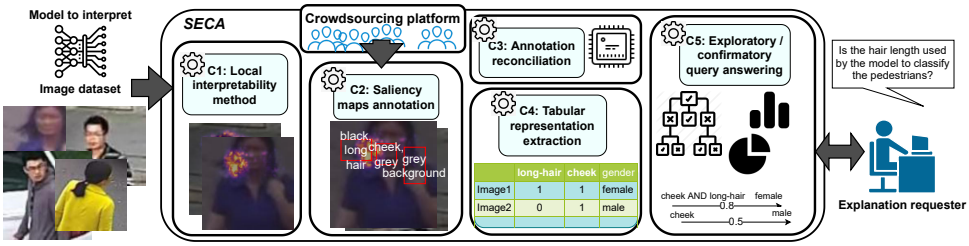


Figure 8.3: Overview of the SECA framework.

Figure 8.3 presents an overview of SECA (SEmantic Concept extraction and Analysis). Given as input (**1**) a trained image classification model and (**2**) a dataset, SECA can answer interpretability questions for validation and exploration purposes. (**C1**) Images in the dataset and their corresponding predicted labels are passed through a local interpretability method. The method generates saliency maps that indicate pixels relevant for the model prediction. (**C2**) All maps and corresponding images are sent to human annotators, to collect semantic annotations about the types and attributes of entities

represented by the salient pixels. (**C3**) Annotations across images are reconciled, and (**C4**) a structured and consolidated representation of all images is built. Finally, (**C5**) data analysis tools are applied, and single and multi-entity concepts and their *typicality scores* (degree of association of the concept and a target label) are outputted.

C1: Saliency Map Extraction. Saliency maps extraction is necessary to provide accurate interpretations while reducing annotation effort: clearly, annotating an entire image would be more expensive, and it could introduce concepts that are not germane to a model's behaviour interpretation. `SECA` is agnostic to the employed local interpretability method. We opted for SmoothGrad [769], which is sensitive to the parameters of a model (thus catering for more accurate capturing of a model behaviour) while minimising noisy results (i.e., highlighting irrelevant pixels). To further reduce annotation efforts, saliency map extraction is performed only on a random sample of all images. An appropriate setting of the number of sampled images depends on the complexity of the machine learning task, e.g., number and diversity of relevant concepts. We study the quality/cost trade-off related to this number in section 8.6.

C2: Saliency Maps Annotation. The annotation task combines two typical crowdsourcing activities: drawing bounding boxes and labelling (parts of) images. We ask workers to (1) identify, for each salient pixel area, the *entity types* corresponding to recognizable object shapes, and the *entity attributes* characterizing the area, e.g., its colors, textures or object property; (2) draw bounding boxes around the pixels corresponding to these types and attributes (we use bounding boxes instead of continuous curves as it is easier and faster for crowd workers); (3) provide a textual description (one word) of the identified types and attributes. For example, if the saliency map focuses on the blue cross image area on the trunk of an ambulance, the annotation would be *type*: `cross`; *attributes*: `blue`; for a gender classification task, a saliency map focusing on a person's short black hair results in *type*: `hair`; *attributes*: `black`, `short`. Entity-attribute information per salient image area is relatively easy to create by annotators, relevant to interpretation (as they are based on saliency maps of model predictions), and naturally intelligible for model developers and auditors. We ask annotators to provide fine-grained annotations, as fine-granularity entities can be later aggregated. Automatic checks are implemented to ensure that each image has at least one bounding box, and each bounding box has at least one entity type and attribute annotated. We employ multiple crowd workers per task to maximize the number and diversity of relevant annotated concepts. We retain concepts annotated by workers who spend more than a pre-defined amount of time on each image. The annotation task design is available on the companion page. Parameters of the *C2* component that affect the cost-effectiveness of `SECA` are *annotation granularity* and *annotator type* (e.g. experts vs crowds). We study their impact in section 8.6.

C3: Annotations Reconciliation. Annotation reconciliation is required as no predefined vocabulary of entity types and attributes is imposed on annotators, thus leading to diversity in vocabulary and/or granularity. First, we correct spelling mistakes with

spell-checkers [2], normalize the annotations by removing white spaces and converting all characters to lowercase, and rename synonyms or highly similar annotations using a reconciled term. The reconciled term is obtained by automatically clustering all the collected terms represented by word embeddings (pre-trained FastText embeddings), and picking the one closest to the centroid of each cluster. We use K-mean clustering, where k is chosen by identifying the value that leads to distributions of Silhouette score per cluster that do not exhibit negative values and that are as much uniform as possible across clusters. Features for the tabular representation are then built by mapping each annotation to one cluster or the association of multiple clusters. E.g., `wheel` is associated to the cluster containing this term, while `front light` is associated to the super cluster that combines the clusters of `front` and `light`. Annotation errors should not propagate as we later retain only interpretations that are statistically significant. In future work, we plan to look into (dynamically) controlling for vocabulary in the annotation task.

C4: Tabular Image Representation. The reconciled annotations of the salient areas of each image are stored in a de-normalised form. We create a binary-value column for entity type-attribute combinations (like `hair-short-black`), but also columns for each component (`hair`, and `short`, and `black`). For each image, we store which entity types and attributes pairs have been connected to any of their salient pixel areas. This denormalized storage helps with further statistical analysis and querying: for instance, a user could investigate three hypotheses: is the `cross` logo indicative of ambulances identified by a model predictions? Are `orange crosses` even more relevant? Has the model learned to check solely for the `color orange` (strongly correlated with ambulances)? The entity type `cross` can address the first question, the pair `cross-orange` the second, and the attribute `orange` the third.

C5: Query Answering. This component generates interpretations to fulfill both interpretation needs of model validation and exploration. The interpretations take the form of tuples corresponding to *a*) a concept, *b*) a prediction label, and *c*) a typicality score that measures the importance of a concept in predicting the label by the model. The tuples are then ranked based on the typicality scores.
Statistical tools. The most relevant concepts to include in output are identified through *statistical tests* assessing the correlation between each concept (i.e. column) present in the tabular representation and the predicted labels. We use the Chi-Square independence test [944], to check whether a concept and the label are independent. We retain concepts that are not independent significantly (p-value < 0.05). We compute the *Cramer's V* test [4] (a test commonly used in interpetability literature) on the retained concepts to obtain a *typicality score* that measures their degree of association with the labels. We also perform a *frequency analysis* of each concept per class, to identify concepts relevant for multiple classes simultaneously.

To facilitate *exploratory* needs, we pre-compute combination of concepts as follows: for each concept found significant, we add to the tabular representation a column with the complementary of the original column of the concept –this encodes the `NOT` operator

[2] SymSpell: https://github.com/wolfgarbe/symspell

of the concept, i.e. its absence. We also add columns that encode the logical AND combination of concepts (e.g. if wheel and light are found significant, we append a wheel AND light column). We then repeat the process of computing the statistical tests to identify the significant concepts among these new columns. Obviously, it is possible to explore all possible combinations of concepts; without loss of generality, in this paper we limit to pairwise combinations.

For model *validation purposes*, users can query over the concepts present in the tabular representation, possibly using logical operators. If not existing, the query is translated into a new column encoding the queried (multi-entity) concept. Statistical tests are then applied to establish the significance of the new column.

Rule extraction tools. The set of concept combinations is extended through rule extraction methods, uncovering multi-entity concepts that involve more than one AND or NOT logical combination. We employ *association rule mining* algorithms and *decision tree* classifiers. Association rules provide indications on the co-occurrence relationships between concepts within the rules. We apply the Apriori algorithm [9] on the original tabular representation, and constrain it to generate rules where the rule bodies are image concepts and the rule heads are the prediction labels. We use the *lift* score (a measure of the importance of a rule) as the typicality score of the rules. Unlike association rules that only captures co-occurrence relations, rules extracted from decision trees [128] contain numerical threshold for each concept. We use accuracy and frequency of the rule as its *typicality scores*. Decision trees require sufficient training data to be employed, so their applicability is conditional to the amount of considered images, but their output is richer.

8.5. EXPERIMENTAL SETUP & RESULTS: PERFORMANCE EVALUATION

We evaluate the interpretation performance of SECA by investigating two questions: *Q1: how correct are interpretations provided by SECA for uncovering biased behaviors?*, and *Q2: how informative are those interpretations in comparison to other interpretability methods?*

8.5.1. EXPERIMENTAL SET-UP

To date, no benchmark exists to measure the performance of interpretability methods for multi-concept questions. Inspired by previous evaluations [437], we design the following procedure.

EVALUATION PROCESS

Correctness. We consider interpretations *correct* if they highlight the concepts used by a model to make its predictions. Correctness is assessed by comparing these interpretations to a ground truth in controlled experiments. As such ground truth is not readily available, we generate it by biasing the models' behavior, i.e. we force models to "focus" on certain types of concepts that are exclusive of different classes. We create this bias either by injecting visual entities into images (e.g. adding time stamps to each image of a selected class), or by re-sampling the dataset based on existing entities (e.g. making

sure that all images of a class present an object from an angle different from images of other classes). We verify that the trained models learn these biases by computing the training accuracy: accuracy close to 1.0 indicates the models fit the data very well, probably thanks to the bias which is easy to pick up on. To further evaluate the correctness of SECA, we check its ability to highlight differences in "less obvious" (or less skewed) variations of model behaviors that are due to differently (less) biased composition of training datasets, or to the variations in the model architectures, under the assumption that these models should rely partly on different concepts to make their predictions. All these interpretability scenarios are summarized in Table 8.1.

Informativeness. Interpretations are informative if they uncover concepts that are *diverse* – presence of single and multi-entity concepts with various logical connections, and *actionable* for model debugging – concepts that show a potential issue and that are enough informative to act on them, e.g., by modifying the distributions of the corresponding visual entities in the training dataset.

EVALUATION DETAILS

Learning tasks. We select three classification tasks from two popular datasets for computer vision benchmarking: a *gender classification task* (T1) from pedestrian images using the PA-100K dataset [510][3]; a *three-class "fish" classification task* (T2) containing lobster, great white shark and tench images; a *two-class vehicle classification task* (T3) with moving van and ambulance images from the ImageNet ILSVRC-2012 dataset [699].[4] We crop and rescale the dataset images to input them to the machine learning models. We balance the data for equal representation of the classes (49000 images for T1, 4500 for T2, 3000 for T3).[5]

Machine learning models. We experiment with Inception V3 [813] (M1) and VGG16 [755] (M2), both pre-trained on ImageNet, and fine-tuned on the evaluation datasets. Those models were shown to learn different feature representations [934].

Bias injection in Data. Inspired by Yang and Kim [904], from the PA-100K we create 4 experimental datasets by injecting text as visual entities into the pedestrian task data: *Date dataset* (D1): date stamps on the `female` images and datetime stamps on the `male` ones – the model should rely on the presence or absence of the entity type `time stamp`; *Color dataset* (D2): `white` and `yellow` dates respectively on the `female` and `male` images –the model should rely on the `white` and/or `yellow` color attributes; *Date City dataset* (D3): `date`, or `datetime` and `city name` in the `female` images, `datetime`, or `date` and `city name` in the `male` images –the model should rely on combinations of entity types; *Colored-Date dataset* (D4): `white dates` or `yellow datetimes` in the `female` images, and `yellow dates` or `white datetimes` in the `male` images –the model should rely on pairs of color and entity types. In *Orientation dataset* (D5.2), we resample images of PA-100K (D5.1) by imposing a class-specific pedestrian `orientation` – all `male` images

[3] We acknowledge the limitations of a binary gender, but no other dataset was found.
[4] This task is inspired from [529] that hints at biases in background of these images.
[5] Our pre-processed dataset will be made available upon acceptance of the paper.

Table 8.1: Summary of the interpretation scenarios.

Task	Bias injection
T1: gender	D1-D4: text and color visual entities D5.1 / D5.2: original data / orientation bias
T2: fish	BM1.1 / BM1.2: original data / fine-tuned model
T3: vehicle	BM2.1 / BM2.2: original data / fine-tuned model
ML model	M1 / M2: Inception V3 / VGG16

have a `front orientation` (i.e. the pedestrian face is seen), and all `female` images a `back orientation`. Models trained on it should learn concepts characterizing the front and back of a person. These datasets should bias the model towards diverse concepts based on different entity types, attributes and their combinations, exactly what an interpretability method should uncover.

Bias injection in Model Architectures. We create different model behaviors to compare by using the pre-trained models to make predictions on the fish (BM1.1) and vehicle tasks (BM2.1), and by fine-tuning these models solely on the target classes of these tasks (i.e. training the models further only with the data of these classes) (BM1.2, BM2.2). Fine-tuning should bias the behaviors towards background concepts as these classes bear strong skew towards background entities (e.g. sharks are almost all in the ocean, tench with a fisherman next to a forest or grass, lobsters on a plate).

Baseline. We compare `SECA` interpretations to the only automatic interpretation approach in literature, ACE [289]. We do not consider TCAV [437] because it requires input "query" concepts. The study on the relationship between input patches and interpretation performance is beyond the scope of this paper. ACE outputs sets of 10 image patches, that should be interpreted by the user as single concepts. We retain ACE's sets that have a p-value under 0.05. It is generally difficult to associate meaningful semantic concepts to the sets, because their patches contain different entity types, thus making the underlying concept hard to identify. E.g., the underlying concept for image patches of `grey water`, `grey shark fin`, and `grey shark stomach` is ambiguous (could be the grey color and/or shark body parts)[6]. We retain *recognizable* visual concepts that are present at least in 5 of the 10 example patches of a set.

Annotation of Saliency Maps. To avoid confounding factors from crowd work ambiguity, in these experiments trained annotators (the authors) annotated the saliency maps, with agreement reached on the fine concept granularity. After experimenting on the learning tasks, we set $\sigma = 5$, $n = 10$ for SmoothGrad. For every task, the annotators annotated 300 images – as detailed in section 8.6, this amount is sufficient to cover concepts relevant to model behavior.

[6]The companion page reports highly ranked non-recognizable concepts from ACE.

Table 8.2: Example interpretations of SECA on the pedestrian classification task with simple injected biases.

Bias type	Output interpretations (rank - Cramer's value)
date (D1)	hour, NOT hour, minute, NOT minute (1-.93), hour AND minute(2-.9), day AND minute(4-.47), day(10-.24)
color (D2)	yellow-year(1-.96), yellow(2-.94), white(3-.83), yellow-day(4-.82), yellow-month(5-.81), white-year(6-.72)
date city (D3)	NOT city AND NOT minute(1-.5), NOT city AND NOT hour(2-.49), city AND NOT hour(3-.46), city AND hour(4-.45)
colored date (D4)	yellow-hour(1-.6), yellow-minute(1-.6), white-minute(2-.53), white-hour(3-.52), yellow-day AND yellow-year(4-.37)

8.5.2. RESULTS: CORRECTNESS

In the following tables, we report the simple and multi-entity concepts that appear at the top of the rank, from highest to lowest typicality scores, until 0.2 Cramer's value (threshold explained later). We denote *in italic* concepts identified by both SECA and ACE.

SANITY CHECKS

Table 8.2 provides an overview of the interpretations generated by SECA for the bias injection datasets D1-D4. The results show that SECA identifies all those biases we injected. For instance, for D1, concepts around hour and minute are correctly picked up by the statistical tests, the mined rules and the decision tree and associated to the female class, while the NOT operator provides the concepts corresponding to their absence in the male class. The AND operator and the pairs of types and attributes identify the correct combinations of concepts also in the colored date and date city cases. The output include few possibly irrelevant concepts, always having Cramer's value below 0.2. These concepts are either outliers, i.e. concepts that impact the model's behavior at a low frequency, or noise from the saliency maps (concepts that are spatially close to the main salient visual elements). For instance, the concept coat (not in the table, Cramer's value 0.19) is significant in D3, as it always appears next to the text elements, and it is present in 13% and 2% of the female and male images respectively.

CONCEPT CORRECTNESS

SECA also provides relevant concepts for the learning set-ups with biases induced by resampling (D5, BM1.2, BM2.2), as shown in Tables 8.3 and 8.4. For instance, for BM1.2, concepts matching the background bias are uncovered, e.g. water for the shark, grass and trees for the tench, and plate for the lobsters, while these concepts are not identified as relevant in BM1.1. For D5, identified concepts match with the orientation bias such as hair-related concepts for females, and face-related concepts for males (e.g. cheek, jaw, nose), while for the "unbiased" task, the concepts focus on the hairstyle. The NOT operator exposes even more the bias, since concepts that combine the hair and NOT an element of the face appear more typical than only the hair (e.g. hair AND NOT nose). When comparing the two machine learning models M1, M2, 7 out of the top 10 concepts are the same but with a different ranking, reflecting that the models learned similarly but still with differences. For example, the shark fins and tench heads are used by

Inception V3 and not VGG, which instead looked at the presence of a `shark head` with a higher typicality score.

The typicality scores are also relevant, as they are similar for concepts that appear with comparable frequency in the different classes. The scores evolve correctly when comparing models' behaviors: e.g., simple hair concepts have around 0.7 Cramer's value in the orientation bias data (D5.2) but are not even significant for the "unbiased" case (D5.1) since the model needs hair length.

Table 8.3: Interpretations outputted by SECA using statistical testing and by ACE on the different learning task set-ups. Concepts in *italic* are captured both by SECA and ACE.

Bias	Met.	Interpretations (rank - Cramer's or TCAV value)
Fish (T2)		
yes	SECA	tench_body(1-.9), lobster_claw(2-.83), blue-water, green, *beige*, water(6-.7), face AND tench_body(8-.67), face(10-.65), grass(14-.58), green-grass(14-.58), trees(19-.47), plate(25-.35)
	ACE	*white OR light-grey*(1-.99), *white OR beige*(2-.9)
no	SECA	lobster_claw(1-.9), tench_body(2-.86), shark_body(3-.82), *grey-shark_body*(4-.81), *orange*(5-.8), *orange-lobster_claw*(6-.79), shark_fin(7-.69), tench_fin(9-.67), water, *water AND shark_body*(12-.6), *yellow-green*(14-.57), *white-plate*(32-.31)
	ACE	*orange-lobster*, *grey-blue water OR shark_body*, *grey-shark*, *blue- water OR blue-shark_body OR grey-shark_body*, blue OR grey OR green back, *yellow OR grey*(1-1.0), grey shirt OR tench(2-.96), *white-dish*(3-.86)
Vehicle (T3)		
yes	SECA	light(1-.61), blue-light(3-.53), orange blue(4-.46), blue-light AND grey-car_side(5-.45), stripe-car_side AND orange-car_front(6-.43), cross, light AND cross(9-.39), *road*(10-.32), *chassis AND wheel, black-car under*(11-.28)
	ACE	*light_grey-car_side OR sky OR road*, *black-wheel OR back*, *grey-road OR car_side* OR car_inside(1-1), *letters*, *black-chassis*(2-.98), *dark-grey OR black-wheel*(3-.97), *white-back*(4-.91)
no	SECA	stripe(1-.5), windowAND stripe(2-.5), stripe AND car_side(3-.46), stripe AND mirror(4-.44), stripe AND tire(4-.44), *orange*, *orange-stripe*(5-.38), stripeAND chassis(6-.28), white(15-.2)
	ACE	*black-bumper*, *black-tire OR gray-tire*, *black*, *orange OR red*(1-1.0), *gray-window OR gray-bumper*(2-.99), *black-chassis*(3-.69), *black OR gray*(4-.18), *tire*(5-.15), *white-sky*(6-.05), *orange-letters OR red-letters*(7-.01)

CONCEPT COVERAGE

Compared to ACE as shown in Tables 8.3 and 8.4, SECA generally provides a more complete set of correct concepts, allowing for a more accurate understanding of a model's behavior. ACE identifies mainly concepts that models rely on to classify images from every class, thus not discriminative (e.g. `wheel` is used to identify both ambulances and vans); these are also identified by our frequency analysis. SECA also uncovers certain entity types present in single classes, that are missed by ACE (sometimes ACE outputs some color attributes that might relate to them). For instance, in D5.1, ACE outputs mostly colors that appear possibly in pair with entity types, e.g. `brown` color from hair or background for the `female` class, `white` color with a shirt or background for the `male` class, `gray` color for both classes. Our frequency analysis showed that these colors are

salient in both classes rather equivalently (e.g. `gray` appears in 59% of `female` and 68% of the `male` images, `gray background` in 22% and 30% respectively), meaning they are not the solely used concepts. ACE does not provide any additional insights, but SECA also uncovers concepts relevant for individual classes, primarily related to hair length and presence of `ear` and `neck` for the `male` class –entities often hidden under the hair in the `female` images.

8.5.3. RESULTS: INFORMATIVENESS

Table 8.4: Interpretations of SECA using statistical testing, rule mining and decision trees and of ACE on the gender classification task with and without orientation bias.

Cl.	Met.	Interpretations (ranges of typicality score)
Orientation bias (D5.2)		
F	Stat.	hair, black-hair(.7-.6), long, long-hair, black-hair AND long-hair(.6-.4), shirt AND hair, medium-hair(.4-.2)
	Rule	long AND grayAND black, long-hairANDblack-hair, long-hair, long(1.8-1.6), black-hair AND gray-back(1.4-1.1)
	Tree	long(.275), black, road, white, red(.06-.02)
	ACE	dark-gray hair OR shirt(1-.97), gray shirt OR back(.8-.6)
M	Stat.	neck(.7-.6), cheek, cheek AND neck(.6-.4), jaw, cheek AND jaw, face, neck AND jaw, nose, shirt AND cheek(.4-.2)
	Rule	hair AND neck, black AND short, black-hair AND short-hair, short (1.6-1.4), neck, hair AND ear, ear(1.4-1.1)
	Tree	car, neck, forehead, short, ear(.06-.02)
	ACE	gray, white OR gray shirt, gray sidewalk OR shirt(1-.97), light-brown skin(0.8-.6)
No injected bias (D5.1)		
F	Stat.	long, long-hair, longANDblack, long-hair AND black-hair(.6-.4), long-hair AND gray-back, gray-sidewalk-hair(.4-.2)
	ACE	gray-sidewalk, gray-back, brown-hair OR back(1-.97)
M	Stat.	short, short-hair, black-hair AND short-hair(.6-.4), short AND gray, neck, hair AND neck, short AND brown, ear(.4-.2)
	ACE	white-shirt OR back(1-.97), gray-sidewalk(.8-.6)

The results obtained on the "unbiased" set-ups (BM1.1, BM2.1, D5.1) in Tables 8.3, 8.4 show that we not only obtain correct concepts, but these concepts are also highly informative about a model's behavior, whereas concepts identified by ACE provide fewer and less actionable insights - the prevalence of color-related concepts over entity type-related concepts makes, arguably, dataset modification more difficult. Particularly, the interpretations provided by SECA are more clear and intelligible, more diverse, and more precise.

CONCEPT INTELLIGIBILITY

ACE mainly highlights color related concepts that we can only sometimes associate with entity type concepts. In contrast, our approach outputs more fine-grain concepts with

diverse entity types. This is probably due to technical limitations of the clustering algorithm used in ACE, that cannot precisely cluster entity types, but mostly color attributes. For instance, in BM1.2, ACE highlights `white`, `light gray` (probably coming from the plate, or from face or hand color), the `gray` color (shark skin or the background) for the shark, etc. These concepts are probably all correct, but are difficult to interpret since their provenance is not certain. Our approach on the contrary identifies the entity types that these attributes are associated to (e.g. green-grass, blue-water), thanks to the entity type-attribute pairs. Similarly, in D5.2 Table 8.4, ACE associates the `female` label to `dark` (hair or background) and `pale` colors (clothe or background), and `male` to `pale` and `gray` colors (clothe, background or faces). While it seems incorrect compared to our approach, extrapolating with our knowledge of the task, we see that they partly relate to face or hair concepts (i.e. the injected biases). Consequently, our interpretations are more actionable as concepts are traceable to visual entities in the dataset. Identifying pairs of entity types and attributes allows to uncover surprising and spurious biases, that are not clearly exhibited by ACE, but on which the dataset could be redistributed to mitigate the biases. For instance, in D5.1, `SECA` shows that the model primarily relies on the hairstyle, especially the stereotype of `long` / `short hair`, rather than pedestrian morphology. It also exhibits strong correlations between `hair` and `dark` colors, due to the low diversity of the dataset collected solely in Hong Kong.

CONCEPT DIVERSITY

The diversity in the nature of the concepts outputted by `SECA`, such as concept combinations and absence of concepts, allows to uncover richer behaviors than with ACE in Table 8.3. For instance, in BM2.1, `SECA` shows that a) the co-occurrence of a vehicle `side view` and a colored `stripe` indicates an `ambulance`, but the co-occurrence of this view and a `chassis` indicates a `van` according to the statistical tests; b) the co-occurrence of a `white` vehicle `side`, a `black tire` and an `orange stripe` indicates an `ambulance` according to the mined rules; c) not having `stripes` and `flashing light` or having `stripes` and no `light` are associated with the `van` respectively with 0.47 and 0.44 Cramer's value (`stripes` are often indicative of `ambulances`), using `AND` and `NOT` operators. ACE misses these correlations that require the identification of absence concepts and the ability to calculate the significance of multiple concepts simultaneously – it would require image patches with multiple concepts represented next to each other, like a `tire` and a `flashing light`.

INTERPRETATION RICHNESS

The exploration tools of `SECA` allow to explore various, precise model behaviors that other approaches do not uncover, and that might not be straightforward to query.

While the frequency-based analysis and the statistical tests identify simpler significant concepts (in validation, they allow the user to query combinations of concepts however), association rule mining uncovers more complex combinations, e.g. Table 8.4 "`long AND gray AND black`" has the highest typicality. Simply by varying the configuration of the rule mining algorithm, it is possible to focus on diverse interpretation goals, such as finding frequent concepts by filtering out concepts with low support, or finding complex concepts that are less frequent by lowering such threshold. E.g., in BM1.1, the

rule `tench head AND tench body AND tench fin` has a top lift score but is fairly rare in the data, hence it is outputted only with a support threshold under 0.2.

Decision trees discover complex behavior rules, and the information attached to them tell how common they are. For instance, in D5.1, the tree shows that `NOT long AND NOT ear AND NOT background AND NOT black AND NOT road` classifies `males` with 96% accuracy for 25 out of 300 records – which matches the intuitions about the data obtained from the statistical tests. Concepts appearing in the higher parts of the trees are accurately distinctive of the two classes (e.g. `long hair` is the first identified concept). Concepts in lower level do not correspond to expectations for the unbiased tasks: background elements appear as salient as parts of the body such as the `ear` or `neck` that we found are more important using the other methods. Because there are many visual elements but few rows in our tabular data (e.g. 78 elements for the pedestrian scenario and 300 records), the tree overfits to the data – curse of dimensionality– as confirmed by the low importance scores. Hence, only rules with high accuracy should be extracted from the branches, accounting for their frequency, and only the first levels of the tree should be used to extract individual concepts when few data are available.

8.5.4. DISCUSSION

Results show that SECA correctly identifies different types of biases in model behavior –biases of visual entities, those arising from skewed data distribution and those from model architecture– and that it generates a rich set of interpretations for exploratory analysis of model behavior. Compared to ACE, SECA identifies a larger and more diverse set of concepts that are useful to identify more (biased) behavior patterns of a model. In particular, SECA identifies concepts with entity types and those comprising multiple sub-concepts that are often missed by ACE. We also observe that the different analysis tools of SECA allow to uncover various model behaviors.

A clear experimental limitation is the lack of an exact ground truth for what a model learns, making it challenging to conduct a full evaluation (especially in terms of interpretation completeness). We cope with this issue by setting up controlled experiments with manually induced biases of various types, which allow to evaluate interpretation effectiveness and informativeness from the bias angle. Another area of improvement concerns the amount and diversity of learning tasks and datasets. However, we stress that to date ours is one of the most comprehensive interpretability evaluation effort.

8.6. EXPERIMENTAL SETUP & RESULTS: COST PERFORMANCE TRADE-OFF

In this section, we investigate *Q3: how do the main parameters that configure SECA impact the trade-offs between cost, correctness, and informativeness of the interpretations?*

8.6.1. EXPERIMENTAL SET-UP

EVALUATION PROCESS

We study the impact that number of annotated images, annotation granularity, and the type of annotators (i.e., crowd-workers vs. trained annotators) have on the correctness and informativeness of the explanations generated by SECA. We use the same tasks as in

the previous section.

Number of annotated images. As a reference, we use SECA to create interpretations based on a high number of annotated images (400). As we have shown above, SECA can generate satisfactory quality interpretations, i.e., interpretations that match the reference ones. We incrementally create interpretations from lower numbers of annotated images (between 20 and 400, in increments of 10). Finally, we compute the precision and recall of the concepts and the mean absolute error of Cramer's values, comparing the interpretations using smaller labeled image sets to the reference with 400 labeled images.

We hypothesize that the complexity of a learning task, which depends on a dataset characteristics, impacts the number of images needed to obtain similar correctness. The more classes to learn (need to uncover behaviors for more classes), the more diverse the visual entities and attributes per class (forces the model to use more concepts for classification), and the more concepts co-occur across classes (a model might rely on complex combinations of concepts), the more images should be needed to uncover a model's behavior. We investigate this by comparing the metrics computed on biased and unbiased scenarios (variation of intra-class semantic content diversity), and across tasks (more classes and lower inter-class concept co-occurrence in the fish task T2 than in T1 and T3).

Annotation granularity. We vary granularity from large to fine grained for both entity types and the attributes, defining different categories: for the entity type granularity category *E1*, all visual entities inherently part of the class (e.g. a `blue star` for the `ambulance` class, an `antenna` for the `lobster` class) are annotated as the class name, and all background objects are annotated as "background". In category *E2*, we distinguish the different parts of classes (e.g. `claw`, `antennas`, `legs`, `body`, `head` for the `lobster`), and we categorize background elements into large-grain categories (e.g. `nature`, `food`). Finally, in category *E3*, we refine the background annotations (e.g. `rice`, `tomato`, `pavement`) and the non-background ones when finer-grain entities can be identified. For the attributes, the category *A1a* combines color variations into seven main colors, and textures into large categories; in category *A1b* colors are combined depending on dark or light aspects. In category *A2* no combination is performed. We consider the reference granularity being the finest-grain ones, i.e. *E3* and *A2* and compare the resulting interpretations with coarser granularity categories.

Annotators. We compare the interpretations originating from saliency maps annotated by trained annotators (the authors) with saliency maps annotated by crowd workers, also computing the precision, recall, and mean absolute error.

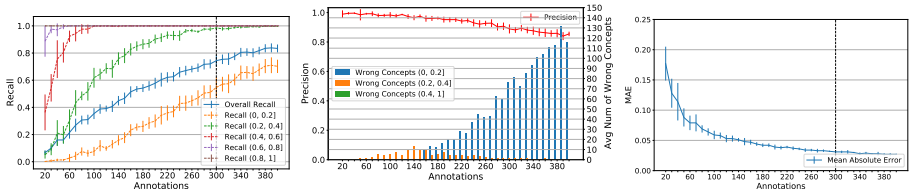
EVALUATION DETAILS

Experiments on number of annotated images. For the three learning tasks, we annotate 800 images, sample 400 images to form the reference interpretations, and sets of k images among the 400 remaining ones to form the interpretations to compare. We repeat this process 10 times to obtain statistically significant measures. We hypothesize that the

precision and recall will be low for concepts with low Cramer's value. To verify this, we divide the reference concepts into 5 batches with Cramer's values equally divided between 0 and 1 (i.e., between 0 and 0.2, 0.2 and 0.4, etc.), and compute the recall per batch with all the concepts to compare with. We cannot do this for the precision as we cannot directly compare the reference batches to comparison concepts – small errors in Cramer's values would make the measures wrong (e.g. a comparison concept of Cramer's value 0.61 would lower precision if its reference concept is in the batch 0.4 – 0.6). Instead, we simply count the number of wrongly retrieved concepts in the comparison set. We also compute the mean absolute error per batch as we hypothesize that low Cramer's value concepts are attributed less accurate values due to the sampling error.

Experiments on annotators. In this experiment, we compare annotations of trained annotators to untrained crowd workers recruited on crowdsourcing platforms, focusing on general annotation properties (like amount of bounding boxes, coverage of the salient areas, amount of time spent, feedback questionnaires, etc.), and we investigate how the provided concepts compare semantically. For this semantic comparison, we automatically map concepts provided by crowdworkers to those provided by the trained annotators by computing a similarity score between the concepts word embeddings, fixing a threshold T and retaining as matching only the concepts with similarity above T. We repeat this with the different annotation granularities. Assuming that the authors' annotations are indeed of high quality, we can now investigate the precision and recall of the crowd compared to the authors. Furthermore, we investigate the effect of annotation reconciliation (step C3 of our approach) which is necessary when multiple crowdworkers provide annotations with varying vocabulary.

Crowdsourcing component implementation. We deployed the annotation task on Amazon Mechanical Turk. Each HIT was composed of a set of ten images and their saliency maps, and was assigned to three crowd workers[7]. The instructions encourage the workers to search for domain knowledge to give precise annotations as a pilot study showed diverse annotation precision. $k = 125$ clusters are used for the reconciliation component as it provides the best Silhouette scores.



(a) Recall.

(b) Precision and number of incorrect concepts.

(c) Mean absolute error.

Figure 8.4: Analysis of the number of image annotations required by SECA for the ImageNet Fish task. The values in brackets correspond to the Cramer's values of the reference concepts used to compute the corresponding curves.

[7] We included workers from UK and USA with at least 5K approved hits, and a HIT approval rate greater than 85%.

8

8.6.2. RESULTS: NUMBER OF IMAGES

Figure 8.4 shows an example of the curves obtained for the fish task BM1.1 (results for other datasets are similar, and reported in the companion page). We observe that 300 annotations provide satisfactory concept sets and Cramer's values, and only 200 annotations are needed if we do not need to identify less significant concepts. We do not observe significant differences across tasks and biases.

Recall. For all the learning tasks, concepts are retrieved with only 200-300 annotations. Although the overall recall might not seem satisfying even for 400 annotations, the recall for all concepts with Cramer's value greater than 0.2, closely approaches 1 (and 0 standard deviation) with 300 annotations, and a minimum of 0.9 recall is observed with 200 annotations. Concepts of Cramer's value between 0.4 and 1 are even retrieved with just 100 annotations. Lower Cramer's values are indicative of less significant, possibly irrelevant concepts (see subsection 8.5.2), picked up by a model in lower frequencies, thus they are more susceptible to sampling noise, and need more images to be retrieved.

Recall curves are similar across tasks. For instance, BM1.2 also needs 300 images but with a standard deviation lower than BM1.1, probably because of its lower intra-class complexity. D5.1 (pedestrian) just requires 20 more images to approximate a recall of 1 with a standard deviation lower than 0.02 –probably due to more inter-class co-occurrences than BM1.1. Generally, this is because the impacts of the data characteristics balance each other, e.g. although there are more classes in T2 (fish), the image content in T3 (vehicle) or T1 (pedestrian) is more diverse.

Precision. The precision is also satisfying with only 200 images. The precision curve decreases from 1 with 10 images to 0.93 with 200 images and 0.9 with 300 images, the standard deviation remains constant at 0.04. A closer look at Figure 8.4b shows that, once more, most incorrect concepts have Cramer's values inferior to 0.2 when increasing the number of images since such concepts are more subject to sampling noise. Not accounting for these concepts allows to keep a precision higher than 0.9 for every number of images.

The curves are similar across tasks, with T1 and T2 having a larger standard deviation around 0.1 and 0.07 respectively, verifying our hypotheses. Only tasks with many more classes and higher visual entity intra-class diversity or inter-class co-occurence would probably require to annotate more images.

Mean absolute error. The approximation of Cramer's values is accurate even for less than 200 annotations (again except for concepts of Cramer's values below 0.2). The error decreases rapidly with more images, going from 0.2 with 0.1 standard deviation for 20 annotations, to 0.026 and 0.001 standard deviation for 300 images and above. This is because having more annotations allows to approach the real joint distribution of concepts and classes in the data, on which the Cramer's values are computed.

8.6.3. RESULTS: GRANULARITY OF THE ANNOTATIONS

Entity types. We report the results on BM1.2 in Table 8.5 (results from other tasks point to similar conclusions, and are reported in the companion page). With large grain anno-

Table 8.5: SECA interpretations on the fish bias task for various **Gra**nularity of entity types. Granularity *E3* is in Table 8.4.

Gra.	Interpretations (Cramer's value)
E1	lobster(.95), tench(.92), shark(.83), back AND lobster(.89), tench AND back(.88), orange(.81), grey-tench(.83), orange-lobster(.79), green-back(.78), light grey-back(.74)
E2	tench_body(.89), lobster_claw(.85), lobster_body(.73), orange-lobster_claw(.72), blue-water(.75), water(.7), beige-human_body_part(.63), food(.46), table_tool, clothe(.4)

tations (*E1* and *A1a*), the retrieved concepts are correct but poorly informative as actionable insights. E.g., `lobster`, `tench` and `shark` are the most salient concepts, followed by color concepts, combinations of the background concept and one of the previous fish-related concepts, or pairs of color and fish concepts (e.g. `orange-lobster`). This interestingly indicates that the model uses both concepts related to the classes and background concepts, but without more details we can neither conclude about the validity of this behavior – certain background concepts could make sense, e.g. shark in the water, nor identify visual background entities to redistribute in order to remedy to the potential background bias.

Finer grain annotations bring more precise debugging information. For instance, *E2* uncovers the different parts of the concept classes (e.g. `lobster claw`) possibly in combinations with colors (e.g. `orange-lobster claw`), and the background entities (e.g. `blue-water`, `beige-human body part`) used by the model and based on which a dataset can be transformed to mitigate biases. Further detailing background concepts *E3* provides even more detailed information, e.g. the `face` is the human body part the most associated with the `tench`, the `shirt` is the most associated cloth.

Hence, depending on the interpretation need, the granularity of annotations needed differs. The medium granularity is enough to explore the general functioning and validity of a model, while the finest-granularity provides precise information to mitigate behavior biases. If the finest granularity is employed, we recommend to obtain a higher level overview of the model's behavior by querying combinations of concepts with an `OR` logic connection –equivalent to aggregating concepts into larger grain ones. For instance, the medium granularity uncovers shark-related concepts with Cramer's values around 0.65 or lower, while aggregated altogether the value increases to 0.83, above the background concepts (0.74), showing the potential correctness of the model's behavior.

Entity attributes. The granularity of attributes on the contrary do not lead to differences that impact the interpretations of the models. This is probably due to the limited range of distinct colors that a human is able to annotate easily. Automatic annotation methods using pixel values might bring additional insights on the color shades that are the most important for classification.

8.6.4. RESULTS: CROWD VS. TRAINED ANNOTATORS

COMPONENTS' QUALITY.

1) *Annotations.* Crowd workers took $\mu = 28m, \sigma = 11$ minutes to execute the task. Quality of annotation was good. Most workers who took less than 15 minutes provided 1-2 an-

notations of simple salient areas per images, while the ones who took more time provide 2.8 annotations per image in average, with a maximum of 66 per HIT. The difficulty of identifying salient areas, drawing bounding boxes and annotating entity types and attributes was evaluated with an average of 3.3, 3.1, 3.1 and 3.3 respectively on a scale of 1 (easy) to 5 (difficult). Few annotators provide full coverage of the salient areas, either due to not identifying certain entities, or due to not drawing boxes around the entire areas. This has limited impact on interpretation quality, as having precise bounding boxes is not important, and using multiple annotators proved to provide the needed coverage.

2) *Annotation Reconciliation.* The clustering approach used to determine reconciled annotations is satisfactory: most clusters are relevant for the interpretation task. They reconcile wording differences (e.g. `tooth` and `teeth`), synonyms and terms that designate similar concepts (e.g. `belly`, `stomach`). Mistakes are introduced by words with multiple meanings, e.g. lobster `antenna` is grouped with network infrastructure words, because no context is used to create the embeddings. Some terms that relate to different granularities are grouped (e.g. `hand`, `fingers` and `thumb`), which might impact the interpretations when the finest granularity is needed.

CORRECTNESS OF THE INTERPRETATIONS.

We report the results for the fish bias task. The interpretations from the crowd uncover the main expected biases, e.g. presence of `water` for the `shark` images, `grass`, `trees` and `human body parts` around the `tench` images, `plates` for the `lobster` images, and only a few concepts do not appear, e.g. certain food concepts such as `corn` for the lobster. However, we obtain only 0.48 precision, 0.61 recall and 0.18 mean absolute errors of Cramer's value on significant concepts retrieved for the finest granularity. The medium and large granularity respectively reach a precision of 0.53, 0.57, a recall of 0.70, 1.0 and a mean absolute error of 0.19, 0.40. Only a few concepts are not mentioned by the crowd (e.g. `lemon`), probably because they appear small in the background of the images, behind the main objects. As hinted by these increasing values, this contradiction is mainly due to measurement errors: differences in the vocabulary and granularity of annotations cause errors in the mapping used in the evaluation process, which makes precision and recall low. Most reference concepts that appear as missing from the crowd interpretations are actually retrieved. For instance, the concepts from the trained annotators `shellfish` and `sauce` are annotated by the crowd with `oyster`, `shrimp` and `soup`, `liquid`. The crowd annotations are often more fine-grained, which also lowers the precision. For instance, heads annotated with `boy head`, `man head`, `woman head` and some with `human head` instead of solely the latter like the trained annotators', formed two distinct clusters (`human` associated with `animal` and the others together), one appearing irrelevant. The average mean absolute error increases with larger granularity because we modify only the granularity of the trained annotators' concepts, while the worker's concepts remain distinct with lower Cramer's values. Overall, employing the crowd with simple post-processing methods provides interpretations of similar correctness, with only few fine-grain concepts missing.

INFORMATIVENESS OF THE INTERPRETATIONS.

Certain interpretations obtained from the crowd are richer in terms of granularity than those from the trained annotators. For instance, the crowd interpretations differentiate

between the shark fins, e.g. caudal fin, dorsal fin, whereas only fin appears in the trained annotators' concepts. This is because certain workers provide precise vocabulary (as encouraged in the instructions) that a trained annotator might not have thought of (e.g. pectoral, caudal, dorsal fins, etc.), or for which a trained annotator does not have domain knowledge like the species of fish labeled by the crowd (e.g. muskellunge, carp, tench). This is the main advantage of using the crowd instead of trained annotators. Having multiple, lower cost, annotators allows to mitigate individual bias, as different persons focus on different entities, granularity and labels.

8.6.5. DISCUSSION

SECA can produce correct and informative interpretations already with few images annotated (300) using crowd workers. Significant concepts are well covered with even fewer images (100, Cramer's value above 0.4), with satisfactory performance. While finest-grain concepts are useful to understand precise model behavior and debug it, medium-grain concepts seem to be satisfying for model validation and general exploration purposes. Crowd annotations generally align with those from trained annotators, but with a richer vocabulary that allows to gain comprehensive understanding of model behavior. While workers' contributions are not always accurate, we stress the simplicity of our task design. Experiments show that crowd workers can be systematically employed to support saliency map annotations, thus enabling an accurate, scalable, and relatively cheap post-hoc interpretability method. We acknowledge though, that our experiment is limited to binary/three classes problems. Experiments on tasks with more classes can help quantify the impact of the class number and diversity on cost effectiveness trade-off.

8.7. CONCLUSION

We presented SECA, a framework to support post-hoc, interactive explanation of machine learning models for image classification. SECA offers explanations based on easily understandable semantic concepts (entities and attributes). These concepts are obtained via crowd-sourcing from local explainability saliency maps, and then reconciled and consolidated into a unified and structured representation which allows the use of different statistical mining techniques to discover or query for concepts relevant for a model's decision making. Extensive experiments showed that, compared to related work, SECA can discover more informative and complete concepts, and that these concepts are more interpretable and actionable to debug a model. Results show that using crowd workers to provide semantics to annotate salient image areas provides results with sufficient performance at lower costs, and that also smaller sample of annotated images lead to actionable results. While we now know that SECA allows to collect high-fidelity explanations of model learned mechanisms, it remains unclear to what extent these explanations would be useful to machine learning developers. This is what we investigate later in Part III Chapter 10. Beforehand, in Chapter 9, we develop a technical solution to collect the expected mechanisms for a machine learning model, that developers could potentially use together with the SECA's explanations of the model learned mechanisms when diagnosing their models.

9

OBTAINING EXPECTED MECHANISMS

9.1. INTRODUCTION

Part II showed us the need for machine learning developers to have domain knowledge about the domain of application of their machine learning system, to be able to diagnose its failures and bugs. Knowledge can be used as expected mechanisms to assess the validity of the "knowledge patterns" acquired by machine learning models (i.e., its learned mechanisms) and highlighted by recent explainability works [717, 715] for various inference tasks [489, 419]. Our interviews with developers also showed the challenges for them to obtain such domain knowledge. Hence, in this chapter, we investigate how to collect domain knowledge efficiently, with the aim of using such knowledge in future to diagnose machine learning models.

Knowledge engineering is the area of research that focuses on developing methods to gather knowledge [756]. Knowledge is gathered by interrogating humans through simple interfaces or complex interactions such as games with a purpose, by mining existing textual resources, or by logically reasoning about known facts to infer new ones [917, 364]. In light of the renewed need for knowledge, we have identified three important gaps pertaining to these knowledge elicitation methods, that we address in this work.

Our understanding of the *type of knowledge* that can be gathered through these methods remains shallow. Knowledge can be categorized using different typologies of qualities depending on the domain and its envisioned use. It varies from explicit to tacit, from general to specific, from conceptual to situational, from shallow to deep, from commonsense to expertise, etc. Yet, previous works have not provided an in-depth characterization of the knowledge they collected. This might be a barrier to leveraging such knowledge in the context of AI tasks. For example, consider the question "*What does one gain from getting a divorce?*", and the choices –bankruptcy, sadness, depression, tears and freedom. While the first four seem highly relevant to "divorce", the mention of "gain" indicates positivity, hence "freedom" is the right answer. Here, it is important to asso-

ciate "gain" with something positive, which humans are capable of doing tacitly. Tacit and commonsense knowledge –"knowledge about the everyday world that is possessed by all people"[507], that has the qualities of being shared by multiple persons, and of being fundamental, implicit, large-scale, open-domain [917]– has been heralded as a pivotal ingredient for future AI systems [532].

Gathered knowledge remains limited and *incomplete* [451], leading to errors in certain tasks. Elicitation methods largely facilitate the creation of *generative* knowledge, but neither *discriminative*, nor *negative* knowledge –despite the fact that novel AI tasks require such knowledge, *e.g.,* for discarding erroneous AI models [451, 39, 38]. Discriminative knowledge allows to distinguish between two concepts (*e.g., octopus*, contrary to *fish, do not have fins*) — as opposed to generative knowledge that qualifies a single concept. Negative knowledge informs on the invalidity of a tuple to characterize a concept or two compared concepts (*e.g., man* is **not** a *profession*). Leveraging human intelligence and commonsense knowledge can allow to collect targeted knowledge beyond what is found in existing resources. However, owing to a lack of understanding of types of knowledge that can be elicited from humans (or online crowd workers), and the concomitant breadth of knowledge, typical knowledge acquisition methods are not readily configurable to meet varying requirements (*e.g.,* knowledge tacitness, specificity).

We position our work in the context of knowledge elicitation techniques involving the crowd [851, 917, 364]. Herein, we draw inspiration from prior work in the realms of games with a purpose (GWAPs), which have shown promise in collecting diverse knowledge in an efficient manner. Popular GWAPs, such as the ESP game [852], Peekaboom [855], and Phetch [853] have provided evidence to show the efficiency of this approach, and its flexibility (*e.g.,* use of gamification and mechanics such as taboo words to tune the type of collected data). Combined with the development of crowd computing frameworks [271], GWAPs can allow for large-scale acquisition of knowledge while engaging humans using different incentives. To the best of our knowledge, however, no GWAP has been developed or proposed to gather discriminative or negative knowledge. Hence, we first design and implement a novel GWAP called '`FindItOut`', to elicit plural knowledge from players. We then characterize the diversity of knowledge that can be collected using `FindItOut`, and the utility of such knowledge in relevant AI tasks. We highlight the suitability of `FindItOut` in encouraging players to combine explicit knowledge and externalize relevant tacit knowledge. Finally, we demonstrate the efficiency of the game subject to different parameters. We make the following contributions:

- A novel configurable GWAP[1] that facilitates the collection of positive and negative, generative and discriminative knowledge, while ensuring an enjoyable player experience.
- A structured set of dimensions through which one can characterize knowledge collected through user interactions.
- A characterization of the types and quality of knowledge that can result from using `FindItOut` and paid online crowdsourcing.

[1] https://github.com/delftcrowd/FindItOut

- An extensive evaluation of the throughput and utility of the game for two machine learning tasks.

Our results demonstrate that `FindItOut` is highly efficient in obtaining tacit, discriminative and negative knowledge — absent from existing knowledge bases. We also show that the configurability of the game allows to elicit knowledge that can be particularly useful for AI tasks like commonsense question answering and identification of discriminative attributes.

9.2. RELATED WORK

9.2.1. KNOWLEDGE AS A TOPIC OF ENQUIRY

In the Social Sciences. Different typologies of knowledge have emerged [644]. One of the most common ones considers explicitness. Explicit knowledge "can be articulated into formal language [.. and] can also be readily transmitted to others."[187]. Conversely, tacit knowledge is hard to articulate. It "consists of informal, hard-to-pin-down skills, [..] mental models, beliefs, and perspectives so ingrained that we take them for granted and cannot easily articulate them" [585].

There is a higher chance that explicit knowledge already resides in available knowledge bases, as opposed to tacit knowledge [388]. The game we propose involves human players and pushes them to formulate statements about concepts they might not immediately think of. We therefore hypothesise (and evaluate) that our game allows to collect tacit knowledge in addition to the explicit kind.

The distinction between tacit and explicit knowledge has primarily been used to formalise the process of knowledge creation in organizations [585]. Particularly, *combination* [585] is the process of synthesizing explicit knowledge from the combination of previous explicit knowledge. Our game realizes this by synthesizing explicit knowledge about diverse concepts into a single knowledge repository. *Externalization* [585] is the process of creating explicit knowledge from tacit knowledge, often using interviews and questionnaire with experts, or expert's self-analysis [577]. In our work, we evaluate the extent to which our GWAP, `FindItOut`, can support and operationalize externalization through the game mechanics.

In Computer Science. Recent machine learning inference tasks describe *discriminative knowledge* in contrast to *generative knowledge*. While generative knowledge broadly corresponds to information about different entities, discriminative knowledge allows to identify differences between these entities, which "allow to grasp subtle aspects of meaning [.. and] contribute to the progress in computational modeling of meaning" [451]. Recent works [39, 38] on knowledge inference under the open-world assumption also discuss the importance of *negative knowledge*. It may enhance knowledge bases for knowledge exploration and question answering. Biswas et. al [111] also propose to leverage negative statements as clues to help players find answers to specific questions. Concomitant with the growing interest in these types of knowledge, `FindItOut` is the first GWAP that directly collects discriminative and negative knowledge, which can always be turned into generative one via simple post-processing.

9.2.2. GWAPS FOR KNOWLEDGE ELICITATION

Games with a purpose (GWAP) are used to collect large quantities of knowledge efficiently from the crowd [851]. They have been shown to perform well to collect certain types of knowledge.

Multiplayer GWAPs. Verbosity [854] was the first GWAP proposed for collecting commonsense knowledge. It is a two-player, Taboo-inspired, collaborative game, where a narrator player gives hints to a guesser player who should guess the word the narrator is hinting at. It uses a scoring system to incentivize players to provide the most relevant inputs. A single-player version also exists in order to validate the collected knowledge. The hints have a template format with a relation to fill in with additional words. Common Consensus [502] is a competitive game inspired from FamilyFeud, that collects goal-specific knowledge. It generates questions based on a list of goals and a list of template-questions, and players enter as many possible answers (single words) as possible. Scores are computed based on the number of players with the same answers.

Single-player GWAPs. RobotTrainer [683] is a game, that collects knowledge rules, ranks their appropriateness, and evaluates their validity. For this, it is organized in three levels, where players get to write template-based rules that should serve to answer a question about a given short story, or evaluate these rules. It is shown to provide similar results to non-game based interactions, but with more engagement of the users. The 20 Questions game [781] requires the player to think about a concept, and the game sequentially generates a list of 20 relation-template based questions to try guessing the concept, questions that the player should answer truthfully. Despite a simple design, players were found to enjoy this game more than a simple template-based input system. The Concept Game [354] similarly generates rules that a player is asked to verify, in order to reduce the cognitive load of players generating assertions. Other games have been proposed such as Virtual Pet, Rapport, Guess What?!, OntoProto, SpotTheLink [756], that ask players to agree on the relation between concepts, to guess concepts described by other concepts, or to answer questions to extract knowledge.

In comparison to existing GWAPs: (a) `FindItOut` by design, has a higher throughput than previous games. It operationalizes the idea of making both questions and answers relevant to the creation of knowledge. This leads to collect more knowledge in comparison to the aforementioned two-player games, since the two players contribute distinct tuples of knowledge simultaneously, contrary to the other games where players interactions allow for the creation of a single knowledge tuple. (b) `FindItOut` is the only game that directly allows to collect discriminative and negative knowledge. Previous games require either to directly input concepts in relation to a pre-existing characteristic, or to fill in template. They do not leave the space for negative inputs, which also removes the opportunity to indirectly elicit discriminative knowledge. (c) The knowledge that `FindItOut` elicits is, by design, more diverse. While it re-uses the previous ideas of relation templates to fill in, and of scoring systems, it varies from 20 Questions and Common Consensus in that the knowledge it creates is more varied since the rules within the templates are human-generated, and richer than single words (association of relation and up to 5 words).

9.2.3. ELICITATION THROUGH CROWD INTERACTIONS

Besides GWAPs, other interactive methods [917] exist for knowledge elicitation. A fundamental feature of `FindItOut` is its question answering workflow, which is inspired from the offline game *Guess Who?*, and from crowdsourcing frameworks such as CuriousCat [126], that collects contextual commonsense knowledge, by asking questions to crowd workers that refer to their current environment (*e.g.,* size of a restaurant they are present in). Cosmos QA [379] and Socialiqa [722] are datasets collected by asking crowd workers to formulate questions and answers that require commonsense knowledge, in relation to textual descriptions of everyday situations taken from blogs or prior knowledge bases (*e.g.,* ATOMIC). We draw inspiration from these works and incentivize crowd workers to formulate questions through the game mechanics.

9.3. PROPOSITION: DIVERSE KNOWLEDGE EXTRACTION

To elicit and collect discriminative and generative knowledge, that is both positive and negative, we propose `FindItOut` [63] — a competitive 2-player game inspired by the popular game "Guess Who?'. The functional and non-functional requirements that governed the design of the game are elucidated in the companion page [2].

9.3.1. KNOWLEDGE ELICITATION

In line with existing knowledge bases, we aim to collect knowledge in the form of relations between concepts.

Generative knowledge. A triple of generative knowledge that we collect corresponds to a concept, a relation and a characterizing input, and takes two possible formats. It can be a *positive triple* +<concept, relation, input> where the input is text entered by players in the game. For instance, `+<teapot,UsedFor,making tea>` indicates that the concept *teapot* can be used for *making tea*. We also collect negative knowledge as *negative triples* —<concept, relation, input> that indicate that the relation and input do not apply to the concept. For instance, `--<teapot,UserFor,running>` indicates that the concept *teapot* cannot be used for *running*.

Discriminative knowledge. We also aim to collect discriminative knowledge. This knowledge is represented by positive quadruples +<concept#1, concept#2, relation, input>, where the relation and its associated input apply to concept#1 but not to concept#2, allowing to discriminate between the two. For instance, `<teapot,shoe, UsedFor,making tea>` indicates that the concept *teapot* is different from the concept *shoe* in that only the teapot can be used for making tea. Negative quadruples instead, mean that the relation and input do not allow to discriminate between the two concepts.

9.3.2. GAME MECHANICS OF `FindItOut`

Initialisation. At the start of the game, both players are presented with a board of multiple cards, that represent different semantic concepts. Each card shows a picture that illustrates the concept, its name, and its potential definitions when one hovers over the card. Game boards can be configured and laid out based on target requirements. These boards are generated with a greedy approach: once a few initial concepts are retrieved

[2] https://sites.google.com/view/finditout-www22/home

for one board, other related ones are appended to the board, either by searching within the WordNet taxonomy, or by adapting to the task at hand — when one wants to understand the difference between two pre-defined concepts, these two concepts can be added simultaneously).

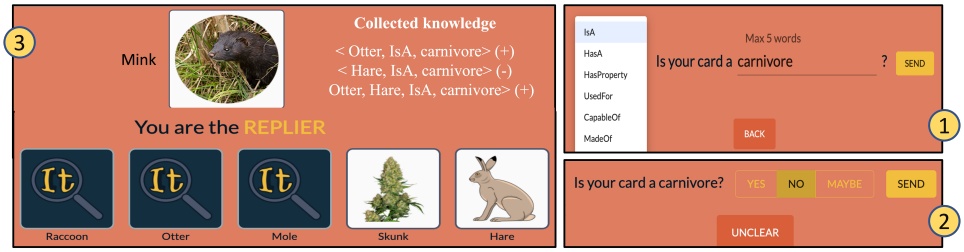


Figure 9.1: `FindItOut` main interface and workflow. (1) The `Asker` inputs a question. (2) The `Replier` selects an answer. (3) The `Asker` flips relevant cards. Example collected knowledge from this turn is presented in the right top corner of (3) (not in game).

The game randomly assigns a card on the board to each player as their `IT card`. The main goal for each player is to guess the opponent's `IT card` (before their card is identified) by iteratively asking questions and eliminating the possible candidates based on the opponent's responses. The game difficulty can be configured, affecting the number of cards on the board. Game boards with more cards are expected to be more challenging, since they require players to think of questions that ideally discriminate between more concepts simultaneously. We also expect that these boards push players towards articulating more tacit knowledge.

Taking turns in questioning and answering. To balance out the opportunity to win for both players and following the best practices for knowledge elicitation through GWAPs [320], the two players take turns playing the roles of the `Asker` and the `Replier`.

Let Player One be the `Asker` for a given turn. They are given the choice between two actions: `ASKing` or `GUESSing`. Choosing `ASK` prompts Player One to formulate a question to ask Player Two. Player Two is then asked to answer Player One's question, by selecting one among four choices: "yes", "no", "maybe", "unclear". "Maybe" is an appropriate answer in cases where it is ambiguous whether a relation applies to a concept, or if it applies only under certain conditions. Selecting "unclear" indicates that the question needs to be reformulated by Player One, since Player Two failed to comprehend it. Depending on the answer, Player One flips the cards on the board by clicking on them to eliminate them from contention, and narrow down the possible candidates for Player Two's `IT card`. It is then the end of the turn, and Player Two becomes the `Asker`.

Choosing `GUESS` allows Player One to designate one card on the board as their guess for being Player Two's `IT card`. Then, Player Two is prompted for their own guess, after which the game ends. Player One wins if their guess matches Player Two's `IT card`, otherwise they lose. This action can only be chosen after each player has asked either 2 or 3 questions depending on the easy or difficult game levels respectively. This design choice dissuades players from attempting random guesses that would not contribute to knowledge creation. Figure 9.1 illustrates this workflow and gameplay.

Question formulation. The questions formulated by the `Asker` follow a template

<relation, input>. The *relation* is selected among a pre-defined set of relations, and the input is a natural language proposition to be manually entered by the `Asker` limited to 5 words (for ease of post-processing and to limit the potential for cheating).

We adopt this template-based question answering strategy since previous works have demonstrated their potential efficiency. For instance, the OMCS project [761, 760] identified that structured, relation-based templates are more efficient at collecting rule-type knowledge and the results are more usable than relying entirely on natural language. Thus, by using a combination of template-based and natural language question formulation, `FindItOut` provides us with the configurability of tuning the potential target knowledge.

Taboo words. We employ taboo words to ensure that the questions asked by the players are not too simple, and allow to extract useful knowledge. We prevent players from entering natural language inputs that contain words with the same root as the concepts on the board. For example, if a concept on the game board is "bird", a player cannot ask "is my card a bird?". New taboo words can be added over time to prevent collecting redundant knowledge.

9.3.3. POST-PROCESSING THE RESULTING KNOWLEDGE

Extracting knowledge. We process each turn to create knowledge based on heuristics. After receiving a response from the opponent, the asker's flipping card actions provide all information needed to gather new tuples in the form of `(+/-)<card?,relation,input>`, where "`card?`" and sign `(+/-)` are inferred based on whether the card is flipped. Specifically, when the answer to a question is received, the relation and input in the question directly apply to **batch A: reserved cards**, *i.e.,* the batch of cards that were previously unflipped and that remain unflipped, with the sign corresponding to the answer (yes is +, and no is –). The batch of cards that were previously unflipped and are flipped during the turn (**batch B: flipped cards**) receives the inverse of the sign of the answer. For example, consider the sequence where the question is "does my card have wings", the answer is "no", and then the `Asker` flips the "bird" card, we build the knowledge triple `+<bird, has, wings>`.

Discriminative knowledge is extracted with two concepts in the batch (both A and B) and with a quadruple template. Any concept pair can be gathered to generate discriminative knowledge, which results in $\binom{n}{2}$ (n is the game board size) tuples of knowledge. Considering one concept from each batch allows us to create positive discriminative knowledge, while both concepts from the same batch result in negative discriminative knowledge.

Quality control. It is in the best interest of the `Replier` to lie when replying to a question, such that the `Asker` will be misled (rational game user model [320]). We tackle this issue through our game design. At the end of a game, both players are shown the opponent's `IT card` and their own question history, and can report errors/wrong answers or foul play for any turn. When extracting knowledge from turns, we filter out reported turns automatically and identify outliers for exclusion manually (*e.g.,* players who do not flip cards as required, cheat in the game, ask meaningless questions).

9.3.4. TECHNICAL IMPLEMENTATION

`FindItOut` is implemented as a real-time, responsive web app (see Appendix), for convenience and portability (the game can be served on any platform as long as it supports a web browser). It supports interactions with both voluntary players connecting onto the app, and with players recruited from paid crowdsourcing platforms.

Design choices. The card data are retrieved by querying WordNet for concept definitions, and Google Search for visual representations of the concepts. In the current version of the game, we selected 8 relations, extracted from ConceptNet [507] (IsA, HasA, HasProperty, UsedFor, CapableOf, MadeOf, PartOf, AtLocation) –see Appendix Table 9.3–, based on their commonality, their applicability to nouns, and adaptedness to the concepts displayed in the game boards. Currently, we propose two game difficulties: easy with 8 cards on the board, and difficult with 16 cards.

9.4. EXPERIMENTAL SETUP

`FindItOut` is designed to be configurable and modular, and thereby to facilitate the elicitation of accurate and diverse knowledge (the concepts we collect knowledge about in this study are chosen to be both abstract and concrete nouns). It is designed to create an enjoyable experience for players, while serving as an efficient means to gather knowledge. These are the objectives we evaluate next.

9.4.1. MEASURES AND METRICS

We evaluate `FindItOut` through a combination of qualitative and quantitative analyses of the resulting tuples across the two difficulty levels. In identical conditions, no GWAP with crowdsourcing can serve as a directly comparable baseline. Hence, we leverage the standard evaluation lens used for knowledge collection systems [917], in addition to a qualitative analysis of the knowledge and of the enjoyability of the game. These measures are described below:

Efficiency of knowledge collection. We measure the number of tuples (positive and negative triples and quadruples) resulting from the game, as well as the fraction of overlapping knowledge tuples generated by the two players across games and turns. By also considering the average time and number of rounds that a `FindItOut` game lasts as well as its cost, we can measure the throughput and utility of knowledge generation.

Qualities of collected knowledge. We analyze how correct and diverse each resulting tuple is. To this end, we leverage an objective measure — the types of relations that are used during the games, and a subjective measure — we manually rate each resulting tuple on several dimensions (meaningfulness, correctness and multiplicity of interpretations, bias, typicality, specificity, tacitness).

Player experience. We use the player experience inventory questionnaire [1] to evaluate the experience of the players with `FindItOut` and discern the extent to which they enjoy it. Players are asked to complete this questionnaire at the end of all the games that they choose to play in a session. At this stage, we also collect open-ended comments and remarks about the game from players.

9.4.2. USEFULNESS OF COLLECTED KNOWLEDGE

Although the aforementioned measures can help us to understand and quantify the characteristics of the generated knowledge, they do not directly highlight the usefulness of elicited knowledge for concrete AI tasks. To address this, we investigate the usefulness of the generative and discriminative knowledge that we collect, by considering two independent and popular tasks.

Coverage of the 'Discriminative Attribute' task. The discriminative attribute task was introduced as a part of the 2018 SemEval challenge [451], and consists in "predicting" whether one word allows to discriminate between two concepts (*e.g., urine* is a discriminating feature in the word pair of {*kidney, bone*}). This corresponds well with the discriminative knowledge that we collect through `FindItOut`. Hence, we investigate the extent to which populating boards in our game with the concepts of this task and having players interact with these boards allows us to collect such knowledge. We thereby compute the coverage of the elicited knowledge with the discriminative words of the task.

Taking <concept1, concept2, feature> triples from the discriminative attributes (DA) dataset as reference, we first retrieve knowledge tuples extracted from `FindItOut` that share both concepts. Taking these tuples as candidates, we generate reference-candidate pairs to be annotated. We spread the coverage evaluation (whether candidate tuple covers the reference triple) tasks to 5 volunteers, with 10% reference triples in overlap. To make the knowledge tuples readable, we generate statements for both reference and candidates.

Tacit clues for commonsense reasoning. Usefulness of generative knowledge is typically evaluated by measuring the performance gains in subsequent inference tasks, such as question answering which requires rich commonsense knowledge [917]. We generate game boards to extract tuples for a subset of the commonsense question answering (CSQA) benchmark [818], and assess whether the extracted knowledge helps conduct commonsense reasoning.

After generating knowledge tuples, we use SimCSE [277] as a retrieval toolkit to obtain top-k ($k = 5$) relevant candidates for each question-choice pair. To retain candidates which are highly relevant to questions, we filter out those with a similarity less than 0.5. We only retain questions which have at least 10 candidates reserved for all choices, and thereby obtained a subset of 179 questions. Next, we carry out a manual evaluation to label whether candidate knowledge tuples are (1) correct, (2) highly relevant to the question and possibly helpful to infer the answers, or (3) directly confirm the answer or discard a distraction term. Furthermore, we assess whether the collected *useful* knowledge tuples are covered by the primary existing commonsense knowledge base – ConceptNet.

9.4.3. PARTICIPANTS AND PROCEDURE

Players. We recruited participants from the Prolific crowdsourcing platform [606] to play `FindItOut`. All participants were proficient English-speakers above the age of 18 and they had an approval rate of at least 90% on the Prolific platform. We excluded participants from our analysis if they do not flip cards as expected, or represented an outlier in terms of cheating in game (e.g., tell opponent their IT card or give wrong answer quite often) or asking meaningless questions. All participants were rewarded with £2.5, amounting to an hourly wage of £7.5 deemed to be "good" payment by the platform.

To encourage participants actively play the game, we rewarded participants with extra bonuses of £0.15 for every win. The players are randomly matched by our system when entering the game, and do not know each other. Players are asked to play 5 mandatory games, three at the easy difficulty level and two at the difficult level. The progressing difficulty allows players to gradually familiarize themselves with the game mechanics. After finishing these five games, the players can play additional games or leave with exit-questionnaire.

Generating Game Boards. For the CSQA task, concepts that appear within a same question are appended to one board (*e.g., {aircraft, school, mexico, battle, human, band, factory, doctor}*, or *{countryside, painting, village, train, ground, mountains, rock, cottage}*. In case of the discriminative attributes (DA) task, concepts from a same triple and from the same semantic field are chosen (*e.g., {mirror, necklace, cigarette, lantern, candle, scarf, lamp, chandelier*}, or *{father, king, daughter, son, prince, uncle, brother, cousin*}).

Concepts from the DA task. To cover as many triples from DA dataset as possible with a limited budget, we only consider triples which contain both frequent concepts (*i.e.,* occur at least 5 times in positive discriminative triples). Using every concept as a seed, we generated game boards with a greedy search strategy to maximize the triples possibly covered. Considering that game boards of a good diversity can potentially create a better game experience, we filtered out game boards which have overlapping concepts (with a threshold of 2 for easy games and 6 for difficult games). Finally, 41 easy game boards and 22 difficult game boards were generated.

Concepts from the CSQA task. We select the questions from the CSQA dataset [818] that refer to at least 5 meaningful single-word concepts (both question concept and choice concept), resulting in a subset of 864 questions. Similar to the generation of boards for DA dataset, we utilized a greedy search strategy to maximize concepts that occur in the same question to be placed in one game board. With this criteria, multiple questions can be "merged" into one board (see Appendix 9.6). Finally, 115 easy game boards and 70 difficult game boards were generated pertaining to the CSQA task.

9.4.4. QUALITATIVE ASSESSMENT OF KNOWLEDGE

Definition of qualitative dimensions. Owing to the lack of automated and standardized methods to evaluate the quality of knowledge elicited through GWAPs, we carried out a qualitative evaluation of the generated knowledge with respect to the '*correctness*' and '*diversity*' of the knowledge. We manually rated the factual *correctness* of a tuple with either '`correct`', '`incorrect`', or '`not sure`' (when in doubt). We followed an iterative coding process [799] to characterize the *diversity* of the knowledge based on several dimensions informed by related literature in computer science and social science —*correctness, truth, bias, tacitness, typicality, specificity.* Table 9.1 presents the dimensions used to assess the knowledge tuples. Knowledge is by definition true [644], and it is thus challenging to rate into more than a binary proposition. Hence, we do not use the same Likert-scale dimension as previous works [761], but propose a multi-dimension description of *correctness.*

Annotation procedure. We analyse the qualities of the generative knowledge by selecting and annotating a subset of samples collected from the game boards pertaining to the DA task. We randomly sample 30 difficult games (leading to 1628 generative knowledge

Table 9.1: Dimensions on which knowledge tuples are analysed. Labels correspond to the scales used to gather annotations.

	Dimension	Description	Label	Example
Correctness	Validity	A valid tuple is comprehensible [879], and the input is not the result of cheating (*e.g.*, description of visual	invalid	+(tap, UsedFor, can your card used home), +(mother, HasA, color brown in it)
			valid	+(camel, AtLocation, in africa)
	Truth	Indicates whether a tuple represents a correct fact.	correct	+(lamp, HasProperty, makes light)
			incorrect	-(mole, IsA, predator), -(squirrel, UsedFor, swimming)
	Meaning(s	Indicates whether the tuple can have different interpretations (among	multipl	+(tower, CapableOf, be used as home) (high-rise building/Eiffel tower)
			single	+(avocado, HasProperty, green (most part))
Diversity	Bias	A tuple can be biased due to being true only in certain contexts, since one can biased by their own view of the world.	unbiased	+(cucumber, IsA, fruit), -(dishwasher, UsedFor, preserving food)
			biased	+(crab, HasA, big claws), -(trousers, usedFor, mainly women)
	Typicality	Indicates the perceived typicality of a tuple from one's point of view (so as to	high	+(boat, AtLocation, on water), -(plug, UsedFor, restraining something)
			mediun	+(car, UsedFor, single person), -(finger, AtLocation, on furniture)
			low	+(fan, IsA, mostly black in colour), -(aunt, UsedFor, a married person)
	Specificity	Indicates the level of details provided by the input in the tuple. Negative tuples are always specific as there can be an infinite number of negative examples.	high	+(skirt, IsA, typically female clothing), -(tap, UsedFor, restraining sth.)
			medium	+(zebra, AtLocation, in africa), -(catfish, HasA, shell)
			low	+(lamp, HasProperty, makes light)
	Tacitness	Indicates whether one would have a hard time articulating the fact, and the	high	+(crab, HasA, red shell when cooked), -(bed, PartOf, kitchen appliance)
			mediun	+(crocodile, AtLocation, jungle), -(avocado, PartOf, group or bunch)
			low	+(elephant, IsA, herbivore), -(lion, IsA, herbivore)

tuples), gather the concepts they cover, and then select all knowledge tuples collected through easy games for which the boards include some of the previous concepts (147 games, and 2429 knowledge tuples). The discriminative tuples can be generated from two generative tuples with different signs. Hence, the quality annotation for discriminative tuples is covered by that of generative tuples. 5 authors of this paper annotated 50 generative knowledge tuples selected at random with respect to these dimensions, and refined the codes together until complete agreement was reached. Following this, each of the authors independently annotated **793** tuples, including a common subset of **95** tuples, allowing us to measure the inter-annotator agreement. The Krippendorff's α scores are respectively 0.91 for meaningfulness, 0.37 for correctness (with 0.38 and 0.45 for problematic sign and relation), 0.31 for bias, 0.23 for typicality, 0.39 for specificity (0.51 when using only two values), 0.33 for tacitness (0.43 when using only two values). Disagreement is due to the subjectivity of the task: knowledge and the veracity of a fact vary depending on one's own experience of the world.

9.5. RESULTS & DISCUSSION

9.5.1. GAME EFFICIENCY

Knowledge quantity. Overall, 255 (164 easy, 91 difficult) and 242 (142 easy, 100 difficult) games were played for the DA and CSQA datasets respectively. This led to collecting 75,491 and 85,923 knowledge tuples. For the DA dataset (and the CSQA dataset respectively), 5.28% (4.39%) of the tuples are generative positive tuples, 6.38% (6.66%) generative negative tuples, 22.8% (20.4%) discriminative positive tuples, and 65.6% (68.5%) discriminative negative tuples.

91.1% of the knowledge tuples pertaining to the DA game boards and 97% w.r.t. CSQA boards consist of unique tuples, while the remaining tuples were generated multiple times across turns or games. On average, easy games lasted 367.2s (*SD=722.3*) in case of DA game boards and 377.8 (*SD=192.3*) for CSQA boards, and corresponded to 3.88 (*SD=1.63*) turns on average for DA game boards, and 4.09 (*SD=1.41*) for CSQA boards. Similarly, difficult games lasted 397.5s (*SD=201.4*) for DA boards —resp. 428.4 (*SD=204.3*) for CSQA boards—, and required 5.69 (*SD=1.98*) —resp. 5.78 (*SD=1.63*)— turns.

Throughput. Overall, for the DA dataset, 13.9 tuples are generated per minute, which is ten times more than Verbosity [854][3].

We define the throughput of our game as the number of elicited tuples divided by the time it took (in seconds) to elicit them. In Figure 9.4, Figure 9.5 (cf. the Appendix), we report the throughput of our game for both the DA and CSQA tasks, depending on the round of the game, and the type of knowledge tuple elicited. In both cases, the throughput decreases over rounds as there are less uncovered cards in latter rounds, leading to the generation of less tuples when flipping new cards. As expected, the throughput is higher for difficult than easy games, especially for the first rounds of the game. Since there are more cards on the game boards in difficult games, and players are incentivized to ask questions that eliminate as many cards as possible, more knowledge is directly elicited from the early rounds. That is also the reason why the difference between the amounts of discriminative and generative knowledge is higher for these difficult games than the easy ones (a "good" question for the `Asker` leads to an optimum number of flipped/unflipped cards to generate many discriminative tuples). No major difference is observed across datasets as the game mechanics remain the same.

Utility. We compute utility as the fraction of value extracted per unit of time (in seconds) over the cost (in pounds). For the DA dataset, we consider the value extracted to be the number of tuples elicited that are tacit, specific or atypical, as these are tuples that cannot be easily collected from other sources. For the CSQA dataset, we consider the value extracted to be the number of tuples that are correct and relevant for the CSQA task. In Figure 9.2 and Figure 9.3, we report the normalized utility for the two datasets depending on the round and difficulty of the game. The average utility does not vary significantly over time for the two tasks, albeit with large standard deviations. This is explained by the high variation in the type of knowledge that players elicit through the rounds. Difficult games correspond to a higher utility of `FindItOut` for the DA task, while easy games correspond to a higher utility for the CSQA task. In general, larger game boards can aid the generation of more valuable knowledge tuples efficiently due

[3] According to the approximate numbers reported: *29.47/23.58 = 1.25* tuple per minute.

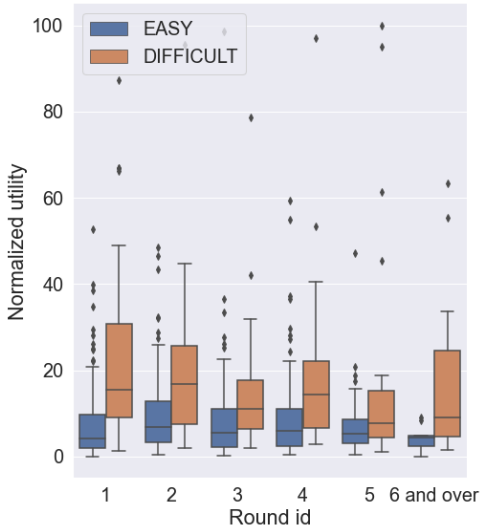


Figure 9.2: Utility of `FindItOut` for the discriminative attribute, and computed over different rounds and difficulty levels.

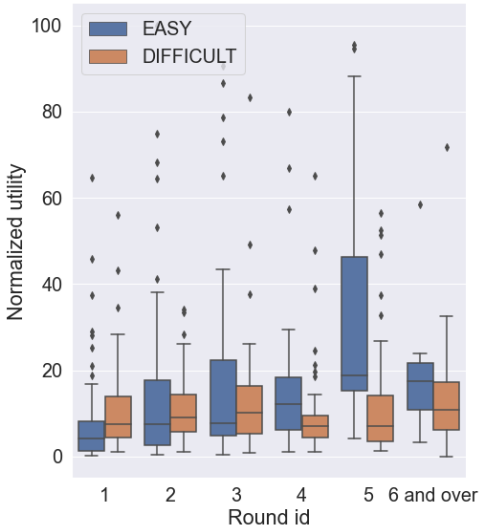


Figure 9.3: Utility of `FindItOut` in relation to the CSQA dataset, and computed over different rounds and difficulty levels.

to more cards being included. As CSQA game boards are generated based on questions, the smaller the game board the higher is the probability to focus on specific questions. This highlights the benefit of configurability of `FindItOut`.

9.5.2. ANALYZING KNOWLEDGE QUALITIES

Below, we report our results for the discriminative attribute dataset.

Correctness. Overall, 95.6% of the generative tuples elicited are meaningful. Among these, 90.6% of the tuples are correct (88.8% and 92.1% respectively for positive and negative tuples). As comparison, Verbosity [854] reports 85% of correct generative tuples elicited. Similarly, 76.2% of the discriminative tuples elicited are correct.

Qualitative study of diversity. As a first indication of the diversity of knowledge types elicited through our game, we investigate the types of relations used by the players. 21.4% of questions employed `IsA`, 20.0% `HasA`, 13.9% `UsedFor`, 13.4% `HasProperty`, 13.1% `CapableOf`, and the other relations in proportions lower than 10%. As each relation corresponds to a different type of information, this shows the diversity of tuples our game collects. A chi-square test of independence to examine the relation between the relations employed by players and the rounds revealed a significant relation, $\chi^2(77, 4235) = 620.59, p < .000$, implying that the relations employed evolved over rounds. In earlier rounds, `IsA` is primarily used as it allows to ask simple, discriminative questions. In later rounds, the frequency of the other relations increases, as more tacit questions need to be asked to distinguish the unflipped cards.

Dimensions. Our qualitative analysis of the elicited knowledge tuples reveal a high diversity in the type of knowledge collected. 86.3% of the tuples are unbiased, 38.3% are

Table 9.2: p-values for Chi-squared tests of independence that were conducted to examine the relation between game rounds and each dimension of the qualitative analysis (†: significant relations).

Level	Correctness	Bias	Typicality	Specificity	Tacitness
All	**3.41e-15**†	**4.55e-08**†	**1.94e-05**†	**1.89e-06**†	**4.89e-04**†
Easy	**5.40e-17**†	**5.22e-04**†	**1.46e-03**†	**1.39e-03**†	2.81e-02
Diff.	**1.22e-05**†	**1.11e-06**†	**2.06e-08**†	**2.24e-03**†	**6.15e-06**†

highly tacit (21.3% medium), 57.5% highly specific (16.9% medium), 7.98% are atypical. These findings confirm that `FindItOut` allow for externalizing tacit knowledge, that is typically not found in existing knowledge bases.

We investigate how the types of knowledge evolved over the rounds, with respect to easy and difficult games, and overall. To this end, we performed Chi-square tests of independence between the annotations of each knowledge dimension and the rounds in the game. To correct for error inflation due to multiple tests, we applied a Bonferroni correction so that the significance threshold of α decreased to $\frac{0.05}{15} = 0.003$. In Table 9.2, we report the p-values of these tests. Overall, we found that each knowledge dimension evolves across the rounds in which the tuples were elicited. This is consistent across easy and difficult games, except for the tacitness of tuples corresponding to easy games. In Figure 9.8, we show the percentage of tuples per dimension collected for each round of the game. This indicates the trend of evolution per round. We found that the number of high typicality tuples decreases over rounds, while tuples with high specificity and high tacitness tend to increase after the initial rounds. The reason for such observation is twofold. After several rounds of a game, reserved concepts are hard to discriminate with general and explicit knowledge. Along the game and its active guessing and thinking mechanisms, players' deeper insights and life experiences are activated/awakened [210].

9.5.3. USEFULNESS FOR AI TASKS

Coverage of discriminative attributes. With 41 easy game boards and 22 difficult game boards generated for the DA dataset, we can cover 3948 triples at most. Due to a limited budget, 55 participants were recruited to play these games, resulting in 3369 triples potentially covered. To filter out noisy reference triples, we manually labelled their validity and found 2987 valid triples (containing 1649 unique concept pairs). These 2987 valid triples are considered as reference. For the annotations of coverage, 5 authors annotated 1102 common samples, and 9808 independent samples. The inter-rater agreement with Krippendorff's α was found to be 0.47, which is reasonable in a subjective task [169]. To evaluate how the generated tuples go beyond the DA dataset, we analyse the correctness of all the candidate tuples (5485) used in coverage annotation. 5 authors annotated 545 common samples, and 4940 independent samples. Inter-rater agreement with Krippendorff's α was found to be 0.43.

For every reference triple, we take all positive discriminative knowledge which have the same concept pairs as candidates. Based on the annotations, we found that 859 (28.8%) of the reference triples are covered. Besides covering a part of the reference triples, we also look into whether the collected candidates can discriminate concept pairs. As manual annotations show, all 1649 concept pairs can be covered with our ex-

tracted knowledge, which indicates the extracted knowledge is of high quality and can even go beyond the scope of the DA dataset.

Commonsense question answering. Among 179 questions (every question has five choices), there are 2.82 choices which can find relevant knowledge tuples (correct and possibly useful) per question, and 0.52 choices which can find useful knowledge tuples (correct and can confirm the answer or discard a distraction term). To further verify the usefulness of our extracted knowledge, we find that 20 knowledge tuples (most are tacit knowledge) among 96 unique useful ones (see Section 9.4.2 last paragraph) are not covered by ConceptNet 5.5. This further verifies the usefulness and necessity of tacit knowledge extracted from `FindItOut`. As all game boards for CSQA subset are only played once (due to a limited budget), we argue that with increased redundancy on the game boards, even more useful knowledge can potentially be elicited.

As shown by previous work [388], existing large-scale commonsense knowledge bases (*e.g.,* ConceptNet [780] and CSKG [389]) are not capable of supporting commonsense reasoning. `FindItOut` fills this gap, by generating both tacit and negative knowledge that is absent from these knowledge bases. Besides reasoning, this negative knowledge can also be leveraged in the future to discard ridiculous inferences and inference mechanisms from machine learning models, which contrast with human commonsense and ethics. This is of great potential to provide trustworthy and robust AI services.

9.5.4. PLAYER EXPERIENCE & ENJOYABILITY

Based on our findings from the player experience inventory questionnaire, the main functionalities of `FindItOut` were well understood and appreciated by players. On average players rated the functional consequences (*i.e.,* "the immediate experiences as a direct result of game design choices") with >1 on a scale of -3 to 3. The ease of control and clarity of goals were the best rated dimensions by the players. These highly-rated functional consequences translated into highly rated psychosocial consequences (*i.e.,* "the second-order emotional experiences, such as immersion or mastery") as well, with an average rating per dimension always above 0. This shows that `FindItOut` was enjoyed by players, arose their curiosity by prompting them to think of topics (differences between concepts) that they probably do not typically think of.

9.5.5. CAVEATS AND LIMITATIONS

Considering that game boards play an instrumental role in shaping the nature of the elicited knowledge, it is important that knowledge requirements are translated well into populating the game boards with concepts. To increase the diversity in knowledge, increased redundancy between game boards is required. In this work, we did not explore how `FindItOut` can be extended to the voluntary player contexts where game elements will play an important role. To generate useful and correct knowledge from `FindItOut` automatically, further mechanisms need to be developed to avoid costs entailing human annotations.

9.6. CONCLUSION

In this paper, we developed a configurable game `FindItOut` to elicit plural knowledge from human players, that could be used to obtain expected mechanisms for a machine learning model. We evaluated and demonstrated the efficiency of the game, the enjoyable player experience it facilitates, the utility and usefulness of the resulting knowledge, through two downstream AI tasks — commonsense question answering and the identification of discriminative attributes. Results showed that our game can generate high-quality discriminative knowledge which goes beyond an existing frame of reference. More importantly, `FindItOut` can generate tacit and negative knowledge which is absent from most mainstream commonsense knowledge bases. `FindItOut` can be easily configured to suit diverse requirements of downstream AI tasks by varying seed concepts, difficulty levels, size of the game boards, the relation sets used for populating question templates, the admissible length of the natural language input from players, using text or image modes, expanding the taboo words that players cannot enter, among other features. Now that we have a method to collect expected mechanisms for a machine learning model via `FindItOut` and another method to collected learned mechanisms of the model via `SECA`, we can now investigate in the next chapter (Chapter 10) to what extent these types of information can be useful for a machine learning developer to diagnose hazardous failures of their models.

APPENDIX

ADDITIONAL DETAILS ON OUR GWAP

Design choice. `FindItOut` can be adjusted to fit different requirements. Here, its parameters (*e.g.,* number of trials before a guess) were calibrated through pilot studies with crowdworkers, geared towards effectiveness and enjoyability of the game. We selected 8 and 16 cards to vary the game difficulty as players managed to formulate interesting questions with less or more effort, while still finding the game enjoyable. The relation-based templates we used to formulate questions are shown in Table 9.3. For SimCSE, relevant literature [277, 734] adopted top-k ($k = 3, 5, 10$) and filtered out low similarity candidates. We set $k = 5$ and similarity threshold to 0.5 for the trade-off between annotation efforts and evaluation quality.

Table 9.3: List of relations used in `FindItOut`.

Relation	Explicit question
IsA	Is your card a(n) _____?
HasA	Does your card have a(n) _____?
HasProperty	Is your card _____(property)?
UsedFor	Can your card be used for _____?
CapableOf	Can your card _____?
MadeOf	Is your card made of _____?
PartOf	Is your card part of (a) _____?
AtLocation	Can your card be found at _____?

Implementation. `FindItOut`'s backend API manages the game logic, and the frontend renders the game screens. The communication between the two ends consists of classic

HTTP REST API for user information, JWT authentication and WebSocket for game lobbying and gameplay, allowing for continuous and bidirectional data flow between the server and client. It is written in Python and served with Flask owing to its simplicity and fast setup. All game data are stored in a PostgreSQL database. The server/client WebSocket communication is implemented using the Socket.IO library. The frontend is written using React javascript library in conjunction with Redux state library, which allows unidirectional data flow; making it predictable, easy to test and flexible.

GAME BOARDS

In our game, the design of game boards is of great importance. To keep the game interesting, we adopted greedy search strategy to retrieve relevant concepts and generate game boards for Discriminative Attributes dataset. The algorithm to generate game boards for DA dataset can be found in Alg. 1.

Algorithm 1 The algorithm to generate DA game boards.

Require: Triple set $\mathscr{T}$, concept set $\mathscr{C}$, game board size n.
Input: seed concept c_0.
Output: Game board g.
initialize game board $g = \{c_0\}$
for $i = 1 \ldots n-1$ **do**
    c_i = MaximizeTripleCover($g, C \setminus g, \mathscr{T}$)
    $g = g \cup c_i$
end for
return g

To generate useful knowledge for the question answering task, we based ourselves on questions of the CSQA dataset to generate game boards. Based on concepts mentioned in a question and its choices, we gather related questions and generate game boards with clustering methods, which take every question as a node and overlap of concepts between questions as edges. The algorithm to generate game boards for CSQA dataset can be found in Alg. 2.

Algorithm 2 The algorithm to generate CSQA game boards.

Require: Question-concept connection set $\mathscr{T}$, question set $\mathscr{Q}$, game board size n.
Input: seed question q_0.
Output: Game board g.
initialize game board g = ObtainQuestionConcepts($\mathscr{T}, q_0$)
initialize covered question set $\mathscr{Q}_c = \{q_0\}$
while Size(g) < n **do**
    q_i = MaximizeConceptOverlap($g, \mathscr{Q} \setminus \mathscr{Q}_c, \mathscr{T}$)
    $g = g \cup ObtainQuestionConcepts(\mathscr{T}, q_i)$
    $\mathscr{Q}_c$ = FindQuestionCovered($g, \mathscr{Q}, \mathscr{T}$)
end while
g = FilterGameSize(g, n)
return g

ADDITIONAL RESULTS

Analysis of correctness. When tuples are incorrect, 62.3% a flipped sign, 29.9% a problematic relation, and 7.49% both a sign and a relation. Problematic relations are typi-

cally explained by a) the fact that a relation and its corresponding natural language input make sense in the question posed by the `Asker` of a round, but not necessarily in the generated tuples where the concept of the game boards might not all be related to this tuple, and b) the difficulty for some players to interpret the different relations. As for the problematic sign, it is either due to ambiguities in the meaning of a concept, or due to players forgetting to cover a card when they receive the answer to their question. Future research would be needed to optimize the post-processing to automatically identify and correct such errors, as well as to improve the user experience in order to support players in selecting the most appropriate relations, and to prompt them to cover all relevant cards at each turn.

Game efficiency. Overall, 2.56% of the knowledge tuples collected within a game are overlapping, and 8.9% of the tuples collected across game boards overlap.

Table 9.4 present the average time taken by round across game level for both the DA and CSQA game boards. The high standard deviation for easy games in the first round is explained by the time taken by the players to learn the rules of the game.

In Figure 9.4 and Figure 9.5, we report the throughput of our game for both the DA and CSQA boards, depending on the round of the game, and the type of knowledge tuple.

Table 9.4: Average time (in second) taken to play a round of the game (round $k = 4$ for easy games and $k = 5$ for difficult ones as more rounds are typically played for the latter).

Game board	Level	round 1	round 2	round k
DA	Easy	176.5 (*SD=735.3*)	91.9 (*SD=57.5*)	74.0 (*SD=40.8*)
	Difficult	85.8 (*SD=33.9*)	72.8 (*SD=38.7*)	66.0 (*SD=36.4*)
CSQA	Easy	141.2 (*SD=88.6*)	100.3 (*SD=68.3*)	76.8 (*SD=69.5*)
	Difficult	94.2 (*SD=46.4*)	81.7 (*SD=43.9*)	74.6 (*SD=51.3*)

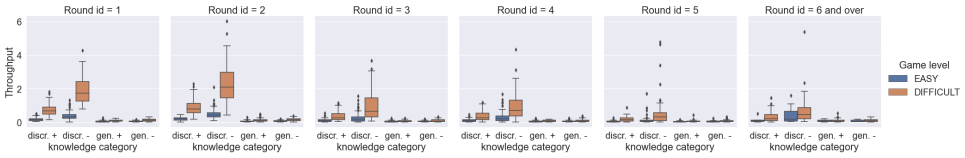


Figure 9.4: Throughput computed over rounds of the game for the discriminative attribute dataset. Round 6 and over are aggregated as less players played them (amount players per round: easy 72, 75, 70, 60, 40, 21 / difficult 51, 50, 49, 50, 44, 40).

Qualitative analysis. In Figure 9.8, we report the percentage of knowledge tuples falling into each of the values of our qualitative dimensions, based on the rounds of the game.

We report in Figure 9.6 the distribution of relations used across rounds of a game. Players tend to use explicit relations (e.g, IsA) to form the questions. After serveral rounds, tacit relations (e.g, UsedFor, PartOf) are used more often.

Enjoyability. We report in Figure 9.7 the enjoyability of the game. Overall, players are

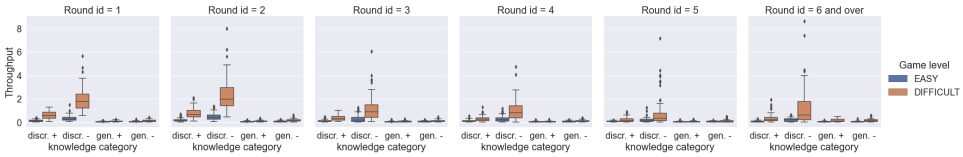


Figure 9.5: Throughput computed over rounds of the game for the CSQA dataset. Round 6 and over are aggregated as less players played them (amount players per round: easy 70, 71, 72, 56, 33, 27 / difficult 59, 61, 55, 57, 59, 51).

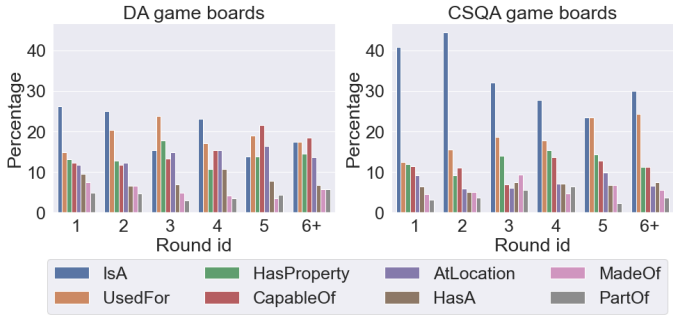


Figure 9.6: Relation distribution along the game rounds.

Figure 9.7: Player Experience Inventory questionnaire.

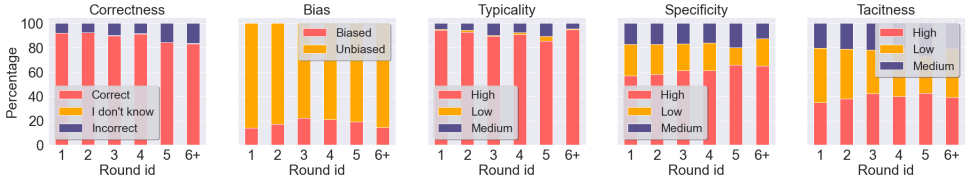


Figure 9.8: Bar plot illustrating the distribution of each dimension in the qualitative analysis of `FindItOut` in relation to the DA dataset, computed across different rounds.

satisfied with the functional consequences, where the average ratings is above 1.0 (scale from -3 to 3).

10

EVALUATING THE USE OF MECHANISMS BY ML DEVELOPERS

10.1. INTRODUCTION

In this last chapter of Part III, we investigate how the explainability method we developed (and other explanations) to obtain learned mechanisms of a machine learning (ML) model, as well as the domain knowledge that our game with a purpose allows to collect (that can be used to elicit expected mechanisms for a model), can be used in practice by ML developers to diagnose their ML models for hazardous failures. Indeed, the ML community develops various explainability methods, often arguing their usefulness for model bugs identification [677, 437, 289, 69, 810]. However, few studies investigate their concrete uses in this process. As a result, it is still unclear what types of explanations (e.g. out-of-domain, global, or interactive [773]) can be useful, for which steps of the process, and how. Hence, we ask: *how could diverse explainability methods be used to support the bug identification process of deep learning computer vision models?* We focus on image classification tasks, as they are prone to model misbehavior, and there is an established body of (post-hoc) explainability methods to possibly support bug identification. Their practical use has not been studied yet, contrary to the ones for tasks that rely on tabular data [365]. We study the identification of model failures and bugs in development; later steps like bug identification in deployment and bug correction are future work.

We draw inspiration from works situated at the intersection of ML and HCI that investigate how ML or related tools (e.g. explainability, debugging user-interfaces, etc.) are used [578], or could be used [365], and how to design them [31, 889]. We build a design probe[1] in the shape of a user-interface by performing literature studies, a formative study, and co-creation sessions consisting of 18 interviews, to explore uses of explainability for debugging. Using an implemented probe and a carefully-crafted use-case, we then perform 18 user-studies with ML developers having different levels of experience with computer vision in various domains.

[1] Code implementation: https://github.com/agathe-balayn/explainability_probe

The user-studies show that a wide range of explanations are useful to identify bugs (e.g., both textual and visual explanations, global and local, companion with domain knowledge, etc.). These explanations are often not theory-heavy, but extremely informative when embedded into an interactive interface. Although they can sometimes be overwhelming and misinterpreted (leading to identify wrong bugs due to confirmation bias), these explanations also allow to identify the potential causes of various issues, and to envision correction strategies. This reveals an urgent need for more research on the design of new explanations relying on diverse user-interactions adapted to different kinds of developers.

10.2. RELATED WORK

10.2.1. BUG IDENTIFICATION IN SOFTWARE AND ML MODELS

To understand what bug identification means, we survey literature about machine learning testing (first step of the debugging process) and traditional software systems where bug identification is more extensively studied.

Failures. Machine learning testing aims at detecting and characterizing differences between current and expected functioning of a model [924]. These differences revolve around inferences (e.g., correctness, robustness, fairness, etc.), data, or code [152, 671]. Main causes of failures are structural or training bugs [519]. Our formative study reveals sub-types of training bugs around datasets or training hyperparameters. We mainly focus on correctness failures (wrong model inferences or features) and dataset bugs, in relation to issues in the model features, as these are still overlooked research-wise despite being the primary debugging goal of developers and directly related to explainability methods. Software engineering distinguishes between *reactive debugging* (a failure is explicitly identified) [34, 328]; *proactive debugging* (no explicit failure manifests); and general *software understanding* (for later debugging) [475, 541], that we all study.

Methods. The *software* debugging workflow consists of four steps [475, 541, 34]: 1) gathering context and hypothesis formulation, 2) instrumenting the hypothesis, 3) testing the hypothesis, 4) correcting the hypothesis, or applying a bug solution. To the best of our knowledge, there is no study of the bug identification *practices* for computer vision models. Instead, research focuses on developing *methods* for debugging models [924, 519, 514, 419] without any human activity or explainability (except [763]). As our formative study shows that none of the automatic methods is employed by developers, we investigate how they could perform manual bug identification supported by explainability.

User interfaces. A few user interfaces [923, 875, 673] support developers in debugging models. None of the ones that make use of explainability methods are adapted to computer vision. The applicable ones all focus on investigating the choice of model and training hyperparameters [729, 728], or visualising the data used to train the model [25]. Our design probe instead presents diverse explanation artifacts designed for computer vision models.

10.2.2. MACHINE LEARNING EXPLAINABILITY

Explainability provides explanations on the functioning of a model. A framework [810] characterizing explainability works identifies model debugging as one of their purposes, and the following tasks developers might perform, e.g., "assessing reliability of a prediction", "detecting arbitrary behavior", etc, that our study also identifies.

Categorization. Explainability methods and resulting explanations can be categorized in various ways [45, 773, 503, 497]. One might want to differentiate them regarding the explanation audience, the explanatory medium, the explanation scope, whether the explanations are about data or models, their faithfulness, etc. Algorithmic research distinguishes between *local* or *global* explanations, depending on the scope of data samples employed. Local explanations provide information on the reasoning a model follows to infer the label of a sample, through saliency maps [754], visual counterfactuals [302], or visualisations of activation layers [595, 366]. Global explanations indicate the general features used by a model, presented as visual hints (e.g., TCAV [437], ACE [289]), or textual information (e.g., SECA [69]). We use these categorizations to identify the explanations relevant to include in our probe.

Usages. Researchers have conducted user-studies on the use of certain explanations for certain stakeholders and data types [23, 889, 409, 174, 185, 427]. Yet, no extensive work involves *developers* debugging *computer vision* models. Only Bhatt et al. [100] conducted inquiry interviews where developers solely reported using saliency maps to understand wrong inferences, or to identify spurious features, and none mentioned other explainability methods. Our work performs a human-grounded exploration where we collect developers' practices based on carefully-crafted debugging tasks.

10.3. METHODOLOGY: PROBE DESIGN PROCESS

The goal of our probe is to explore potential uses of explainability methods for bug identification. Design probes have three fine-grained goals [384]: "*social science* goal of collecting information about the use and the users of the technology in a real world setting, *engineering goal* of field-testing the technology, and *design goal* of inspiring users and designers to think of new kinds of technology to support their needs". Table 10.1 describes the requirements for our probe.

10.3.1. MIXED METHOD RESEARCH

To translate our requirements into a probe, we establish *functionalities* (Fx) to provide, and *orthogonal categories* (Ox) that indicate how these functionalities can be realized. Academic and grey *literature* analyses inform the list of explanations the probe should contain (Rq2). However, it does not focus on developers' experiences, and thus does not inform on other information needed to identify bugs (Rq1). Consequently, we perform a *formative study* in the shape of 18 semi-structured interviews with developers where we investigate their practices, challenges, and wishes. We synthesize the literature and the insights from the study to extract the functionalities (Fx) and orthogonal categories (Ox) (Table 10.2). Finally, we perform iterative *co-creation sessions* where we present designs

Index	Requirement	Description
Rq1	Completeness of functionalities for bug identification	The probe should present the main information a developer might look at when debugging.
Rq2	Completeness of explanations	The probe should offer the main available types of explanations for computer vision models.
Rq3	Clarity of the presented information	To proceed to valid user-studies, the participants should understand clearly the functionalities presented to them, without being overwhelmed by the offered information.
Rq4	Flexibility and objective presentation of the information	The probe should not enforce a certain workflow within the tool not to skew participants' behaviors towards certain explanations, but instead make the interactions as free as possible.
Rq5	Engineering feasibility	The probe should be fully functional to exploit the explanations of an actual model, and to let participants make use of the technology.

Table 10.1: List of requirements defined for the probe.

of the probe to developers, and collect feedback to fine-tune information visualisations, and identify the minimum set of necessary interactions with these functionalities (Rq3, Rq4). The probe is then implemented so as to create a valuable user-experience (Rq5).

Concretely, in the formative study and co-creation sessions, we present to the participants a use-case involving the development of a deep learning model for a scene classification task. We describe an initial model that has been (hypothetically) built, and show example of images from the training dataset with their ground truth and model inferences. We make sure these examples present both cases where the model makes right and wrong inferences, using relevant and irrelevant features (same approach as in section 10.5). For the formative study, we then ask the participants to describe the approach they would follow to define whether this model is ready for deployment, and if not, to characterize what the exact model failures to solve are. We analyse the results of such sessions by extracting intermediate goals (in the shape of questions in Table 10.2) the participants have while investigating the model, and types of information and tools they use to fulfill these. For the co-creation sessions, we ask the same questions. Yet, we additionally present the participants with mock-up user interfaces containing various types of explanations, and prompt them to envision how they would use such interfaces to answer the questions. We also ask them for feedback on the interfaces (e.g. missing, irrelevant, or unclear functionalities), and we iterate on the interfaces after each interview, going initially from low-fidelity mockups, to high-fidelity ones in the last interviews.

10.3.2. PROBE FUNCTIONALITIES

We elicited the functionalities below needed for the probe. We illustrate them with the scenario of the user-studies (section 10.5): building a model that classifies the species of a bird displayed in an image. Importantly, our participants often referred to semantic concepts in relation to relevant sample pixels or potential human-interpretable model features, to reason about potential failures and bugs. These concepts were either entities (e.g. `cactus`), attributes (e.g. `green`), entity-attribute combinations (e.g. `green-cactus`), or their logical negation (e.g. `NOT cactus`, i.e. absence of cactus).

- **F1: performance understanding**: Understand overall and class-specific performance of the model. Looking at metrics gives a first indication of the performance of the model, and the type of errors to investigate. Participants use the class-specific metrics

Table 10.2: Summary of observations from the literature and formative study, and their mapping to the probe functionalities (Fx) and orthogonal categories (Ox).

Topic	Provenance	Description	Fx	Ox
Model input	Interviews	What kind of data does the system learn from?	F2	-
		To what extent the data is diverse enough to represent each class?	F2, F9	-
		What are the differences between these two classes?	F2, F5, F8	-
Performance	Interviews	How well does the model perform for each class? Errors with high or low confidence?	F1	-
Exp. breadth/scope	Both	The extent to which an explanation can be generalised [773]. Participants of our formative study use a larger range of breadth than local and global.	F3, F4, F8	O1
Local	Both	What features of this instance lead to this inference?	F3	
Global	Both	Which visual elements does the model generally use?	F4	
Intermediate	Interviews	What are the features used to distinguish these 2 classes?	F3, F4, F8	
Comparisons	Both	[365] insists on allowing comparisons of explanations across samples. Participants of our formative study performed comparisons across samples but also across classes.	F5, F6	O1, O2, O3
		Why are instances A and B given the same/different predictions?	F5, F3	
		How does the model weigh different features?	F6, F3, F4	
Exp. family	Literature	Sokol et al. [773] discuss a) associations between antecedent and consequent, b) contrasts and differences (using examples), c) causal mechanisms, as potentially used types of explanations. Our participants primarily relied on b), some also hinted at a) and c).	-	O3
Associations		The local and global explanations mentioned above primarily refer to a).	F3, F4	-
Contrasts		The comparisons performed with these explanations refer to b).	F5, F6	-
Causal mechanisms		Why is this sample predicted P instead of Q? What would the model predict if this sample is changed to ...?	F7	-
Exp. domain /medium	Both	A mixed domain approach consists in explanations within the original domain of inputs (images), and in a transformed domain (essentially text such as in dialogues [72]). A few participants hinted at the potential usefulness of having textual explanations.	-	O2
Interactivity /passivity	Literature	[773] distinguishes between static and interactive explanations. While most explainability works do not address interactivity, some [69, 437, 289, 365] propose query interactions.	F8	-
		This connects to varying the breadth and domain of explanations, performing various types of comparisons, and exploring questions around causal mechanisms.	F5, F6, F7, F8	O1, O2, O3
	Interviews	Does the model use this feature?	F8	-
Domain kno.	Interviews	What features do we expect the model to learn for this class?	F9	-
		Should the model pick up on more visual elements for this image/class?	F9, F3, F4	-

to decide for which types of samples to improve the model first.

- **F2: data-neighbor exploration**: Understand and compare the content of data samples. Participants regularly explore the data to estimate the complexity of the task, to

reason about causes for model failures, and identify features of the model. F1 and F2 can be supported with information about performance metrics and datasets.

- **F3: local explanations**: Understand how the model made an inference for a sample. Participants scrutinize or wish to scrutinize saliency maps, to detect overfitting, or to judge the relevance of model features. This can be facilitated with explanations of single samples that show the connection between the sample content and its label (e.g. the model classified this image as `gila woodpecker` by looking at the pixels of the `cactus` the bird is standing on Figure 10.4).

- **F4: global explanations**: Identify the main reasons for the model to classify samples into this class. Participants progressively achieve a global understanding of the model by formulating a hypothesis based on a single sample, and iterate on it by evaluating its validity across more samples. Some participants wished to have statistical summaries of visual concepts across images (e.g. for 80% images classified as `gila woodpecker`, the model looked at pixels representing a `cactus`) to speedup their process and improve its results. For that, some participants suggested using crowdsourcing or object detectors to annotate images at scale (similarly to what the SECA method offers [69]).

- **F5: explanation comparison**: Compare the reasoning of the model across samples or classes. Comparisons serve to judge the validity of feature hypotheses, and to understand mis-classifications (e.g. the model classified this `gila woodpecker` image correctly using the `cactus` pixels, but that one incorrectly using the `wings`).

- **F6: explanation importance**: Rank the explanations based on their frequency, or on the type of (in)correct inferences they lead to. A few participants mentioned that it would be convenient to automatically obtain a list of the most important features for the model. We foresee they might want to query and rank explanations according to different properties such as explanations that lead to correct or incorrect inferences (e.g. 20% of times when the model used the `breast` and `belly` pixels, it made a correct prediction, contrary to 90% for the `cactus` pixels).

- **F7: counterfactuals**: Ask "what-if" questions to see the type of reasoning and inference class received by a sample with/out these visual concepts. This family of explanation was not directly mentioned as counterfactual, yet a few participants mentioned testing transformations of images based on certain concepts to understand how they impact the inferences (e.g. what would the model predict if there was `no cactus` in the image?). As setting up such transformations is complex (Rq5), we propose proxy textual-explanation based transformations (subsection 10.4.2).

- **F8: explanation recommendation**: Visualise explanations, or search for specific ones. While participants do not talk about this as they are used to search for local explanations by themselves, being able to query specific explanations might speed up their process. This is also connected to the complexity of an explanation method. As participants did not know about many explanations, they did not reflect further on their complexity. Yet, they might want to delve deeper into the parameters of explainability methods once they are more familiar with them.

- **F9: domain expertise**: Know what a domain expert (e.g. an ornithologist) would consider good reasons to classify a sample into a class. This functionality was contentious among participants. A few participants did not use domain knowledge explicitly but still relied on their understanding of the domain to understand potential wrong features of the model, while others advocated for the necessity of understanding the domain even before looking into the model.

We identify the following orthogonal categories:

- **O1: breadth**: While literature refers to local and global explanations as the two scopes of explanations, we see them as the two extremes of a scale. The participants did not always look into a single sample (local) or the overall set of data and inferences (global), but focused on various sets of classes (e.g. two classes or entire dataset), or samples with correct or incorrect inferences.

- **O2: medium**: Participants are more accustomed to image-based explanations. Yet, a few participants insisted on getting textual explanations to more easily receive feedback from domain experts on feature relevance, or to query learned features or images of the training dataset.

- **O3: granularity/type**: Participants typically reasoned about semantic concepts to identify issues with model features. Certain participants varied the granularity of these concepts and went to fine-granularities when they could not identify a pattern of reasoning within higher-granularity ones (e.g. `entire wing` or `sub-parts` with different colors).

10.4. PROPOSITION: RESULTING DESIGN PROBE

We first describe the main types of explanations in the probe (F3, F4, O1 - O3), corresponding to the basic required functionalities. We then explain how we organized these functionalities into a set of interactive tabs, to fulfill the other functionality requirements (F1, F2, F5 - F9). In our user-study (section 10.5), developers will use the probe to debug a model for the bird classification scenario.

10.4.1. MATERIALISATION OF THE PROBE EXPLANATIONS

LOCAL EXPLANATIONS (**F3**)

To vary the medium of explanations (**O2**), we provide both visual ones (saliency maps Figure 10.2 (5a)) and textual ones (semantic features (5b)). We opted for SmoothGrad to retrieve the saliency maps [769]. This method is sensitive to the parameters of a model while minimising noisy results, catering for more accurate capturing of a model behaviour. The semantic features are retrieved as by-products of applying the global explainability method.

GLOBAL EXPLANATIONS (**F4**)

We choose the SECA framework [69] to extract explanations that reflect the overall features of a model. It provides more complete explanations than ACE [289], it is more

Table 10.3: Overview of the scores in the global explanations.

Score	Example
Overall explanations	
Percentage of times the features are used by the model within the dataset.	If 100 images are in the dataset, and the model used the feature "cactus" in 20 of them, then the score is 20%.
Percentage of times the features led to a correct prediction across all images for which the model used the features.	In the 20 previous images, if the model made 5 correct predictions, the score is $5 * 100/20 = 25\%$.
Typicality score (from SECA): correlation between the presence of the features and the predicted classes, i.e. how strongly the features serve to distinguish one class from the others.	If cactus is associated to all gila woodpecker images, but to no image of any other class, then typicality would be high since the correlation would be strong, while wing which is used for all classes would have a lower typicality score.
Class-specific explanations	
Percentage of images that contain the features of interest among all images with the predicted class.	If 100 images were predicted to be a gila woodpecker, and 20 of these images have a cactus, then the score for *cactus* → *gila woodpecker* is 20%.
Percentage of images that received a correct prediction among images that contain the features and have this predicted class.	Among the 20 previous images, if 5 images were indeed gila woodpecker images, then the score is of 25%.
Typicality score (from SECA): indicates how strongly the features serves to distinguish the specific class from the others.	See above example for typicality.

tractable than TCAV [437] in an interactive mode (Rq5), and provides textual explanations that participants wished for.

SECA takes as input images from each class of the dataset (we choose a balanced, random set of samples of the test dataset). It extracts the corresponding saliency maps and has them annotated by crowd workers. Then, it reconciles the annotations, and transforms them into a table of semantic features. Post-processing techniques (e.g. rule mining) finally identify combinations (logical conjunctions or disjunctions) of features (entities and/or attributes) highly correlated with certain predicted classes. This approach provides explanations at different levels of granularity (e.g. `wing`, or `primary coverts`, `alula`, etc.) depending on the granularity of the annotations requested to the annotators (**O3)**. The feature combinations are accompanied with six scores (see Table 10.3) referring to overall explanations and class-specific ones. Overall explanations represent the primary features used by the model to distinguish between classes (see Figure 10.1 (3)). In class-specific explanations, the combinations of features are associated to a specific class (see Figure 10.1 (4)), and the scores indicate the relevance of these features to this class. We represent these scores in bar plots for easy comparisons, as a result of multiple iterations where participants indicated the difficulty in making use of numbers (Rq3).

The user can rank (**F6**) or filter the global explanations according to the scores (Figure 10.1 (5)). To vary the explanation scope **(O1)**, one can compute the scores on various data subsets: (1) entire dataset: explains the general inference mechanisms a model follows; (2) samples that received a (in)correct prediction: identifies and compares mech-

anisms for such predictions; (3) subset of the classes: identifies features used to distinguish between these classes. Where these choices can be made, we setup default parameters to reduce the complexity of the probe understanding (Rq3).

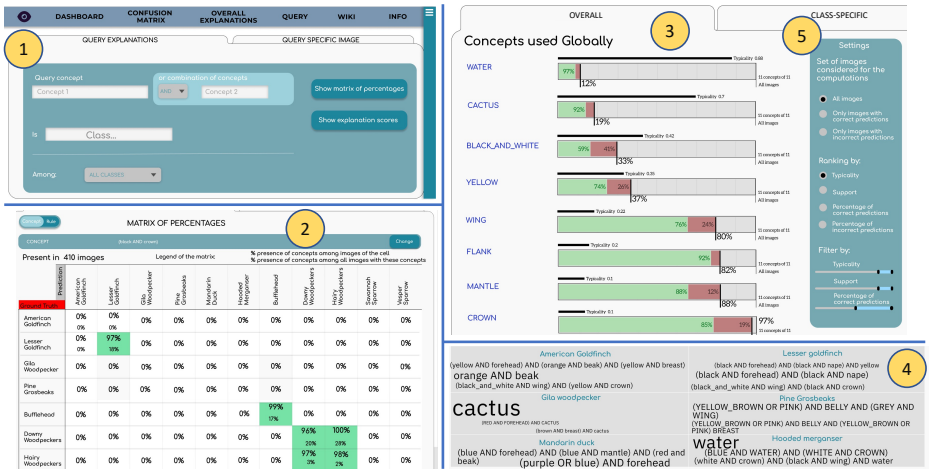


Figure 10.1: Query tab (left) and overall explanations tab (right). When querying (1) explanations, results are displayed underneath (2). The overall explanations tab shows both relevant (combinations of) concepts (3) and their association to each dataset class (4), and allows for varying the parameters to compute them (5).

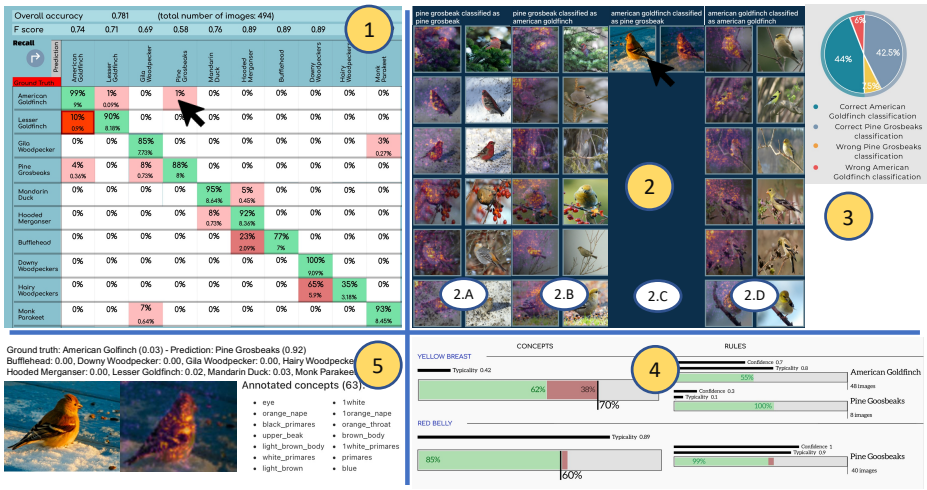


Figure 10.2: Confusion matrix interactions. Our probe allows for different interactions with the explanations. E.g., when one clicks on a cell of the confusion matrix (1) corresponding to the predicted class A and ground truth class B, she is directed towards the corresponding local (2) (images corresponding to the cells A-A, A-B, B-A, B-B of the matrix) and global (4) explanations, as well as more performance indications (3). Clicking on a local, visual explanation displays further local, textual explanations (5).

10.4.2. THE TAB STRUCTURE OF THE PROBE

To avoid skewing the participants towards a particular workflow (Rq4), we organize the primary functionalities into tabs, making them independent and equally important. In the user-study, we inform the participants that there is no sequential dependency between the tabs. The tabs allow us to provide the higher-level explainability functionalities (**F5 - F7**), as well as the other necessary information about the model (**F1, F2, F9**). **F8** (recommendation) is provided all along the probe through the various parameters to choose as well as the query tab.

WIKI TAB (**F9**)

This tab displays the domain knowledge about each dataset class, that an expert typically possesses. It indicates relevant and irrelevant features for recognizing an image class.

QUERY TAB (**F7**)

This tab (Figure 10.1 (1)) allows to query global and local explanations, and images with specific visual content, their predictions, explanations, and ground truth, allowing to a certain extent to answer what-if questions.

The user is presented with text fields to fill in with features of interest, types of logical combinations, and/or class. They choose to query explanations within all images, or only in the correctly or incorrectly classified ones, or within specific classes (**O1**). The results are displayed underneath. These can be a) scores of a queried explanation, b) distribution of the presence of the queried features across the dataset, or c) samples associated to the local query. b) is displayed in a confusion matrix-like table (Figure 10.1 (2)) that shows, per cell, the percentage of images that have the features among the images of a cell, and the percentage of cell images that have the features among all images that have this feature.

CONFUSION MATRIX TAB (**F1**)

This tab shows the accuracy and F1-score of the model, and its confusion matrix (see Figure 10.2 (1)). Each cell presents two rates. One (bottom) is computed over the entire dataset similarly to any confusion matrix. The other (top) is computed over the data of a single row corresponding to the precision or recall per class depending on what the rows and columns encode (ground truth or prediction). One can transpose these.

Users can click on the matrix cells to open a new page with the corresponding images (**F2**), as well as local and global explanations (**F3, F4**). The images and local explanations (Figure 10.2 (2)) are organized into four columns corresponding to the 4 cells of the matrix associated to the classes clicked initially, i.e. the ground truth A and predicted class B of the initial cell (A-B), as well as the corresponding diagonal cells of the two classes A-A and B-B, and the opposite cell that would invert the ground truth and prediction classes (B-A). This allows to compare these explanations (**F3, F5**). Clicking on an image or saliency map allows to zoom on it, and its related textual, local explanations (Figure 10.2 (5)).

The global explanations corresponding to the four cells (equivalent to considering a binary classification task involving classes A and B Figure 10.2 (3)) are also displayed in lists allowing for their comparisons (Figure 10.2 (4)) (**F4, F5**).

GLOBAL EXPLANATION TAB (**F4, F5, F6**)

This tab displays the global explanations computed over the entire dataset. It shows both the overall and class-specific ones (respectively Figure 10.1 (3), (4)).

DASHBOARD TAB

A few participants from the co-creation sessions wished to see all the main functionalities on a unique page. The dashboard tab does so. Its top left part provides the performance functionalities (**F1, F2**), and the top right the corresponding local explanations (**F3**). At the bottom, the query functionality is enabled (**F7**). This organisation lets users explore explanations for different images, and compare these with additional queried information (**F5, F6, F8**).

10.5. EXPERIMENTAL SETUP: USER-STUDY

To study how developers would use explainability methods for bug identification, we conduct 18 user-studies of one hour each. We prompt our participants to answer a design brief with the design probe. We ask them to explain out-loud what they do, and we note their interactions with the probe (order of visited interfaces, functionalities used, etc.). When they identify a potential failure and the related bug, we ask which action they would take to solve it. Each session ends with an exit interview and a questionnaire to collect ratings around the usefulness and usability of the interfaces. The questions combine the short version of the User Engagement Scale [592], and 7-point Likert scale questions around their likelihood to use the probe in the future. Before each session, we ask the participants for their agreement for recording. We later transcribe the recordings into anonymized transcripts, and destroy the recordings. The interview process has been reviewed by the ethics committee of our institution. We analyse the results of the user-study qualitatively in relation to the functionalities and orthogonal categories identified in section 10.3, and quantitatively based on the questionnaires, the count of commonalities in the steps followed by each participant, and the numbers of bugs identified.

Participants. The 18 participants were recruited through the networks of the authors, searches on professional social networks, and by snowball within the contacts of the first eight recruited participants. We only recruit participants who have experience with machine learning, but not necessarily with computer vision (CV), as they should understand the basic concepts around model failures. We categorize the participants based on their level of experience with CV. Low-CV experience participants (6) have never or only once developed a CV model, mid-CV experience participants (5) have less than 4 years of CV model development experience, and high-CV experience participants (7) have more.

Design brief. The design brief presents a model bug identification scenario (Figure 10.3). It is typical and simple enough for participants to reflect on their own practices without envisioning entirely new workflows. Bird classification might require domain-knowledge, raising reflections on the need to have domain expertise for bug identification. We scope the brief to the development setting as it encompasses a varied set of activities, with both reactive and proactive debugging.

BRIEF

Context:
A company wants to develop a system to support bird lovers in identifying the birds they might see in their daily life.

Current model:
An intern developed a deep learning model for 10-class bird classification.
For this, he created a dataset by scraping images from the Web using Google search engine, and applied some typical data augmentation methods (e.g. flipping and cropping images, brightness transformation).
He then fine-tuned a ResNet model pre-trained on ImageNet on this data.

Your task:
Unfortunately, his internship now ended. The company asks you to take over his model. It asks you to investigate whether the model developed by the intern can be deployed, or whether it needs improvement. In this case, what issues should be improved on, and how?

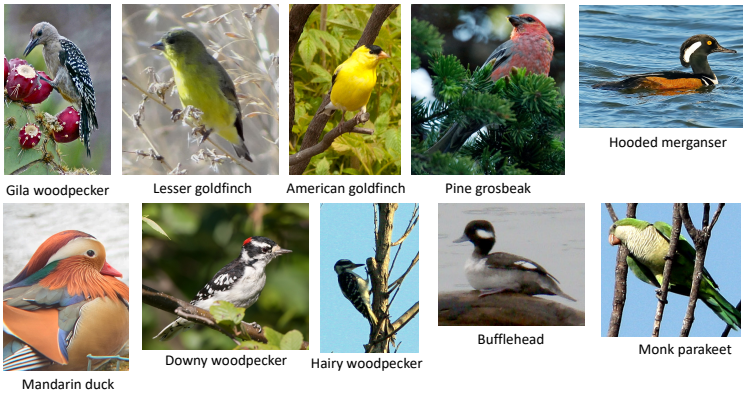


Figure 10.3: Overview of the design brief. Examples of samples of each class the model to be analyzed was trained on.

Model. We train the machine learning model to be debugged to classify 10 species of birds. The training dataset is built with the idea of introducing both explicit (low test accuracy and mitigated confusion matrix) and implicit model failures and various bugs that explain these failures, as summarized in Table 10.4. To the best of our knowledge, there is no established list of bugs and failures for computer vision models. We propose a preliminary one, inspired by the literature on data biases [828, 826, 493], data shifts [350], robustness to adversarial [15] and natural perturbations [351], models using wrong reasoning for making inferences [348], and from the bugs mentioned in the formative study. Besides distribution shifts, we create wrong sets of features (incompleteness or irrelevance) that lead to correct or incorrect predictions, i.e. implicit or explicit failures. See examples in Figure 10.4. To introduce these bugs, we vary the image content in training and test data, around class-specific features (e.g. bird appearance), and less specific features (e.g. background).

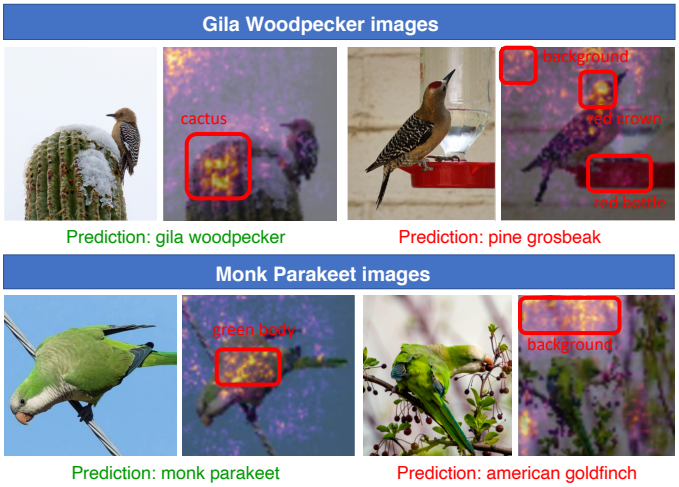


Figure 10.4: Examples of implicit (green) and explicit (red) failures caused by irrelevant and incomplete features, e.g. the model incorrectly uses the pixels corresponding to the `cactus` to correctly predict the `gila woodpecker` class in the left image. The bounding boxes show the features of the model. We create the first failures by making sure that `cacti` are present in all and only training samples of `gila woodpeckers`, while the test images do not all contain a `cactus`. The second ones are created by making sure that only the `monk parakeet` training samples present a `green` bird (and in a standard position), while the test samples are more diverse.

Table 10.4: Bugs introduced in the models of our design brief.

Bug	Description	Example	Creation method
Distrib. shift	Large difference in training and deployment images.	We mention that training data are scraped from the Web, and a different context for the deployment data.	
Simplistic/incomplete, relevant features			
Explicit failure	The features are relevant but incomplete, and lead to incorrect inferences.	The model learns the `red` color for the `pine grosbeak`, which is correct for the males, but not for the yellow females.	We choose a subset of training images (e.g. male images) that give a partial view of the entire class.
Implicit failure	The model learns features that are relevant, but insufficiently representing a class, while still allowing for correct inferences.	The `monk parakeet` class is identified by the model solely through the color `green`.	We choose classes so that certain have a unique feature compared to the others.
Spurious/irrelevant features			
Explicit failure	The model learns features that are not semantically related to the species, and lead to incorrect inferences.	The model recognizes `gila woodpecker` by identifying `cactus` in images, but there is not always a cactus in the image.	Training images of a class contain an irrelevant feature, absent from other training samples and test set.
Implicit failure	The model learns irrelevant features, but still makes correct inferences.	The model learns the presence of `water` to identify `hooded mergansers`.	Same as above, but with similar training and test sets.

10.6. RESULTS

In this section, we present the results of our user-study, essentially the impact that the explanations in the probe have on the bug identification process, and how they are used in this process.

10.6.1. IMPACT OF EXPLANATIONS ON THE BUG IDENTIFICATION PROCESS

Figure 10.5 summarizes the number of bugs identified by the participants in relation to the different types of model failures we introduced. We count one issue (1 point) as completely identified when a participant identifies both a bug and a relevant correction method, and give 0.5 point when the bug is well-characterized but no relevant correction method is found. This way, we make sure that the bug is characterized well-enough for the participant to propose a meaningful bug correction solution[2].

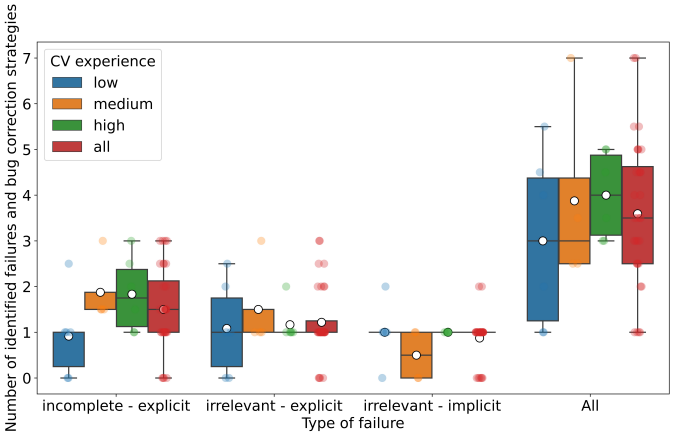


Figure 10.5: Number of bugs and relevant correction methods identified by our different participants during the user study.

SUCCESSFUL BUG IDENTIFICATION

The bug identification process of our participants was in majority successful, with 3.5 bugs and correction methods identified on average, and up to 7 bugs identified by experienced participants. For consistency, we first let the participants explore the probe and failures they deemed important, and later discussed four specific failures. They were typically able to reflect on these failures, but not at the same speed, explaining the large standard deviations. Besides rapidity, three factors explain such deviation. 1) The low-CV participants deemed certain low-rate failures not worthy to debug due to their rarity. This can yet be wrong as high-CV participants discussed, since the error might be rare due to the data distribution, but still harmful. 2) The rare failure (one single lesser

[2]We do not plot the numbers related to implicit, incomplete features because they are identical to the ones for implicit irrelevant features: participants who succeed in identifying the latter always mention that the features are irrelevant and by extension incomplete –other ones should have been used.

`goldfinch` mis-classified as a `hairy woodpecker`) was challenging, and only two high-CV participants proposed plausible bugs. The others pointed out to the lack of additional examples of this failure, preventing them from comparing local explanations. 3) The participants did not think of the existence of implicit failures, except when nudged.

Overall, these results show that a probe presenting various types of explanations allows to debug various feature failures, in relation to various dataset bugs. In order to achieve such successful bug identification, participants used varied workflows to navigate the different functionalities. These workflows are discussed in the next subsections.

DISPARATE RESULTS FOR DIFFERENT EXPLANATION AUDIENCES

Among these successful results, we observe a high disparity in the number of bugs identified between participants with different levels of experience in computer vision (CV).

Low-CV experience participants miss guidance. In general, participants with computer vision experience identify more bugs than the ones without experience. The participants without experience who identified zero or one bug did not know where to start the process, how to proceed, and what kind of corrections to envision.

Misaligned mental models. Yet, three high-CV participants (removed from the plots) identified less than two bugs. Their mental model of bug identification was not aligned with our probe. They did not want to look into model features for bug identification, and one was solely interested in unknown unknowns [938, 52, 506] (outside the probe scope).

EXPLAINABILITY ALLOWS TO ENVISION VARIOUS, RELEVANT BUG CORRECTION METHODS

The probe led the participants to formulate bug correction methods that are diverse, relevant, and to-the-point, thanks to the different kinds of explanations that allowed the identification of highly specific data bugs. For instance, three participants discussed inappropriate data processing as a cause of failure, e.g. the image resolution is too small or the bird/background ratio too large, making the differences between certain bird species undetectable, suggesting for transforming the data pipeline. Five participants suggested restructuring the inference task by adding more classes, as a result of better characterizing the source of bugs, e.g. they identified the color differences between male (red) and female (yellow) `pinegrosbeaks` leading to high error rates for the female ones (confused with the yellow `american goldfinch`), and suggested to separate them into two classes to ease the learning. This is in line with other bug identification frameworks [763] which report they support idea generation for bug correction. Particularly, we notice these envisioned correction methods are more precise and potentially more effective than in our formative study where few types of explanations were mentioned.

PARTICIPANTS STILL MISSED CERTAIN BUGS

Incorrect features vs. correct inference. Participants focused on failures visible through the confusion matrix, either when a percentage of a diagonal cell is low, or when out-of-diagonal cell percentages are high. They often forgot that even classes with high accuracy might be based on problematic features. Some participants identified these issues serendipitously when attempting to understand visible failures.

Confirmation bias. One participant identified a very general bug from a few images: color bias of the model for most species. Confirmation bias led them to validate this bug

by looking at images of different species, without going in more detail into the problematic colors and species, or searching for other bugs. We discuss these results further in section 10.7.

10.6.2. DIFFERENT CATEGORIES OF EXPLANATIONS FOR DIFFERENT USERS AND BUG IDENTIFICATION STEPS

Figure 10.6 displays the perceived usefulness of each tab as rated by our participants. Overall, all tabs are perceived useful with an average rating of at least 4 out of 7, yet the mean rating and standard deviation vary across tabs. We discuss below these variations in relation to the functionalities provided by each tab.

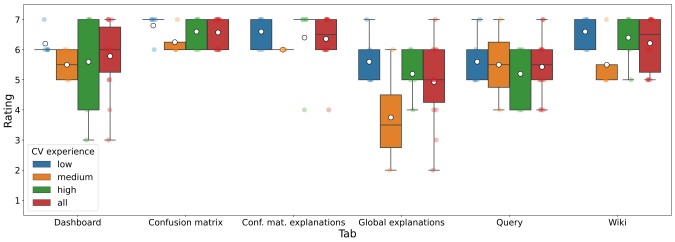


Figure 10.6: Perceived usefulness of the different tabs of the design probe. The ratings are displayed for each category of participants.

LOCAL VERSUS GLOBAL EXPLANATIONS (F3, F4)

Hypothesis validation emphasized with the diversity of explanations. The participants primarily used the local and global explanations from the dashboard and confusion matrix, as testify the higher ratings for these two tabs (Figure 10.6). These explanations served for generating bug hypotheses and validating them. This shows that proposing a diversity of explanations nudges a more extensive bug identification process than with fewer or no explanations where most participants skip hypothesis validation, as we observed in the formative study.

Participants investigated explicit failures by entering different cells of the confusion matrix. The implicit failures required more diverse entry points to be identified: 1) Serendipitously, while investigating explicit failures. While investigating an explicit failure by looking at the four columns of images/saliency maps in the tab obtained from clicking on a cell of the confusion matrix, they would notice that saliency maps would highlight irrelevant features even in the A-A or B-B columns that present samples with correct predictions. 2) By deciding to explore the global explanation tab (without having a specific kind of failure in mind) and spotting clearly surprising features (e.g. `water` or `branch` are not features one would expect the model to focus on when classifying bird images. Instead, parts and characteristic colors of the bird would be expected for the model to generalize to new images with more varied background for instance) for the context, or features that were not sufficient (e.g. `purple` only for the `mandarin duck`, whereas images displaying other birds might also have the purple color, e.g. in their background). 3) By deciding to look into the diagonal of the confusion matrix (typically starting with cells that have low rates) and the corresponding explanations (saliency

maps, and rankings of textual explanations by frequency). By looking into these specific features, they would reflect on whether something is irrelevant or incomplete.

Local and global explanations are complementary. The choice of starting point does not have a consistent motivation. Typically, participants who use local explanations to generate feature hypotheses validate their hypotheses by looking at local explanations for more images, or by verifying the presence of the features in the global explanations. Instead, participants using global explanations for hypothesis generation validate the features by making sure these features are reflected in a few local explanations across correct or incorrect inferences.

Within hypothesis generation, many participants combined the two approaches as the types of features and correction methods they lead to identify intersect but do not entirely overlap. For instance, incomplete or irrelevant but frequent features were typically identified from global explanations through the different ranking systems (**F6**). Instead, infrequent failures and their correction methods were better understood by looking at the actual images and saliency maps (**F5**). Global explanations were also used to identify the features influencing the majority of classification (typically the correct ones), which are in turn compared to the features used for incorrect inferences identified through local explanations (**F5**: comparisons across explanation types). For instance, they identified that overall, the color red is used by the model to infer `pine grosbeaks`, and locally understood that the only `american goldfinch` predicted as `pine grosbeak` was also displaying a `red` feature due to the brightness of the picture. Such finding could not be reached through a sole look at global explanations for which brightness is not reflected.

The choice of explanation type depends on the developers' experience with explainability. We do not identify a strong correlation between the categories of explanations used by the participants and their expertise. Yet, most participants with high-CV experience are more reticent towards unfamiliar types of explanations, and use primarily local, visual explanations, i.e. saliency maps (*"the dashboard gives almost everything. I'm more familiar with its explanations" Participant 4 high-CV*). Instead, the participants with fewer experience operate smoother transitions between local and global ones, and explore more types of explanations. This explains the higher ratings they gave to the tab reached from the confusion matrix (that presents all types of explanations) compared to the dashboard that only presents local explanations. Using global explanations can be faster than using local ones, but it was also more tedious as participants need to get accustomed to the scores and ways to interpret them. All participants argued these explanations should be used particularly when many images present similar failures, as it is not tractable to look at each image.

The use of local and global explanations led to incorrect bugs. Two types of errors are typically done when using the local explanations for hypothesis generation. a) Participants wrongly assumed the local explanations for images that got correct inferences to be relevant features for the model. This led them to automatically judge as irrelevant the features of samples with wrong inferences, while this is not necessarily the case. Warning about this assumption enabled them to reflect further about the potential bugs. b) Some participants formulated an incorrect hypothesis about a feature by looking at very few images, and did not further verify it, leading to develop incorrect bug correction methods. They mentioned that the global explanations could allow to avoid such er-

rors. Global explanations were misleading when participants would identify interesting features with very low support, not being representative of most images.

EXPLANATION DOMAIN AND MEDIUM (**O2**)

Participants intuitively prefer in-domain explanations. All but one participant preferred using visual explanations than textual ones. They argued the cognitive load is lower and it is faster to make sense of features by glossing over several local, visual explanations, than textual ones.

The two types of medium are complementary locally. Yet, textual explanations were also used. The participants mentioned that since they are not familiar with the task domain, they cannot easily interpret the saliency maps to identify meaningful features. Hence, they look at the local, textual explanations (and map them to the visual ones) to identify relevant bird features that one would expect the model to learn. They could also directly relate the wiki information that displays expected features according to an expert to these explanations. One participant also suggested a functionality that only textual explanations support: giving the freedom to explore new features as combinations of existing ones, to vary their granularity and create a taxonomy, e.g. combining `plants` and `leaves` into a larger `green background`. While this is possible within the query page, they would have liked to access this faster within the other interfaces, and to visualise the created taxonomy.

The preferred, global medium depends on the familiarity with the task domain. Participants mentioned a difficulty in interpreting the textual, global explanations as they were not familiar with the domain of the task. They however said that if they would know more about the domain, it would be easier to use as they could quickly get an idea of what a feature means on an image and what might be problematic with it.

EXPLANATION SCOPE (**O1**)

Preference for explanations of binaries. Participants primarily focused on two-class explanations. These explanations align with reactive bug identification for failures in specific cells of the confusion matrix. Reflecting on two classes is also easier than considering more classes: it is harder to relate overall explanations to model failures. Half of the participants explained that overall explanations are also useful but less natural as they start from the out-of-diagonal cells of the matrix. This shows clearly in the ratings given to the dashboard or confusion matrix tabs that provide binary explanations, in comparison to the ratings for the global-explanations tab.

Global explanations as a quick diagnostic tool. Yet, participants still find uses to the global explanations computed on the entire dataset, as the large standard deviation testifies in comparison to the standard deviations for the other tabs. Participants prefer using such global explanations for tasks whose domain is familiar, and for diagonal cells of the confusion matrix. Simply by looking at these lists of explanations without having to click on each cell of a confusion matrix, they get a good overview of the features the model has learned per class, and can identify the pertinent, irrelevant or incomplete ones. Five participants actually used these explanations and their background knowledge to reflect on the validity of the features, e.g. they quickly spotted potential issues with `cactus` or `water` concepts that one might not expect to classify birds, and with the large number of color features while the model should also relate on shapes.

Questioning the faithfulness of binary explanations. These explanations are complementary. Global explanations more accurately account for the features of the model and allow for a faster spotting of problematic features. Yet, developers prefer to understand specific cells of the confusion matrix with binary explanations, which might lead to erroneous feature interpretations (one feature might seem discriminative for two classes but might not be important overall). A single mid-CV participant accurately reflected on such limitation, that developers should be warned about. However, this reflection was also problematic for our participant as it prevented them from obtaining insights from the probe: the participant constantly worried that correcting a specific bug would create new ones in other matrix cells.

USE OF DOMAIN KNOWLEDGE (**F9**)
Domain knowledge is used for successful hypothesis formulation and validation. This knowledge serves to a) formulate hypothesis on relevant features the model should learn for a class, and to compare them to the actual features, or b) to validate hypotheses about problematic features. For instance, *Participant 4 high-CV* naturally started to use it for specific confusion cases where the model accurately looks at the bird (according to the saliency map) but apparently not at the right or complete bird features as it makes incorrect inferences.

QUERY-RELATED EXPLANATIONS (**F7**)
All participants used the passive mode of exploring the explanations, since it is less cognitively-demanding, and they are used to such explanations. Active querying is used only by half of the participants. This shows clearly by the lower average ratings and higher standard deviation the query tab got in comparison to the dashboard and confusion matrix tabs. Active querying allows to validate potential hypotheses around problematic features. For that, participants query the matrix of percentage to verify that a feature is only used for images that present specific miss-classifications. Three participants mentioned that active queries are especially efficient once one is familiar with the expected and the often problematic features. For instance, an expert participant mentioned that in their own medical use-case where it is known that the model might learn incorrect features relating to the background of X-ray images (e.g. a part of a pacemaker), they would like to query background features directly.

INTERACTIVITY (**F8**)
Interactivity to speed up and augment the bug identification process. Besides having functionalities that are currently not available (global explanations and query), the primary advantage of the probe was its interactivity and practicality, aligning with results for tabular data [365]. It was especially useful to compare diverse images (the four types of images in a binary classification task) and explanations to estimate the feature's relative importance. For instance, some participants compared two queries where the only difference is the addition of one feature, to check how much this feature impacts the model inferences. *"If the tool is ergonomic, fast and malleable, it would definitely help me fasten my process, and it would help combine more information that I don't usually look at." Participant 8 high-CV.* A third of the participants even suggested ways to have even more interactivity and fast transitions between explanation types.

Interactivity to select relevant explanations. To navigate global explanations, the participants used one main interactivity feature, the choice of settings, to rank or filter explanations (**F6**). They could identify a) frequent mistakes by ranking the explanations based on the number of incorrect predictions they lead to, b) frequent features by ranking explanations based on typicality scores and filtering out low-support ones, and c) features that lead solely to correct or incorrect inferences by computing the explanations independently on the set of samples which received correct or incorrect predictions. These settings are necessary due to the amount of information the probe provides.

10.7. DISCUSSION

10.7.1. SUMMARY OF FINDINGS

Our user-study brought new insights on the use of explanations towards bug identification, summarized in Table 10.5. While the most common explanations, i.e. local visual explanations, were primarily used due to their simplicity and familiarity, our probe also highlighted the importance to present diverse explanations. Global, textual, active, interactive, and binary explanations, as well as domain knowledge, were also exploited to achieve different objectives, e.g. identifying new hypotheses, or the same objectives more efficiently. Yet, by acknowledging the disparity in the use of the functionalities and in the number of bugs they led to identify, we can extract further implications for future explainability, debugging and HCI research. We now discuss the limitations of our work and these findings.

Table 10.5: Summary of the insights from our user-studies.

Category	Insight
Impact of explanations on the debugging process.	
Effectiveness	Successful bug identification process. A few missed/incorrect bugs due to misinterpretations of features and confirmation bias.
Variations	Low-CV: need for guidance. High-CV: misaligned mental models.
Corrections	More diverse and precise bug correction methods are envisioned.
Different categories of explanations for different users and debugging steps.	
Local/global	Complementarity. Emphasis on hypothesis validation. Preference based on developers' experience with explainability.
Domain, medium	Intuitive preference for in-domain explanations. Local level: complementarity of in- and out of- domain explanations. Global level: preference depends on the familiarity with the task domain.
Scope	Preference for binaries. Global explanations as quick diagnostic tool. Lack of questioning around the faithfulness of binaries.
Knowledge	Domain knowledge used both for successful hypothesis formulation and validation.
Active query	Low use despite usefulness for hypothesis validation.
Interactivity	Speeding up and augmenting the debugging process. Selecting relevant explanations. Wish for model comparisons.

10.7.2. LIMITATIONS

There are several limitations in our probe and study. While we do not think they impact the validity of our results, they would need to be tackled in the future for more compre-

hensiveness.

Scope of the probe. Our probe is adapted for a specific type of computer vision models: deep learning models that perform classification tasks and from which local, visual explanations can be extracted. The global explanations can only be computed when it is possible to annotate local explanations with semantic features, and can be costly depending on the size of the dataset and number of classes. Hence, it can be challenging to use for certain applications. Adapting these explanations to other use-cases is a challenge on its own. Balancing the trade-off between cost and faithfulness of the explanations and making developers aware of it would also merit being investigated.

Scope of the study. While the work involved a considerable number of participants (18 for the formative study and co-creation sessions, and 18 for the user-study) with various backgrounds, we cannot fully guarantee the generalizability of the results. Similarly, our study employed a use-case that requires domain knowledge none of our participants had (to bring consistency), and we made sure to provide the required knowledge. It would be interesting to study how participants, familiar with a use-case, would go about bug identification. This is however challenging as participants should share their data, it is costly to annotate, and the use-case would not be consistent across participants. Scaling our study to use-cases with more classes is also important as other works identify that "as scale increases, interpretability and satisfaction decrease" [365].

Impact of the probe design. The results of our user-study are inevitably mediated by the design, implementation and usability of our probe. As discussed in section 10.4, we however made sure to allow for diverse workflows and interactions with the explanations without biasing the users towards specific ones. As for usability, the answers (Figure 10.7) to our exit questionnaire give an indication of how it might have affected our results. Most factors received high-ratings, confirming that our participants appreciated the functionalities within our probe, and were likely not negatively impacted by them. Especially, they found it rewarding to use our probe and were eager to reuse it on their own use-cases, saying that it was more convenient than their usual development environment.

However, some participants felt overwhelmed at first by the amount of functionalities (perceived usability ratings confirm this). While they got used to them, they would have liked guidance from the probe in the process. We could not do so not to skew them, yet this is an important indication for future tools. Similarly, they gave an average rating to the attractiveness (mean of 3.31 out of 5 points) as they would appreciate the probe having a more modern look.

10.7.3. IMPLICATIONS & FUTURE WORK

NEED TO DEVELOP USER-EXPERIENCES

Guidance. As some participants had a hard time envisioning uses of certain explanations, future tools need to provide hints. Hints should be enough as simply explaining ways other participants used the explanations led the participants in difficulty to successfully identify bugs.

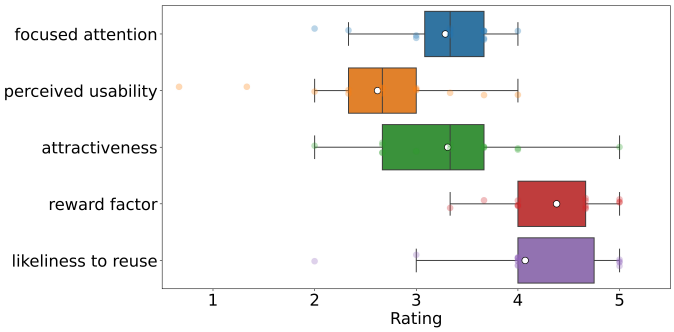


Figure 10.7: Aggregated factors of the User-Engagement form (short version) presented in boxplots.

Besides, the current probe allows for any sequence of interactions with the different types of explanations supported (in order not to skew our user-study participants towards certain explanations and activities). Yet, further guiding these interactions by suggesting potential sequences of activities would also support developers further in debugging their model. Several participants mentioned that an ideal user-interface would not leave them as much freedom as currently is, but instead narrow down possibilities so as to simplify the debugging process and guide them towards the relevant activities for each type of failures and bugs. Hence, future tools would benefit from identifying the minimum set of user-journeys different types of developers and failures would require.

Participants with high-CV and explainability experience however require further investigation to understand when they would be ready to use less familiar explanations. Especially, for these participants, our observations differ from explanation practices on tabular data [365]. The GAMUT probe led to find a strong correlation between the level of explainability expertise and the use of diverse explanations, result totally opposed to ours. This could be motivated by the lack of practice, even for our high-expertise developers, with global, textual explanations for computer vision, contrary to developers working on tabular data who are more familiar with both types of explanations.

Warnings around typical misinterpretations. Blindly following the explanations sometimes leads to identify incorrect bugs. Yet, not all participants are aware of these dangers, and trust the explanations similarly. Only two participants asked us how the saliency maps were computed, and none reflected on the potential noise in the salient pixels. As for the global explanations, only 4 participants questioned their faithfulness and the fact that an annotation of salient pixels does not necessarily reflect what the model actually looks at (i.e. colors, textures, or shapes, etc.).

These observations around trust in explanations are aligned with the ones for tabular data, e.g. Hohman et al. [365] mention needing "healthy skepticism" from developers. They are also inline with the notion of *misuse* of explanations [427]: certain participants would misinterpret explanations by taking a brief look at them simply because they seemed to confirm their hypothesis. Future tools would merit displaying warnings against these limitations and misinterpretations.

Integration of structural and training bugs. Some participants tended to explain all bugs with issues of data content or data pipeline, without elaborating on other potential bugs, e.g. related to the model structure or training hyperparameters. They were either skewed by the focus of our probe on such types of bugs due to the visualisation of data content, or because they did not have in mind the other concerns. Some participants envisioned to use our probe once other bugs are corrected, but others nuanced this view arguing for a more iterative process, where all types of bugs might need simultaneous considerations depending on the ways the bugs are corrected (e.g. data augmentation for balancing might lead to overfitting and to increase the size of the model architecture). This shows the need to investigate how to best combine the functionalities in our probe to the functionalities around the other types of bugs (e.g. tools such as [728]), without overwhelming a user.

USEFULNESS OF DIFFERENT EXPLANATION TYPES

Explanations for data enquiry. Explanations primarily served as artifacts for surfacing feature failures, identifying data bugs and bias-variance issues. Similarly to observations made for explainability with tabular data [365], the explanations were also used by four of the participants as an access point into the data. These participants used the query functionality with specific features, ground truth and predicted classes, to better understand what they look like within the dataset, and whether they are comprehensively represented. Such understanding was later used to refine hypotheses about dataset bugs. Future interfaces would hence merit combining further the extensive exploration of training datasets to the model exploration, and facilitating common interactions with the explanations towards that end.

Complementarity of explanation types. Our study showed that all explanation types are useful for participants in different stages of the bug identification workflow to answer different questions. Their use often depends on the degree of familiarity of the participants with the task domain, and with these types of explanations.

More research is needed to further develop these different types of explanations since, so far, research focuses primarily on local, visual ones. Especially, attention on *textual explanations* could benefit developers, e.g. for understanding how to best represent and query concepts and their combinations, taxonomies of concepts, etc. *What-if (causal) questions* that were rarely expressed here could merit research on accessibility as well. Finally, future tools could further *combine in- and out of- domain explanations* by showing example image patches corresponding to any displayed textual explanations so as to increase the learning rate of the developers. Two participants also suggested *global, visual explanations* by automatically clustering similar-looking, salient image patches. While this might be hard to realize in practice, this further shows their appreciation for visual information, and the need for further research.

Interactivity versus complexity. Surprisingly, our study showed that rather simple explanations can lead to successfully understand a large number of bugs: the global explanations were simple statistics computed over textual annotations of the dataset, but allowed for a global understanding of the model. While simple in their calculation, their

interpretation was already complex enough for the participants not familiar with the textual and global explainability paradigms. We argue that these simple explanations were useful thanks to the usability of the interactive interface and its focus on comparisons, which allowed to identify many similarities and differences across images receiving different predictions.

This shows that it might not be urgent to develop highly complex explainability methods yet, as they are new black-boxes for the developers who might trust their faithfulness too much, while having a hard time using them. Instead, more research on the development of interactive interfaces could be more beneficial to the developers.

Manual exploration versus automation. Multiple participants suggested to automate parts of the interface to speed up and direct the debugging process. For instance, they would like to automatically be presented with explanations that reflect bugs, or at least with a reduced set of potentially problematic features (the number of global explanations is otherwise overwhelming) through an automatic comparison of the explanations to the domain-knowledge.

Yet, we argue that extensive automation is not possible and desired. The relevance of a feature to a model is sometimes ambiguous, e.g. relevance of the `cactus` for `gila woodpeckers`, so the automatic comparison would lead to a skewed and non-transparent result. Besides, attention should be put into not making the debugging tool another black-box (besides the model to explain), as our participants already tended not to question the completeness and faithfulness of the displayed explanations. A way to limit automation could be to provide even quicker interactions, for instance to go from binary explanations to global ones so as to accurately estimate their relevance.

Nevertheless, facing the amount of debugging methods developed in machine learning testing literature that are unknown to developers, it is important to also investigate how much these methods are complementary to the manual process, and how to best involve them in this process.

Reliance on domain knowledge. The study confirmed the importance of domain knowledge. All but one participant (who did not reflect on features) used it (hence the high ratings the wiki tab obtained). Unfortunately, investigating the wiki was not consistently performed across failures, leading to miss certain bugs. For instance, two participants who correctly understood the difference between the similar-looking bird species `gila` and `hairy woodpeckers` (`brown or white body` respectively) and the missing feature (body color), did not use the wiki page to inspect the `pine grosbeak` and `american goldfinch`, missing the hint for another bug (difference of colors for female and male grosbeaks). Using domain knowledge merits more support. Studying how to make developers and domain experts interact is important, i.e. the format in which they can best communicate, the inputs developers need, but also the most effective way for domain experts who are often not familiar with technical terms to provide useful information for the developers.

10.8. CONCLUSION

In this chapter, we engaged in a formative study and a co-creation process to design a probe for investigating the interaction between explanations of learned and expected mechanisms of a model, and bug identification when developers aim at diagnosing failures of their models. We then performed 18 user-studies with this probe. Our participants varied in their bug identification workflows, but managed to identify a consequential amount of bugs. These results showed that explanations of learned and expected model mechanisms can be used in various steps of the process for different purposes, and especially for characterizing diverse types of feature failures. Different categories of explanations (e.g., global, out-of-domain, active, and interactive) showed to be useful and often complementary. Yet, our participants also struggled with various aspects of the process, falling into certain explainability traps, or being shy to explore unfamiliar explanations.

This shows the urgent need for more HCI research to provide the right amount of guidance to developers engaged in bug identification activities and having access to explainability methods, while still allowing for freedom and adaptability of the process. Especially, the process should be supported through the use of interactive interfaces with various types of interactions not only with data and explanations but also with other artifacts to address non-feature failures. Additionally, our study points out to research directions for other communities: specific types of explanations merit further development by the machine learning explainability community, and the effectiveness of machine learning testing methods needs to be characterized in comparison to the one of human debugging for future integration.

APPENDIX

EXAMPLE MOCK-UPS USED IN THE CO-CREATION SESSIONS

Figure 10.8, Figure 10.9, Figure 10.10.

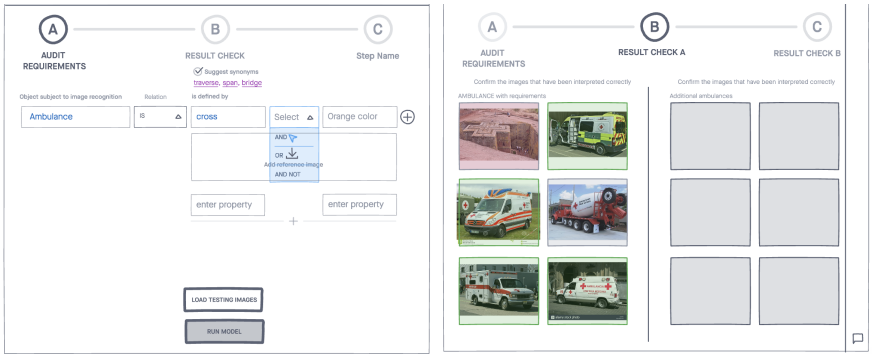


Figure 10.8: Low-fidelity mock-up used in the co-creation sessions: query functionality and the result interface after a query.

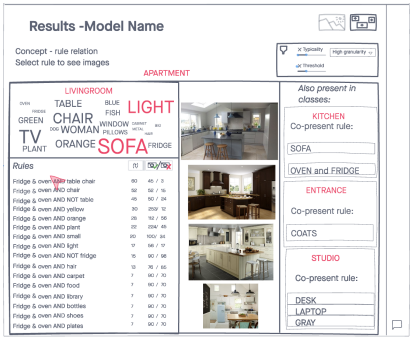


Figure 10.9: Low-fidelity mock-up used in the co-creation sessions: example display of important concepts and rules for one class, and their co-presence in other classes.

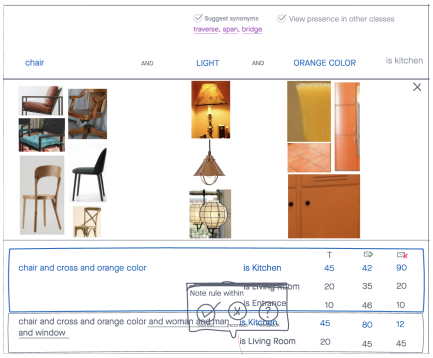


Figure 10.10: Low-fidelity mock-up used in the co-creation sessions: another example display of important rules and scores, in comparison to the scores of related rules for other classes.

FIRST IMPLEMENTED PROTOTYPE FOR THE PROBE

Figure 10.11, Figure 10.12, Figure 10.13, Figure 10.14, Figure 10.15, Figure 10.16.

Figure 10.11: Display of the saliency maps within the probe.

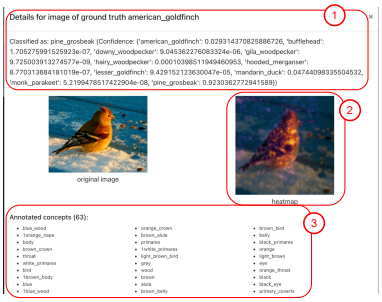


Figure 10.12: Display of further local explanations.

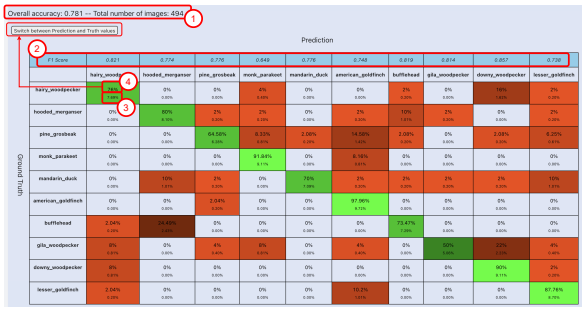


Figure 10.13: Overall performance information provided in the design probe.

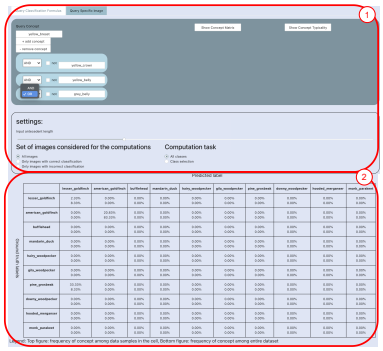


Figure 10.14: Query page with (1) query input and (2) query results.

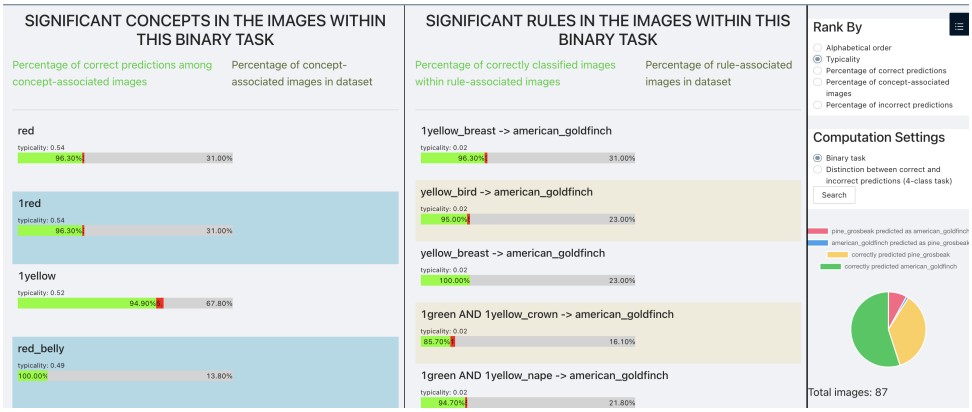


Figure 10.15: Local explanations presented as a result of clicking on a confusion matrix cell.

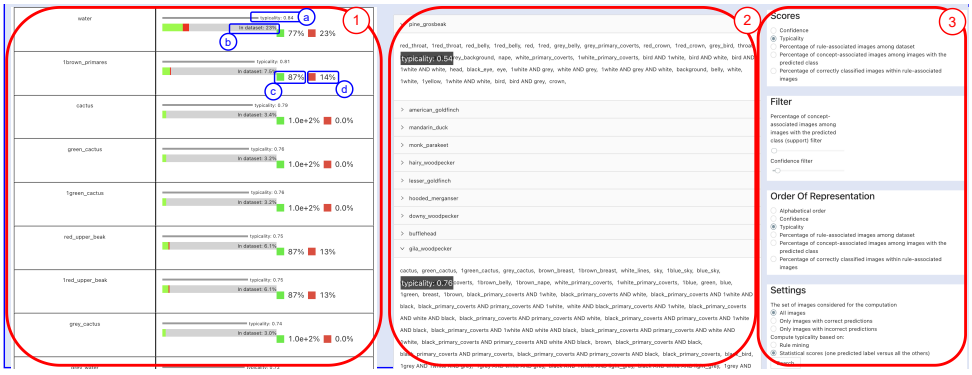


Figure 10.16: Display of global explanations within our probe. (1) shows the overall explanations, (2) shows the class-specific explanations, (3) shows the settings that can be tuned to compute the explanations. In (1), we show the global explanations displayed within the interface: (a) shows the typicality score, (b) the frequency of times the concept (or rule) is salient within the dataset, (c) the percentage of times when the image where the concept is salient got a correct inference, (d) and conversely when it got an incorrect inference.

11

Conclusion & Discussion

Machine learning (ML) is a technology that shows promises in various sectors[1], such as agriculture, manufacturing, healthcare, and transport. Yet, similarly to every other technology, its existence is not without potential harm to society [546]. What makes ML unique however is its socio-technical nature [193, 230]. Where most technologies might cause obvious physical harms due to errors happening in the technological systems (e.g., a plane crash due to a faulty engine), in interaction with their users (e.g., a plane crash due to a wrong command of the pilot), or due to the goal of the technology itself (e.g., the nuclear bomb), ML, besides those physical harms and this set of causes, might induce surreptitious social harms.

Naturally, it is arduous to develop any other technology, and to make it non-hazardous (when desired) because of their technical complexity and the required scientific understanding of the natural phenomena whose control allow for the technologies to develop (let alone the complexity of the user interactions with complex human factors). ML does not escape this technical complexity, as its foundation is a plethora of complex optimization algorithms.

Additionally to that, we see ML as even more challenging, especially in the context of automated decision making systems, because of the impossible separation of concerns between the technical complexity and the social considerations in the design of the technology before even imagining deploying it for practical use [230]. Indeed, the optimization algorithms always encode human values [312]. As it is impossible to avoid errors in the outputs of these algorithms, one, intentionally or unintentionally, encodes a specific set of human values and trigger social harms for certain categories of the population when designing a ML algorithm [442, 182]. Next to the technical complexity of designing a ML algorithm without any social considerations in mind, it is also challenging to identify and characterize the potential social harms of a ML system (as any social harms also not coming from this technology), and to decide on a "fair" distribution of this harm across a population (e.g., it involves considerations from other fields such as political philosophy). It becomes even more challenging to bridge the connection between the social considerations and the technical design, both to technically represent the harm accurately and to tune the algorithm to avoid this harm or allocate it according to the defined, desired, fair distribution. It is a constant negotiation between the current and envisioned technical capabilities of the algorithms, and the desired, potentially ambiguous, and certainly subjective, social considerations to encode in them [619].

In this thesis, we have directed our efforts towards several types of contributions, with the aim of supporting the design of less hazardous and harmful systems relying on ML technology, accounting for this constant negotiation between social and technical considerations. In this concluding chapter, we first summarize our contributions, and then reflect on their implications for future work.

11

[1] https://www.fortunebusinessinsights.com/machine-learning-market-102226

11.1. SUMMARY OF ANSWERS TO OUR RESEARCH QUESTIONS

We have tackled the design of less hazardous (and consequentially less harmful) ML-based systems via three main angles: 1) the current state of the research characterizing and mitigating ML hazards and harms and main future research directions (Part I); 2) the practices, challenges, and needs, of the developers who design and build ML-based systems (Part II); 3) potential solutions to the developers' needs, i.e., the development of technical and design solutions to certain of the identified challenges and the proposition of broader re-directions of the research for other challenges (Part III). Each of the three angles is addressed by an individual part of this thesis.

11.1.1. CHARACTERIZING THE STATE-OF-THE-ART RESEARCH (PART I)

To address **RQ1, RQ2**, and **RQ3**, we conducted four rigorous surveys of the literature. We first investigated the ML harms and causes of harms envisioned by the research community [70, 62]. We then delved deeper into the technical solutions developed to identify and mitigate these harms within a ML systems, especially focusing on the realms of ML robustness [825] and ML fairness [66]. We finally looked into critical reflections around the limitations of these (primarily) technical solutions to address the harms [62].

RQ1: We found that an ML lifecycle is made of multiple stages, and that all these stages are potential sources of harms as well as the loci to mitigate the consequent harms. (Chapter 2)

RQ2, RQ3: We observed that the ML community has primarily focused on a few of these stages, those around the algorithm (feature engineering, model training, post-processing, testing) and the data processing aspects (data transformations or augmentation). The rest of the technical stages revealed to be neglected, such as the creation and cleaning of data, and the socio-technical stages as well, such as the establishment of requirements and specifications for an ML system, the validation of the model, or the design of the stakeholder interactions with the systems. A number of limitations in the current technical approaches appeared, in terms of their effectiveness towards their stated goals, their usability for developers in organisations, and their conceptual inability to comprehensively represent all harms of ML. Hence, we identified the need for technical and socio-technical research to better translate the harms into mathematical formulations (if that is even possible) and optimize against these harms, to build systems that are more cognizant of these harms, to support developers in building less hazardous models using the techniques stemming from research, and finally to continue identifying new types of harms and sources of harms in those ML systems and organisations that have not been investigated yet. Finally, we also noted the lack of research about ML developers' practices around questions of robustness and harms broader than fairness. (Chapter 3, Chapter 4)

11.1.2. ANALYSING THE RESEARCH/PRACTICE GAP (PART II)

To address **RQ4**, **RQ5**, and **RQ6**, we conducted interviews with 50 ML developers. We designed the interview sessions based on our analysis of the literature, especially based on the main research directions that are currently followed in ML and on the insights from other research fields (e.g., software engineering practices, political philosophy and resource allocation), that could apply to ML. We then thoroughly analysed the interview

transcripts to identify the main challenges faced by developers, and to characterize the consequent misalignment between research and practice.

RQ4: We found that ML developers all adopt very different goals and approaches when designing, implementing, and evaluating ML-based systems [68]. In terms of goals, they do not all consider the same types of failures and bugs as important to solve, and do not adopt the same reasoning to judge on the importance of each issue and to decide when their system is ready for deployment. In terms of approaches, their workflows vary, with more or less rigorous experimentation to test the goodness of their systems, to proactively identify issues in their systems, or to mitigate identified issues. They do not all use the same methods, tools, and artifacts to do so, with some developers having experience with explainability methods or fairness toolkits, while others do not have. (Chapter 5)

RQ5: Naturally, when analysing further these practices in terms of harms the resulting systems might cause, we found a set of bad practices, that should be changed in the future, and that were surprisingly not always coming from the least experienced developers but also from more experienced ones [61]. For instance, certain developers did not envision any harm of their systems; other developers envisioned vague unfairness issues and operated arbitrary trade-offs between arbitrarily selected fairness metrics to declare a system acceptable; while other developers unintentionally silenced certain populations from their systems by removing any data pertaining to them, leading to potentially unsafe system predictions for these populations. (Chapter 6)

RQ6: Delving deeper into the reasons for these differences and potential flawed practices, we identified a multitude of relevant, intertwined factors [71]. As expected, we found a lack of technical research to support developers in debugging or optimizing their ML models based on the constraints they have, e.g., optimizing for output accuracy and fairness while not having access to a large number of informative training data. We also identified that several research directions could be useful to the developers, but are not known or usable by them, e.g., the explanations from explainability methods are sometimes challenging to interpret for them, hindering their debugging process. We also found many human factors, e.g., around past interpersonal or ML related experiences, cultural background, education in computer science or other fields, education in ML or ML fairness, and access to supportive tools or not, that impact the practices, the perceptions of harms, and the approaches to tackle them. Finally, organisational factors, e.g., business incentives, responsibility allocation, also impact the practices. These findings call for efforts in various directions in the future, be it technical and design research to support developers further with convenient tools, human factor research to better understand how to personalise these tools for individual obstacles but also to propose broader educational support, and structural changes, e.g., in terms of regulations, education programs, or internal organisation functioning and principles. (Chapter 7)

11.1.3. PROPOSING SOLUTIONS (PART III)

In the last part (Part III), we decided to address some of the technical and design research challenges identified above. To address **RQ7**, **RQ8**, and **RQ9**, we engaged in the design of several methods and tools, that rely on our findings from the previous two parts. We proposed a method for cost-efficiently extracting post-hoc, semantic, model learned mech-

anism explanations [69]. We designed a game with a purpose to cost-efficiently extract explicit and implicit knowledge humans have, that can serve to judge the relevance of a model's learned mechanisms [63, 64]. We co-created a user-interface, that gathers a diversity of information relevant for debugging a model, among which various types of model mechanism explanations [67]. Finally, we conducted empirical quantitative studies, and qualitative, user-studies in order to investigate the correctness, informativeness, cost-efficiency, utility, and usability of these different artifacts [67].

RQ7: We showed that, while explainability methods are often not easily interpretable by ML developers, we can enhance them a human-in-the-loop framework, that increases their informativeness and interpretability while remaining cost-efficient. (Chapter 8)

RQ8: We also showed that it is also possible to leverage the intelligence of the crowd in a cost-efficient way, via a well-designed game, with carefully-crafted input data formats and data processing heuristics, as well as well-selected, personalised-to-the-information-need, data probing the game players. This game allows to collect various types of rich data, with low or high typicality, specificity, and tacitness. All in all, these two works showed that interacting with crowd workers from crowdsourcing platforms can be extremely useful to develop more reliable models, going beyond the typical crowd worker-ML interactions in terms of data sample annotations with ground truth labels. (Chapter 9)

RQ9: Finally, we showed that the two types of information that we collect via the above two contributions, are useful to ML developers in developing and debugging their models. When the developers used our user-interface that combines these different explanations, they engaged in successful debugging activities. This information was used in combination to prior types of explanations by the developers in order to identify various types of failures in their models, to formulate hypotheses around bugs, or to test these hypotheses. This represents a great proof-of-concept, from which to expand to tools even more usable by developers. (Chapter 10)

11.2. IMPLICATIONS & LIMITATIONS OF THIS THESIS

Based on our execution of the works summarized above and the analysis of our findings, we conclude this thesis by discussing the implications of our research for future research efforts, for society, and for various research communities[2]. We especially discuss implications of the results and related research challenges, and reflect on the approach we adopted for this thesis.

11.2.1. IMPLICATIONS OF OUR RESULTS & DIRECT FUTURE WORK

In terms of implications of the results, we first discuss implications for the technical work we performed to provide new theoretical methods, then implications for the practical work performed to make the theoretical methods usable in practice, and finally implications for the socio-technical work we performed to investigate practices.

IMPLICATIONS FOR TECHNICAL WORK DEVELOPING THEORETICAL METHODS

Improving our contributions: Involvement of lay-persons in model diagnosis. Our work (Part III) demonstrated the utility and feasibility of involving humans in different stages of the model debugging process. Previous works around crowdsourcing and ML primarily revolved around ground truth label annotation —how to design crowdsourcing tasks for lay-persons to annotate data samples with their ground truth label, how to identify data samples for which having their ground truth label annotated would allow for the largest increase in accuracy when training a model with it, how to allocate the most appropriate tasks to crowd workers accounting for their expertise—. Yet, our work shows that lay-persons can also be involved effectively and cost-efficiently in other stages of the ML lifecycle. Future work would merit investigating which other stages of the lifecycle, beyond training data annotation and model debugging, would benefit from involving humans.

Besides, more investigation is needed to improve the design of our crowdsourcing tasks to cost-efficiently collect needed data annotations for learned mechanisms, as well as expected mechanisms. For instance, we noted that post-processing efforts are needed to reconcile the annotations from the workers, who might use different vocabulary or granularity of terms to designate related concepts, not allowing for a high-fidelity model debugging. In order to reduce such effort, one could envision controlling further for the crowd workers' input discrepancies by design, e.g., by recommending to them potential relevant vocabulary that was used by previous workers or that comes from relevant taxonomies, by structuring the breadth of vocabulary employed by previous workers into an easily-queryable hierarchy, etc. It would be important while designing these transformations of the current tasks, to keep in mind the need for input flexibility, i.e., flexibility in terms of inputs that the workers can provide. Indeed, in our experiments, crowd workers brought new, relevant, insights that the ML models or us, the authors playing the role of proxy-experts, had not thought of. Hence, serendipity in such crowdsourcing tasks is important to promote.

While our results show that it is not necessary to automate every stage of the ML lifecycle, and humans bring new relevant information to build less hazardous models,

[2] In the next and last section, we will discuss additional future work, beyond the scope of this thesis.

avenues for future work in terms of partial automation could contribute to the cost-efficiency of the process. Facing the amount of available pre-trained models or existing knowledge graphs, one could envision complex workflows where human-annotations would only be performed in cases where it is impossible to obtain high-confidence machine-annotations. Naturally, this would be challenging because of the existence of unknown unknowns, i.e., errors made by the machine with a high-confidence.

Finally, we note that the experiments we conducted all involved simple use-case scenarios that do not require domain expertise (except the bird species classification use-case, where we had to acquire the expertise first). Hence, it was relatively easy to recruit crowd workers from any crowdsourcing platform to conduct our experiments, and it is possible to identify existing pre-trained models or knowledge bases to further automate the task. Yet, it might not be as easy once the use-cases employed require domain expertise. Hence, we pose that research is needed to understand to what extent and how crowd workers can be employed for other use-cases, such as for models that make healthcare diagnostics based on X-Ray images. One could for instance investigate whether it is possible to train the workers to perform the tasks, how to characterize the boundary between feasible and non-feasible tasks, or how to develop new crowdsourcing systems or processes to involve domain experts in a more convenient fashion (it is well-known that domain experts are costly and hard to access due to availability constraints).

Beyond our contributions: Tackling hazards more broadly. Our work revealed a multitude of additional research opportunities. In a similar fashion to active learning, we foresee the need for an *active diagnosis framework*, that would carefully indicate which data samples on which both expected and learned mechanisms would be the most useful to obtain, in order to progress on the diagnosis task in a cost-efficient manner.

Besides, we identified a strong boundary between various related areas of research such as ML fairness, ML robustness, privacy-preserving ML, explainability, etc. We believe it is now important to bring them closer together. One should investigate how to build ML models with multiple requirements and constraints in mind. The fairness literature has investigated trade-offs between fairness metrics and impossibility results. It has also expanded with other trade-offs between fairness and other ethical objectives such as privacy. We see the need to expand even broader to the additional requirements developers have to respect (not necessarily ethics-related). While a few works have investigated how to identify unfairness in model outputs in contexts where ML robustness is typically studied (i.e., distribution shifts between training and deployment data), more works need to be done in order to further mitigate unfairness in such contexts. In relation to that, explainability methods have traditionally been investigated to broadly understand a model's functioning, and we have proposed in Part III to extend their use to robustness scenarios (around natural perturbations). It would be interesting and challenging to research how to use explainability to identify and mitigate unfairness issues, in combination to robustness ones.

Finally, we support the current trend around data quality, as data is one of the main challenges towards both model robustness and fairness, but has not been investigated extensively (due to various reasons such as perceived prestige of such work). We argue (Part I), that the ML research community and the data management community should

be brought closer to identify where output fairness could be mitigated from within data engineering pipelines, and how to re-purpose data processing techniques to do so.

Evaluating technical contributions: Developing more appropriate experimental methodologies. One of the main challenges faced in Part III was the lack of rigorous framework, dataset benchmark, or metrics for evaluating model explanations. We overcame this challenge by creating our own evaluation procedure. We injected several types of biases in ML models (via dataset skews or the comparison of multiple model architectures with known resulting behavior differences) in order to obtain a proxy ground truth about the model functioning, and check to what extent the outputs of the explainability method would reflect these biases. This evaluation procedure revealed useful and represents an additional contribution of our thesis. Yet, it is not enough to further quantify the quality of the explanations (it only provides support for a qualitative analysis of the explanations). We pose that more work is necessary in terms of methodologies used by the research community, and especially in terms of evaluation benchmarks, in order to push research in the right direction.

Beyond model explanations, we also see the need for benchmarks of model diagnosis and debugging methods. For instance, we proposed an explainability-based, developer-centered, approach to model diagnosis. Yet, we also discussed the existence of fully automatic methods for model diagnosis and debugging. While both present different advantages and disadvantages, e.g., in terms of transparency, time-efficiency, etc., it could be informative to compare the effectiveness of these two types of approaches. This would bring new challenges, in terms of defining the rigorous, fair, evaluation procedure when only one of the two approaches involves humans.

Next to procedures proposing quantitative evaluations, we also believe that more user-based evaluations are needed. In Part III Chapter 10, a user-based evaluation enabled us to understand how useful the method we proposed was beyond its fidelity, and to identify avenues for future work. We believe that frequent user evaluations to compare multiple proposed solutions would be necessary. This would represent a number of challenges in terms of establishing rigorous procedures for qualitative experiments. Using use-cases that are closer to reality would be particularly relevant there, but building such use-cases would be challenging, e.g., due to necessary domain knowledge, confidentiality issues, etc.

IMPLICATIONS FOR PRACTICAL TOOLS STEMMING FROM THEORETICAL WORK

Development of usable tools. Our work also bears implications for human-computer interaction research. One of the main results in our study of practices was that many of the tools that stem from research are not directly usable by developers (Part II). They might for instance have a too-steep learning rate, they might not be adapted to their workflows and needs, the technical implementation might not be compatible with their development or production environments, etc. Hence, one important avenue for future work is the study of the non-functional requirements for tools usable by ML developers, next to the obvious need for identifying functional requirements. The use of co-creation sessions as we did in Part III is particularly useful to do so.

Besides, we noted that despite proposing to the developers a tool that presents a

plethora of relevant information for their diagnosis process, their process was not completely successful, for multiple reasons. The developers would either be overwhelmed by the diversity and amount of information, or they would be novelty-averse and preferred only using the type of information they were familiar with, or they would misinterpret the information presented to them, e.g., because of confirmation bias. Hence, next to the requirements mentioned above, we emphasize the need for understanding the human factors that impact tool use, and proposing tool designs that cater specifically for these needs. This calls for future research in collaboration between ML, human-computer interaction, and design researchers. Developers were also confused by when to select which tool available, or which metric, optimization method, etc. This shows that before designing these tools, the technical community should also strive to characterize the technical methods, and develop heuristics to advise on their practical use.

Adapting tools to the needed collaboration between stakeholders. When studying ML practices broadly and the use of explainability methods for diagnosis more specifically, we noted that our participants often worked in collaboration with various stakeholders, or required the knowledge of certain stakeholders. For instance, they obtained requirements from model owners, they were advised on how important a failure is by domain experts and decision subjects, other developers informed them about prior design choices, etc. Yet, most tools that have been developed until now to support the ML lifecycle focus on a single stakeholder, always the developer. While this is natural as they are the ones performing most of the works, it is important in the future to understand what kind of interaction is necessary in which stage of the lifecycle, with which stakeholder, and with which challenges. Later on, tools should be adapted to support the work of these different stakeholders. For instance, when using our explainability method, we noticed that the ML developers typically rely on domain knowledge to decide on the appropriateness of a model mechanism. Obtaining a timely answer from a domain expert is not only complicated, but the communication between these two types of stakeholders itself can be challenging as they do not share the same mental model of the problem, nor the same vocabulary. Investigating how they could collaborate via a tool or without one is hence necessary for the future. Qualitative studies with an expert or a developer performing the same task independently and later on together would be insightful towards that end.

IMPLICATIONS FOR SOCIO-TECHNICAL, INTERDISCIPLINARY WORK

Supporting requirement engineering for model development. Another finding was the surprising lack of research and practice around requirement engineering (Part I, II). The developers we interviewed did not necessarily conduct any deep analysis of the requirements for the ML systems they would build. This led to ambiguities in their process and potential harms, since the resulting evaluation of the systems were not based on rigorous requirements. Besides, the current ML research has not deeply investigated how to formulate precise requirements, which does not support developers in doing so themselves. Hence, we emphasize the need for future work in this direction.

Particularly, there is a clear need for investigating the requirements that developers have to account for in practice. Even though they do not explicitly formulate such re-

quirements, it appeared from our interviews that they have more constraints to respect in order develop their models, than the current research accounts for. For instance, their objectives do not necessarily only revolve around output accuracy but also inference speed, or training time, and they do not necessarily have access to many ground truth labels for their data, contrary to what is regularly assumed in ML research. For that, we foresee the need for qualitative studies via interviews but also field observations, to surface and characterize these requirements and constraints.

We also envision that a language that would allow to express various types of requirements, e.g., in terms of model outputs, model inference process, etc., would be especially beneficial for making the developers' process more structured and transparent. In Part III where we focused on mechanism diagnosis, we identified that ML developers sometimes face challenges in comparing learned mechanisms to expected ones. We pose that proposing a common formalism to express both types of mechanisms and performing automatic comparisons would support the developers further in their process, as it could provide them with potential debugging directions where the model might not be working as expected. While not all requirements might be expressible in mathematical terms, attempting at such formalism when meaningful would also be a way to later on investigate potential tensions across requirements.

Questioning the feasibility of abiding by every requirement. As we identified a number of tensions and impossibilities at the basis of ML systems and the need for negotiations between stakeholders, it quickly appeared that one can never make an ML system non-harmful to all relevant stakeholders. For instance, it is well-known that various stakeholders (e.g., the different decision-subjects of an ML model) value different conceptions of fairness and of other ethical values, that might be inherently contradictory, and/or that might be at least technically unfeasible to simultaneously uphold due to impossibility results that have been demonstrated between various fairness metrics or between fairness and other objectives such as privacy or accuracy. Instead, if one does want to employ ML, they will have to trade-off between various benefits and harms. This brings deeper ethical questions to decide when to develop an ML technique, when to deploy an ML system, on which basis to make the choice, and who should be the one making this choice? When should one refrain from using technology to resolve a problem is one important question, whose answer is subjective, but can once again take inspiration from other areas to find preliminary, relevant directions. Techno-solutionism is indeed not a trap that solely touches ML.

Accounting for non-technical factors. We also see direct implications of our work in terms of structural changes that go beyond ML or human-computer interaction research. In Part II, we identified a number of factors that impact practices positively or negatively. Among those were human factors. For instance, we identified that many developers did not use relevant tools in their process simply because they were not aware of their existence, while we also identified other developers lacking critical attitude towards their own practices. This led us to identify that education is an important factor that impacts how developers later on tackle a ML problem. We argue that revising education curricula around ML, and developing more solutions for on-the-job learning will

be important in the future, for ML developers to keep up-to-date with the outcomes of research, share good practices between each other, etc. This is not a problem that only concerns ML research, and hence one should get inspiration from other research areas to do so. An additional relevant question there is whether one should standardize practices in order to guide developers within their workflow and avoid certain harms? We argue that it is not possible and hence not desirable, facing the ever-increasing diversity of ML applications, the diversity of harms, the subjectivity of the problem, the plurality of constraints depending on organisations. Hence, one should investigate how to provide appropriate and actionable education and guidance to developers, without narrowing their views down to single, specific, problems and workflows.

What's more, the adoption of ML in the public and private sectors is accelerating recently thanks to the recent trend of *democratizing ML* [693, 21].[3] Democratizing ML has taken multiple meanings, such as making the governance of the ML systems and ML research more democratic by involving the public in the design of the systems or of new research directions, or making ML-powered services adapted to a large diversity of populations [358]. Here, we refer to the idea of making the resources (e.g., storage, computing power, data, etc.) necessary to develop and deploy ML systems accessible to a large number of developers, and lowering the entrance barrier to the development of such systems by reducing the complexity of building models (e.g., data processing pipelines, model architectures, etc.) via guidance tools or fully automated development processes, e.g., AutoML [385] (even non computer scientists such as domain experts would be able to develop ML systems). In light of these recent trends, accounting for these human factors might become even more necessary and challenging, as anyone might get the opportunity of developing models, even when they do not have a clear understanding of the potential harms [325], and the AutoML tools might obfuscate relevant reflections to have on design choices [861].

Similarly, we identified that a number of organisational factors also impact developers attitudes and practices towards harms. Even though the developers might have all the knowledge necessary to appropriately eliminate a harm, they might not be supported by their organisation to do so. In such case, we pose that policies and regulations are necessary to push changes forward.

11.2.2. REFLECTIONS ON OUR APPROACH

We now engage in a critical reflection on our choice of approach and methodologies in hindsight. This is insightful not only for ML research, but also for any other technology-based research that has potential to cause social harms.

SUCCESS OF OUR APPROACH

We adopted a two-stage, mixed-method, approach, where, for each stage, we made use of various methodologies. In the first stage, we aimed at characterizing the research / practice misalignment to better understand the reasons for the persistence of harms despite the amount of research on the topic, and to identify appropriate avenues for developing solutions. For that, in Part I, we conducted four structured surveys of the technical, interdisciplinary, and socio-technical literature dealing with ML and harms.

[3] https://www.turing.com/kb/ultimate-guide-to-democratization-in-ai

In Part II, we performed qualitative, empirical research via interviews with 50 developers and an interview analysis methodology from grounded theory. And we conducted a critical analysis of the gap between practices and research directions by comparing both, aided by insights from other disciplines. In Part III, to develop a new technical solution that supports developers in investigating potential errors of their models that might cause harms, we adopted a research through design approach with co-creation sessions, technical implementations, and user-studies with 18 developers.

While it is impossible to evaluate at the time of writing the thesis how impactful our work will be —it takes time for developers to adopt solutions stemming from research—, we believe in hindsight that having adopted this approach was a good choice. Thanks to the originality of the approach in the ML context, we contributed a diversity of results that had not been discussed extensively in the literature beforehand. We especially identified a set of future research opportunities that could lead to important changes in practices and in the research community. We believe that adopting a mixed method was the key to our endeavor, as it provided flexibility in the depth of results to obtain. Our discussions with developers testified of the potential utility of our present findings, as they orally reported being very interested in the discussions they had with us, concretely showed some changes in their practices around ML fairness after our discussion, and more successfully handled failures in their models after interacting with our user-interface. These results are encouraging for future work, and comfort us in the choice of approach. We especially emphasize the importance of an iterative, agile approach in this research space, both because it is impossible to build solutions that are perfect from the first try, and because one should regularly assess the evolution of practices (other factors might also impact practices).

RECOMMENDATIONS FOR FUTURE RESEARCH ADOPTING A SIMILAR APPROACH

It is important to recognize however that the approach we adopted was challenging for multiple reasons, and not complete as of now. We discuss such challenges and recommendations for future work here.

Challenging work with developers. Involving developers in our research was challenging in practice. The recruitment process led to around 1 in 12 positive responses to our participation requests. The sampling of developers working with a specific technology, be it deep learning for computer vision or ML based classification on tabular data, is not large in the world. The topics we questioned them over might be considered confidential or sensitive for certain organisations, preventing them from participating. The PhD happening during the COVID period, it was not possible to attend industry events on ML to get to know more developers. Successfully recruited developers were not necessarily available for a full hour, nor for a second interview, did not all agree for recordings of the interview, and we had to tune the interview process to allow for online interviews. Besides, ML in public or private organisation is a field in the making, for which no well-established vocabulary exists, leading to adjustments to make to compare practices across organisations as well as with research outputs.

While we do not think we could have avoided these challenges and they are important to overcome due to the importance of this kind of research, one could envision de-

veloping solutions in the future to allow for easier research with ML developers. For instance, one could develop structured (and privacy-preserving) processes to take note of organisations and developers that are more prone to participate to such interviews, and tools to facilitate the on-boarding process, e.g., by keeping track of contacted developers, of reminders sent, of those developers who have yet to sign consent forms or to share their availabilities to schedule an interview, etc. Structured and easily-queriable glossaries could also be built iteratively while conducting the research, to adapt quickly to the vocabulary of each organisation, in order to avoid mis-understandings (and potentially identify new insights) during the interview sessions, between researchers conducting the research, and with the terminology used by the research community.

We also recommend to adopt more diverse qualitative methods. Performing semi-structured interviews, while insightful, did not allow us to collect a number of insights we had envisioned at first. Within an one-hour interview where developers might not have time to delve deeper into their prior code, it is not possible to enter in-depth into certain activities of the developers, such as the exact type of functions they use to process their data, the exact training methods for their models, etc. Besides, we circumvented potential confidentiality issues by providing developers with our own, made-up, hypothetical scenarios, which did not always allow them to re-use relevant methods similar to those they would typically use. It is also not possible to build an in-depth understanding of their thought process facing each harm of the ML lifecycle, nor is it possible to observe potential evolution over time of their practices facing our questioning or new tools we introduce. For these reasons, we argue that other qualitative methods such as ethnographies should also be used. They would allow to observe a practitioner over time, with more attention to details such as code, within the context of their organisation that might bring new insights on constraints and structural obstacles, and enable triangulation of the already acquired information (e.g., by also accessing organisations' documents).

Another question that imposed was the definition of a *good* sample of developers. While qualitative methods talk about the "saturation point", we had the challenge of defining a scope within which to study practices. The relevant dimensions on which ML practices might vary have not been made clear until now across research publications. We also note a high disparity on the level of details reported across such studies. We propose to investigate these dimensions in the future, in order to make such research more rigorous. For instance, we noted that practices across organisations differed (e.g., based on the organisation's size, business model, resources, history, domains of application, etc.), but also across roles within an organisation (e.g., a same job title across organisations might have different implications), as well as across continents or countries (e.g., while our participant sample presents a skew towards employees of Dutch startups, most publications hint at a skew towards employees of BigTech companies in the United States of America, that might explain a number of different requirements and practices we observed in comparison to those prior works), and of course across individual developers (e.g., based on their cultural background, education, etc.). While it is not possible to sample a significant amount of developers along this plethora of definitions, further indications of a meaningful sampling could be useful.

A new socio-technical organisation of research and practice. Convinced that socio-technical research around ML is important, we now consider the conditions to successfully approach such research area. In hindsight, the main driver of this successful research was inter-disciplinarity. While computer scientists often do not receive training to conduct *qualitative research* or *critical technical work*, it revealed to be necessary for the present contributions. Knowledge and practice of the *design process* were also necessary to develop and evaluate more usable solutions. To foster such lines of works, organisations might want to develop trainings for computer scientists, and foster interdisciplinary collaborations, for which plethora of challenges are already well-known (e.g., difficulties in sharing vocabulary, aligning expectations, developing projects benefiting all parties for instance in terms of research publications, etc.). While ethics statements are becoming increasingly popular in technical research on ML, frameworks to encourage positionality and reflexive statements from qualitative ML researchers would also contribute to the quality of the research, publications, and identification of avenues for future work. While the human-computer interaction community has proposed the establishment of new roles at the intersection between research and practice to support trickling down and bubbling up activities in an effort to bridge the research-practice misalignment, we pose that such roles should also be established within the ML community, both among researchers (e.g., to avoid the lack of venues to publish such niche work) and among developers (e.g., to foster reflections around harms in ML). Establishing such roles would require making structural changes in both public and private organisations to incentivise employees in taking upon these new positions.

Next to fostering interdisciplinary collaborations with fields that are potentially closer to practice, the ML community might also reflect on developing closer collaborations with other technical fields (e.g., data management), as well as the social sciences (e.g., Science and Technology Studies, law), and domains of applications. As we showed in Part I, while these fields might remain disconnected, there are many potentials for repurposing solutions to solve ML issues, as well as adapting knowledge and methods to understand potential limitations of ML and preferences for action over those limitations (e.g., adopting a legal lens might bring to different conclusions on a problem than studying preferences of decision-subjects). Studying specific domains also has the potential to develop a broader understanding of a technology, the potential harms it causes, and the various ways that can be used to solve those (not confined around the ML algorithm itself, but broadening to the design of the user-interactions for example). Again, the interdisciplinary challenges discussed above would apply here too.

11.3. FUTURE WORK BEYOND THE SCOPE OF THIS THESIS

Next to the direct implications and future works stemming from the research reported in this thesis, we also identified needs for research extending beyond the scope of our thesis, that we had narrowed down in Introduction 1.2 (cf. Table 1.1[4]). Hence, we discuss below a broader scope that is necessary to investigate in the future.

Machine learning stakeholders. We focused primarily on the ML developers who develop an ML model to be deployed. However, many more stakeholders intervene along the ML pipeline. For instance, when conducting their work, ML developers might interact with domain experts and model owners, or even decision-subjects and model users, or the workers underlying the ML pipeline (e.g., crowd workers annotating data samples) [810]. Besides, the work of ML developers is often extended by the activity of data engineers, ML engineers, or software engineers that build the entire system around the model. We emphasize the importance of studying specifically each of these other stakeholders, to understand further potential hazards and challenges to solve these hazards.

We did not focus on particular ML developers but interviewed more than 80 individuals from a breadth of educational and cultural background, and a breadth of amount and type of experience with ML, in order to uncover a breadth of challenges and solutions. More focus on the participant sampling could be necessary in the future, to investigate less well-represented categories of developers and their particular challenges. For instance, while we primarily interviewed developers working in European countries (and sometimes in North America), it has been shown that practices, goals, and challenges might differ on other continents [420].

Domain of application. Besides, we did not focus on particular domains of applications, but remained broad both when making experiments or interviewing developers. We also did not focus on specific types of organizations in which developers work (e.g., big tech or startups; public or private; etc.), as it is often challenging to recruit a significant amount of developers across categories. Again, focusing in-depth on a few domains of applications and types of organisations would be useful in the future, to identify more specific challenges and solutions.

Machine learning stages. Within the development phase, we focused primarily on the model debugging stages. Yet, the process of developing a model is often iterative, where iterations over the dataset collection and processing, and model design and training, as well as ML pipeline debugging, are intricate. Hence, we cannot distinguish between debugging and development fully. Still, we took a specific interest in how the model is evaluated and iterated over. We did not touch upon the different ways datasets are created, but gave to our developers already-prepared datasets. Such stages of the ML lifecycle would merit further investigation as they might present a breadth of challenges depending on the application and organisation where ML developers work. Besides, the stages of the lifecycle beyond development, i.e., model deployment, production, monitoring, and updating should also be investigated next, as they are also sources of harms.

[4] We describe here the dimensions that were not extensively discussed in the Introduction.

The frontiers we draw of the ML systems we investigate remain narrow, and centered around the core ML pipeline. By that, we mean that while some hazards related to, e.g., unfairness or physical safety, come not only from the outputs of the model but how they are used in practice by decision-makers [73], we do not investigate this part of the system. This would involve for instance the design of the interactions with the decision-maker, via potential user-interfaces, educational programs, etc., that might skew them towards certain patterns of decisions, and the human factors impacting this interaction. Instead, we solely focus on aspects of the ML pipeline that stop at the model outputs.

Machine learning bugs. In terms of bugs, we focused on the ones due to a wrong configuration of the ML pipeline, as these are still extremely challenging. These bugs are typically coming from the design of the training datasets, but also from the design of the model trained on such datasets, and finally on monitoring and updating the data engineering and model training pipelines. We believe the other bugs, that are due to faulty scripts can already partially be solved by the extensive amount of research on software engineering debugging, while the ones due to a faulty translation between intended configuration and code implementation might not require an extensive technical research. Future research would still merit however to tackle those bugs.

Data and model type. We took a very narrow scope of model architectures. We primarily focused on models based on deep learning algorithms, as these are the most researched currently, and literature, e.g., on explainability, develops an extensive amount of research related to them. Yet, especially for interviews of practitioners working with tabular data, we did not exclude models relying on more traditional ML algorithms, as this remains the main approach there. We did not focus on particular types of models (e.g., GPT-3) and architectures however, as we noticed that this varies across practitioners. We focused on models performing classification tasks or regression tasks, in order to scope down the problem, and because these tasks are the most commonly researched and employed ones. Of course, other tasks such as segmentation, tracking, generation, etc., would also be useful to investigate in the future. In the future, comparisons of practices across these applications could also be insightful.

Socio-technical objectives. We did not take upon all research directions that are currently investigated, or that we identified while studying the research/practice misalignment. For instance, while a large amount of ML fairness research focuses on understanding what are the most appropriate fairness metrics for various contexts by empirically analyzing the perceptions of these metrics by proxy data subjects [786, 338], we did not engage in such user-studies focusing specifically on metrics. Similarly, while many publications investigate the perceptions of model explanations by data subjects [587], we were not interested in this topic, and instead solely focused on direct contributions to model diagnosis for avoiding hazards. We also did not engage in the conceptual identification of the appropriate metrics for relevant contextual factors. Yet, this research is becoming necessary, as we noted that developers typically see the lack of guidance along the ML lifecycle as an obstacle to building less hazardous models, or do not even realize that the models they build are harmful.

BIBLIOGRAPHY

[1] Vero Vanden Abeele, Katta Spiel, Lennart Nacke, D Johnson, and K Gerling. "Development and validation of the player experience inventory: A scale to measure player experiences at the level of functional and psychosocial consequences". In: *Intl.Journal of Human-Computer Studies* 135 (2020), p. 102370.

[2] Serge Abiteboul and Julia Stoyanovich. "Transparency, fairness, data protection, neutrality: Data management challenges in the face of new regulation". In: *Journal of Data and Information Quality (JDIQ)* 11.3 (2019), pp. 1–9.

[3] Chiara Accinelli, Simone Minisi, and Barbara Catania. "Coverage-based Rewriting for Data Preparation." In: *EDBT/ICDT Workshops*. 2020.

[4] Alan C Acock and Gordon R Stavig. "A measure of association for nonparametric statistics". In: *Social Forces* 57.4 (1979), pp. 1381–1386.

[5] D Aerts and L Gabora. "A theory of concepts and their combinations I". In: *Kybernetes* (2005).

[6] Diederik Aerts. "Quantum theory and human perception of the macro-world". In: *How Humans Recognize Objects: Segmentation, Categorization and Individual Identification* (2016), p. 210.

[7] Mouna Afif, Riadh Ayachi, Yahia Said, and Mohamed Atri. "Deep learning based application for indoor scene recognition". In: *Neural Processing Letters* 51.3 (2020), pp. 2827–2837.

[8] Avinash Agarwal, Harsh Agarwal, and Nihaarika Agarwal. "Fairness Score and process standardization: framework for fairness certification in artificial intelligence systems". In: *AI and Ethics* (2022), pp. 1–13.

[9] R Agarwal and al. "Fast algorithms for mining association rules". In: *VLDB*. 1994.

[10] Swati Agarwal and Ashish Sureka. "But i did not mean it!—intent classification of racist posts on tumblr". In: *2016 European Intelligence and Security Informatics Conference (EISIC)*. IEEE. 2016, pp. 124–127.

[11] Aniya Aggarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. "Black Box Fairness Testing of Machine Learning Models". In: *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2019. Tallinn, Estonia: ACM, 2019, pp. 625–635. ISBN: 978-1-4503-5572-8. DOI: 10.1145/3338906.3338937. URL: http://doi.acm.org/10.1145/3338906.3338937.

[12] Sweta Agrawal and Amit Awekar. "Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms". In: *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*. Ed. by Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury. Vol. 10772. Lecture Notes in Computer Science. Springer, 2018, pp. 141–153. ISBN: 978-3-319-76940-0. DOI: 10.1007/978-3-319-76941-7_11. URL: https://doi.org/10.1007/978-3-319-76941-7%5C_11.

[13] P Agre. "Toward a critical technical practice: Lessons learned in trying to reform AI in Bowker". In: *G., Star, S., Turner, W., and Gasser, L., eds, Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide, Erlbaum* (1997).

[14] Icek Ajzen. "The theory of planned behavior". In: *Organizational behavior and human decision processes* 50.2 (1991), pp. 179–211.

[15] Naveed Akhtar and Ajmal Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey". In: *Ieee Access* 6 (2018), pp. 14410–14430.

[16] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V Nori. "FairSquare: probabilistic verification of program fairness". In: *Proceedings of the ACM on Programming Languages* 1.OOPSLA (2017), p. 80.

[17] A. H. Alduailej and M. B. Khan. "The challenge of cyberbullying and its automatic detection in Arabic text". In: *Computer and Applications (ICCA), 2017 International Conference on*. IEEE. 2017.

[18] Ashraf Alhujailli and Waldemar Karwowski. “Emotional and Stress Responses to Cyberbullying”. In: *International Conference on Applied Human Factors and Ergonomics*. Springer. 2018, pp. 33–43.

[19] Syed Mustafa Ali. “A brief introduction to decolonial computing”. In: *XRDS: Crossroads, The ACM Magazine for Students* 22.4 (2016), pp. 16–21.

[20] Sridhar Alla and Suman Kalyan Adari. “What is mlops?” In: *Beginning MLOps with MLFlow*. Springer, 2021, pp. 79–124.

[21] Bibb Allen, Sheela Agarwal, Jayashree Kalpathy-Cramer, and Keith Dreyer. “Democratizing ai”. In: *Journal of the American College of Radiology* 16.7 (2019), pp. 961–963.

[22] Wafa Alorainy, Pete Burnap, Han Liu, and Matthew L Williams. ““The Enemy Among Us” Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings”. In: *ACM Transactions on the Web (TWEB)* 13.3 (2019), pp. 1–26.

[23] Ahmed Alqaraawi et al. “Evaluating saliency map explanations for convolutional neural networks: a user study”. In: *IUI*. 2020, pp. 275–285.

[24] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. “Software engineering for machine learning: A case study”. In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE. 2019, pp. 291–300.

[25] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. “Modeltracker: Redesigning performance analysis tools for machine learning”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 337–346.

[26] Alexander Amini, Ava Soleimany, Wilko Schwarting, Sangeeta Bhatia, and Daniela Rus. “Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure”. In: (2019).

[27] Paul Ammann and Jeff Offutt. *Introduction to Software Testing*. 2nd. USA: Cambridge University Press, 2016. ISBN: 1107172012.

[28] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. “Concrete problems in AI safety”. In: *arXiv preprint arXiv:1606.06565* (2016).

[29] M Ancona, E Ceolini, C Öztireli, and M Gross. “Towards better understanding of gradient-based attribution methods for Deep Neural Networks”. In: *ICLR*. 2018.

[30] Rico Angell, Brittany Johnson, Yuriy Brun, and Alexandra Meliou. “Themis: Automatically Testing Software for Discrimination”. In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2018. Lake Buena Vista, FL, USA: ACM, 2018, pp. 871–875. ISBN: 978-1-4503-5573-5. DOI: 10.1145/3236024.3264590. URL: http://doi.acm.org/10.1145/3236024.3264590.

[31] Ariful Islam Anik and Andrea Bunt. “Data-Centric Explanations: Explaining Training Data of Machine Learning Systems to Promote Transparency”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–13.

[32] Maria Anzovino, Elisabetta Fersini, and Paolo Rosso. “Automatic identification and classification of misogynistic language on twitter”. In: *International Conference on Applications of Natural Language to Information Systems*. Springer. 2018, pp. 57–64.

[33] M. E. Aragón and A. P. López-Monroy. “Author profiling and aggressiveness detection in spanish tweets: Mex-a3t 2018”. In: *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), CEUR WS Proceedings*. 2018.

[34] Keijiro Araki, Zengo Furukawa, and Jingde Cheng. “A general framework for debugging”. In: *IEEE software* 8.3 (1991), pp. 14–20.

[35] Aymé Arango, Jorge Pérez, and Barbara Poblete. “Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation”. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 45–54.

[36] D. Archard. “Insults, Free Speech and Offensiveness”. In: *Journal of Applied Philosophy* 31.2 (2014).

[37] Andrew Armitage and Diane Keeble-Allen. “Undertaking a structured literature review or structuring a literature review: Tales from the field”. In: *Proceedings of the 7th European Conference on Research Methodology for Business and Management Studies: ECRM2008, Regent's College, London*. 2008, p. 35.

[38] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. “Negative Knowledge for Open-world Wikidata”. In: *Companion of The Web Conf. 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2021, pp. 544–551.

[39] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. “Wikinegata: a Knowledge Base with Interesting Negative Statements”. In: *Proc. VLDB Endow.* 14.12 (2021), pp. 2807–2810.

[40] Zachary Arnold and Helen Toner. *AI Accidents: An Emerging Threat: what Could Happen and what to Do*. Center for Security and Emerging Technology, 2021.

[41] Ines Arous, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. “Marta: Leveraging human rationales for explainable text classification”. In: *AAAI*. Vol. 35. 7. 2021, pp. 5868–5876.

[42] Lora Aroyo, Matthew Lease, Praveen Paritosh, and Mike Schaekermann. “Data excellence for AI: why should you care?” In: *Interactions* 29.2 (2022), pp. 66–69.

[43] Lora Aroyo and Chris Welty. “Truth is a lie: Crowd truth and the seven myths of human annotation”. In: *AI Magazine* 36.1 (2015), pp. 15–24.

[44] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilovic, et al. “AI Explainability 360: An Extensible Toolkit for Understanding Data and Machine Learning Models.” In: *J. Mach. Learn. Res.* 21.130 (2020), pp. 1–6.

[45] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. “One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques”. In: (2019).

[46] Carolyn Ashurst, Solon Barocas, Rosie Campbell, and Deborah Raji. “Disentangling the Components of Ethical Research in Machine Learning”. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 2057–2068.

[47] Stavros Assimakopoulos, Fabienne H Baider, and Sharon Millar. *Online hate speech in the European Union: A discourse-analytic perspective*. Springer Nature, 2017.

[48] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. “Designing Fair Ranking Schemes”. In: *Proceedings of the 2019 International Conference on Management of Data*. SIGMOD '19. Amsterdam, Netherlands: ACM, 2019, pp. 1259–1276. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3300079. URL: http://doi.acm.org/10.1145/3299869.3300079.

[49] Abolfazl Asudeh and HV Jagadish. “Fairly evaluating and scoring items in a data set”. In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 3445–3448.

[50] Abolfazl Asudeh, HV Jagadish, and Julia Stoyanovich. “Towards Responsible Data-driven Decision Making in Score-Based Systems”. In: *Data Engineering* (2019), p. 76.

[51] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. “Assessing and remedying coverage for a given dataset”. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 2019, pp. 554–565.

[52] Joshua Attenberg, Panos Ipeirotis, and Foster Provost. “Beat the machine: Challenging humans to find a predictive model's “unknown unknowns””. In: *Journal of Data and Information Quality (JDIQ)* 6.1 (2015), pp. 1–17.

[53] Adithya Avvaru, Sanath Vobilisetty, and Radhika Mamidi. “Detecting Sarcasm in Conversation Context Using Transformer-Based Models”. In: *Proceedings of the Second Workshop on Figurative Language Processing*. 2020, pp. 98–103.

[54] Fatma Basak Aydemir and Fabiano Dalpiaz. “A roadmap for ethics-aware software engineering”. In: *Proceedings of the International Workshop on Software Fairness, FairWare@ICSE 2018, Gothenburg, Sweden, May 29, 2018*. 2018, pp. 15–21. DOI: 10.1145/3194770.3194778. URL: https://doi.org/10.1145/3194770.3194778.

[55] S Bach, A Binder, and al. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7 (2015).

[56] Pinkesh Badjatiya, M Gupta, and V Varma. “Stereotypical bias removal for hate speech detection task using knowledge-based generalizations”. In: *The World Wide Web Conference*. 2019, pp. 49–59.

[57] D Bahdanau, K Cho, and Y Bengio. “Neural machine translation by joIntl. learning to align and translate”. In: *ICLR*. 2015.

[58] A. Balayn, P. Mavridis, A Bozzon, B. Timmermans, and Z. Szlávik. “Characterising and Mitigating Aggregation-Bias in Crowdsourced Toxicity Annotations”. In: *Short Paper Proceedings of the 1st Workshop on Disentangling the Relation Between Crowdsourcing and Bias Management*. CEUR, 2018.

[59] Agathe Balayn and Alessandro Bozzon. “Designing evaluations of machine learning models for subjective inference: the case of sentence toxicity”. In: *Rigorous Evaluation of Machine Learning workshop (HCOMP’19)* (2019).

[60] Agathe Balayn, Alessandro Bozzon, and Zoltan Szlavik. “Unfairness towards subjective opinions in Machine Learning”. In: *Human-Centered Machine Learning workshop (CHI’19)* (2019).

[61] Agathe Balayn, Ujwal Gadiraju, and Jie Yang. “"Accuracy-fairness trade-off, let’s cut the burrito in half"? On the Conceptions and Practices of ML Developers towards Algorithmic Fairness and Harms”. In: 2023.

[62] Agathe Balayn and Seda Gürses. “Beyond Debiasing: Regulating AI and its inequalities”. In: *Report for the European Digital Rights organisation (EDRi). https://edri. org/wp-content/uploads/2021/09/EDRi_Beyond-Debiasing-Report_Online. pdf* (2021).

[63] Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. “Finditout: A multiplayer gwap for collecting plural knowledge”. In: *Vol. 9 (2021): Proceedings of the Ninth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 2021.

[64] Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. “Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game”. In: *the Web Conference (WWW)*. 2022, pp. 1709–1719.

[65] Agathe Balayn, Bogdan Kulynych, and Seda Guerses. “Exploring Data Pipelines through the Process Lens: a Reference Model for Computer Vision”. In: *Beyond Fairness workshop (CVPR’21)* (2021).

[66] Agathe Balayn, Christoph Lofi, and Geert-Jan Houben. “Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems”. In: *The VLDB Journal (VLDBJ)* 30.5 (2021), pp. 739–768.

[67] Agathe Balayn, Natasa Rikalo, Christoph Lofi, Jie Yang, and Alessandro Bozzon. “How can Explainability Methods be Used to Support Bug Identification in Computer Vision Models?” In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2022, pp. 1–16.

[68] Agathe Balayn, Natasa Rikalo, Jie Yang, and Alessandro Bozzon. “Faulty or Ready? Handling Failures in Deep-Learning Computer Vision Models until Deployment: A Study of Practices, Challenges, and Needs”. In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2023, pp. 1–20.

[69] Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, and Alessandro Bozzon. “What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis”. In: *Proceedings of the Web Conference 2021 (WWW)*. 2021, pp. 1937–1948.

[70] Agathe Balayn, Jie Yang, Zoltan Szlavik, and Alessandro Bozzon. “Automatic Identification of Harmful, Aggressive, Abusive, and Offensive Language on the Web: A Survey of Technical Biases Informed by Psychology Literature”. In: *ACM Transactions on Social Computing (TSC)* 4.3 (2021), pp. 1–56.

[71] Agathe Balayn, Mireia Yurrita, Jie Yang, and Ujwal Gadiraju. ““☑ Fairness Toolkits, A Checkbox Culture?” On the Factors that Fragment Developer Practices in Handling Algorithmic Harms”. In: *ACM Conference on AI, Ethics, and Society (AIES)*. 2023.

[72] Gagan Bansal. “Explanatory Dialogs: Towards Actionable, Interactive Explanations”. In: *AIES ’18*. 2018, pp. 356–357. ISBN: 9781450360128.

[73] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. “Is the most accurate ai the best teammate? optimizing ai for teamwork”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 11405–11414.

[74] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. "Beyond accuracy: The role of mental models in human-AI team performance". In: *Proceedings of the AAAI conference on human computation and crowdsourcing*. Vol. 7. 2019, pp. 2–11.

[75] Gagan Bansal and Daniel S Weld. "A coverage-based utility model for identifying unknown unknowns". In: *AAAI, 2018* (2018).

[76] Srijan Bansal, Vishal Garimella, Ayush Suhane, Jasabanta Patro, and Animesh Mukherjee. "Code-switching patterns can be an effective route to improve performance of downstream NLP applications: A case study of humour, sarcasm and hate speech detection". In: *arXiv preprint arXiv:2005.02295* (2020).

[77] Niels Bantilan. "Themis-ml: A fairness-aware machine learning interface for end-to-end discrimination discovery and mitigation". In: *Journal of Technology in Human Services* 36.1 (2018), pp. 15–30.

[78] Chelsea Barabas, Madars Virza, Karthik Dinakar, Joichi Ito, and Jonathan Zittrain. "Interventions over Predictions: Reframing the Ethical Debate for Actuarial Risk Assessment". In: *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by Sorelle A. Friedler and Christo Wilson. Vol. 81. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, 23–24 Feb 2018, pp. 62–76. URL: http://proceedings.mlr.press/v81/barabas18a.html.

[79] Natã M Barbosa and Monchu Chen. "Rehumanized crowdsourcing: a labeling framework addressing bias and ethics in machine learning". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.

[80] A Barbu, D Mayo, and al. "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models". In: *NeurIPS*. 2019, pp. 9453–9463.

[81] Pinar Barlas, Styliani Kleanthous, Kyriakos Kyriakou, and Jahna Otterbacher. "What Makes an Image Tagger Fair?" In: *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP '19. Larnaca, Cyprus: ACM, 2019, pp. 95–103. ISBN: 978-1-4503-6021-0. DOI: 10.1145/3320435.3320442. URL: http://doi.acm.org/10.1145/3320435.3320442.

[82] Solon Barocas, Asia J Biega, Benjamin Fish, Jędrzej Niklas, and Luke Stark. "When not to design, build, or deploy". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 695–695.

[83] Solon Barocas, Anhong Guo, Ece Kamar, Jacquelyn Krones, Meredith Ringel Morris, Jennifer Wortman Vaughan, W Duncan Wadsworth, and Hanna Wallach. "Designing disaggregated evaluations of ai systems: Choices, considerations, and tradeoffs". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 368–378.

[84] Solon Barocas, Moritz Hardt, and Arvind Narayanan. "Fairness in machine learning". In: *NIPS Tutorial* (2017).

[85] Solon Barocas and Andrew D Selbst. "Big data's disparate impact". In: *California law review* (2016), pp. 671–732.

[86] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. "Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 54–63.

[87] Alex Bäuerle, Ángel Alexander Cabrera, Fred Hohman, Megan Maher, David Koski, Xavier Suau, Titus Barik, and Dominik Moritz. "Symphony: Composing Interactive Interfaces for Machine Learning". In: *CHI*. 2022, pp. 1–14.

[88] J. Bayzick. "Detecting the Presence of Cyberbullying Using Computer Software". In: *Proceedings of the 3rd International Web Science Conference*. 2011, pp. 1–2.

[89] Marvin van Bekkum and Frederik Zuiderveen Borgesius. "Using sensitive data to prevent discrimination by artificial intelligence: Does the GDPR need a new exception?" In: *Computer Law & Security Review* 48 (2023), p. 105770.

[90] Andrew Bell, Ian Solano-Kamaiko, Oded Nov, and Julia Stoyanovich. "It's Just Not That Simple: An Empirical Study of the Accuracy-Explainability Trade-off in Machine Learning for Public Policy". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 248–266.

[91] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias". In: *IBM Journal of Research and Development* 63.4/5 (2019), pp. 4–1.

[92] A. Bellmore, J. Calvin, J-M. Xu, and X. Zhu. "The five W's of "bullying" on Twitter: Who, What, Why, Where, and When". In: *Computers in Human Behavior* 44 (Mar. 2015). ISSN: 0747-5632. DOI: 10.1016/J.CHB.2014.11.052. URL: https://www.sciencedirect.com/science/article/pii/S0747563214006621.

[93] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 610–623.

[94] Sebastian Benthall and Bruce D. Haynes. "Racial Categories in Machine Learning". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT* '19. Atlanta, GA, USA: ACM, 2019, pp. 289–298. ISBN: 978-1-4503-6125-5. DOI: 10.1145/3287560.3287575. URL: http://doi.acm.org/10.1145/3287560.3287575.

[95] Janine Berg, Marianne Furrer, Ellie Harmon, Uma Rani, and M Six Silberman. "Digital labour platforms and the future of work". In: *Towards Decent Work in the Online World. Rapport de l'OIT* (2018).

[96] Karl Berggren, Qiangfei Xia, Konstantin K Likharev, Dmitri B Strukov, Hao Jiang, Thomas Mikolajick, Damien Querlioz, Martin Salinga, John R Erickson, Shuang Pi, et al. "Roadmap on emerging hardware and technology for machine learning". In: *Nanotechnology* 32.1 (2020), p. 012002.

[97] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. "Fairness in Recommendation Ranking through Pairwise Comparisons". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 2019, pp. 2212–2220. DOI: 10.1145/3292500.3330745. URL: https://doi.org/10.1145/3292500.3330745.

[98] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H Chi. "Putting fairness principles into practice: Challenges, metrics, and improvements". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 453–459.

[99] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. "Are we done with ImageNet?" In: *arXiv preprint arXiv:2006.07159* (2020).

[100] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. "Explainable machine learning in deployment". In: *FAccT*. 2020, pp. 648–657.

[101] Adrien Bibal, Michael Lognoul, Alexandre De Streel, and Benoit Frenay. "Legal requirements on explainability in machine learning". In: *Artificial Intelligence and Law* 29.2 (2021), pp. 149–169.

[102] Elettra Bietti. "From ethics washing to ethics bashing: a view on tech ethics from within moral philosophy". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 210–219.

[103] R. Binns, M. Veale, M. Van Kleek, and N. Shadbolt. "Like trainer, like bot? Inheritance of bias in algorithmic content moderation". In: *International Conference on Social Informatics*. Springer. 2017.

[104] Reuben Binns. "Fairness in machine learning: Lessons from political philosophy". In: *Conference on Fairness, Accountability and Transparency*. PMLR. 2018, pp. 149–159.

[105] Reuben Binns. "On the apparent conflict between individual and group fairness". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 514–524.

[106] Christian Bird, Adrian Bachmann, Eirik Aune, John Duffy, Abraham Bernstein, Vladimir Filkov, and Premkumar Devanbu. "Fair and Balanced?: Bias in Bug-fix Datasets". In: *Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*. ESEC/FSE '09. Amsterdam, The Netherlands: ACM, 2009, pp. 121–130. ISBN: 978-1-60558-001-2. DOI: 10.1145/1595696.1595716. URL: http://doi.acm.org/10.1145/1595696.1595716.

[107] Sarah Bird, Miro Dudik, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. “Fairlearn: A toolkit for assessing and improving fairness in AI”. In: *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).

[108] A Birhane and O Guest. “Towards decolonising computational sciences”. In: *Women, Gender and Resesarch* 2021 (2021), pp. 60–73.

[109] Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. “The values encoded in machine learning research”. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 173–184.

[110] Abeba Birhane and Vinay Uday Prabhu. “Large image datasets: A pyrrhic win for computer vision?” In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2021, pp. 1536–1546.

[111] Aditya Bikram Biswas, Hiba Arnaout, and Simon Razniewski. “Neguess: Wikidata-entity guessing game with negative clues”. In: (2021).

[112] Erik Bleich. “The Rise of Hate Speech and Hate Crime Laws in Liberal Democracies”. In: *Journal of Ethnic and Migration Studies* 37.6 (2011), pp. 917–934. DOI: 10.1080/1369183X.2011.576195. eprint: https://doi.org/10.1080/1369183X.2011.576195. URL: https://doi.org/10.1080/1369183X.2011.576195.

[113] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. “Language (Technology) is Power: A Critical Survey of" Bias" in NLP”. In: *arXiv preprint arXiv:2005.14050* (2020).

[114] R. J. Boeckmann and J. Liew. “Hate speech: Asian American students’ justice judgments and psychological responses”. In: *Journal of Social Issues* 58.2 (2002), pp. 363–381.

[115] Matthias Boehm, Arun Kumar, and Jun Yang. “Data management in machine learning systems”. In: *Synthesis Lectures on Data Management* 11.1 (2019), pp. 1–173.

[116] Angie Boggust, Benjamin Hoover, Arvind Satyanarayan, and Hendrik Strobelt. “Shared Interest: Measuring Human-AI Alignment to Identify Recurring Patterns in Model Behavior”. In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–17.

[117] Veronika Bogina, Alan Hartman, Tsvi Kuflik, and Avital Shulner-Tal. “Educating Software and AI Stakeholders About Algorithmic Fairness, Accountability, Transparency and Ethics”. In: *International Journal of Artificial Intelligence in Education* (2021), pp. 1–26.

[118] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava. “A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection”. In: *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2018, pp. 36–41. DOI: 10.18653/v1/W18-1105. URL: http://aclweb.org/anthology/W18-1105.

[119] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. “Man is to computer programmer as woman is to homemaker? debiasing word embeddings”. In: *Advances in neural information processing systems*. 2016, pp. 4349–4357.

[120] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).

[121] Jason Borenstein and Ayanna Howard. “Emerging challenges in AI and the need for AI ethics education”. In: *AI and Ethics* 1.1 (2021), pp. 61–65.

[122] Ria Mae Borromeo, Thomas Laurent, Motomichi Toyama, and Sihem Amer-Yahia. “Fairness and Transparency in Crowdsourcing”. In: *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017*. 2017, pp. 466–469. DOI: 10.5441/002/edbt.2017.46. URL: https://doi.org/10.5441/002/edbt.2017.46.

[123] Pierre Bourque, Richard E Fairley, et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press, 2014.

[124] Danah Boyd and Kate Crawford. “Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon”. In: *Information, communication & society* 15.5 (2012), pp. 662–679.

[125] Karen L Boyd. “Datasheets for Datasets help ML Engineers Notice and Understand Ethical Issues in Training Data”. In: *ACM on Human-Computer Interaction* 5.CSCW2 (2021), pp. 1–27.

[126] Luka Bradeško, Michael Witbrock, Janez Starc, Zala Herga, Marko Grobelnik, and Dunja Mladenić. “Curious Cat–Mobile, Context-Aware Conversational Crowdsourcing Knowledge Acquisition”. In: *TOIS* 35.4 (2017), pp. 1–46.

[127] Virginia Braun and Victoria Clarke. “Using thematic analysis in psychology”. In: *Qualitative research in psychology* 3.2 (2006), pp. 77–101.

[128] L Breiman, J Friedman, and al. *Classification and regression trees*. 1984.

[129] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.

[130] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[131] Uwe Bretschneider and Ralf Peters. “Detecting Offensive Statements towards Foreigners in Social Media”. In: *50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, Hawaii, USA, January 4-7, 2017*. Ed. by Tung Bui. AIS Electronic Library (AISeL), 2017, pp. 1–10. URL: http://hdl.handle.net/10125/41423.

[132] Benedetta Brevini. “Black boxes, not green: Mythologizing artificial intelligence and omitting the environment”. In: *Big Data & Society* 7.2 (2020), p. 2053951720935141.

[133] Elmira van den Broek, Anastasia Sergeeva, and Marleen Huysman. “When the machine meets the expert: an ethnography of developing AI for hiring”. In: *MIS Quarterly* 45.3 (2021).

[134] Yuriy Brun and Alexandra Meliou. “Software Fairness”. In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2018. Lake Buena Vista, FL, USA: ACM, 2018, pp. 754–759. ISBN: 978-1-4503-5573-5. DOI: 10.1145/3236024.3264838. URL: http://doi.acm.org/10.1145/3236024.3264838.

[135] Marc-Etienne Brunet, C Alkalay-Houlihan, A Anderson, and R Zemel. “Understanding the origins of bias in word embeddings”. In: *International Conference on Machine Learning*. 2019, pp. 803–811.

[136] Elizabeth Buie. “HCI standards: A mixed blessing”. In: *Interactions* 6.2 (1999), pp. 36–42.

[137] Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society. *Special Issue on Fairness, Diversity, and Transparency in Data Systems, Vol. 42 No. 3*. available at: http://sites.computer.org/debull/A19sept/A19SEPT-CD.pdf (Feb. 2020). Sept. 2019.

[138] Joy Buolamwini and Timnit Gebru. “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91.

[139] Victoria K Burbank. “Cross-cultural perspectives on aggression in women and girls: An introduction”. In: *Sex Roles* 30.3-4 (1994), pp. 169–176.

[140] Nadia Burkart and Marco F Huber. “A survey on the explainability of supervised machine learning”. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.

[141] Robin Burke. “Multisided fairness for recommendation”. In: *arXiv preprint arXiv:1707.00093* (2017).

[142] P. Burnap and M. L. Williams. “Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making”. In: *Policy & Internet* 7 (2015). ISSN: 19442866. DOI: 10.1002/poi3.85. URL: http://doi.wiley.com/10.1002/poi3.85.

[143] P. Burnap and M. L. Williams. “Hate speech, machine classification and statistical modelling of information flows on Twitter: Interpretation and communication for policy decision making”. In: *Internet, Policy and Politics* (2014).

[144] P. Burnap and M. L. Williams. “Us and them: identifying cyber hate on Twitter across multiple protected characteristics”. In: *EPJ Data Science* 5.1 (Dec. 2016), p. 11. ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-016-0072-6. URL: http://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-016-0072-6.

[145] Jenna Burrell. “How the machine ‘thinks’: Understanding opacity in machine learning algorithms”. In: *Big data & society* 3.1 (2016), p. 2053951715622512.

[146] Jenna Burrell, Zoe Kahn, Anne Jonas, and Daniel Griffin. “When users control the algorithms: values expressed in practices on twitter”. In: *Proceedings of the ACM on human-computer interaction* 3.CSCW (2019), pp. 1–20.

[147] Emanuelle Burton, Judy Goldsmith, and Nicholas Mattei. “Teaching AI Ethics Using Science Fiction.” In: *Aaai workshop: Ai and ethics*. Citeseer. 2015.

[148] Chelmis C., Zois D-S., and Yao M. “Mining Patterns of Cyberbullying on Twitter”. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, Nov. 2017, pp. 126–133. ISBN: 978-1-5386-3800-2. DOI: 10.1109/ICDMW.2017.22. URL: http://ieeexplore.ieee.org/document/8215653/.

[149] Ángel Alexander Cabrera, Abraham J Druck, Jason I Hong, and Adam Perer. “Discovering and Validating AI Errors With Crowdsourced Failure Reports”. In: *ACM on Human-Computer Interaction* 5.CSCW2 (2021), pp. 1–22.

[150] Gabriel Cadamuro, Ran Gilad-Bachrach, and Xiaojin Zhu. “Debugging machine learning models”. In: *ICML Workshop on Reliable Machine Learning in the Wild*. Vol. 103. 2016.

[151] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. “" Hello AI": uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making”. In: *Proceedings of the ACM on Human-computer Interaction* 3.CSCW (2019), pp. 1–24.

[152] Shanqing Cai, Eric Breck, Eric Nielsen, M Salib, and D Sculley. “Tensorflow debugger: Debugging dataflow graphs for machine learning”. In: (2016).

[153] Toon Calders and Sicco Verwer. “Three naive Bayes approaches for discrimination-free classification”. In: *Data Mining and Knowledge Discovery* 21.2 (2010), pp. 277–292.

[154] Gul Calikli, Ayse Bener, and Berna Arslan. “An Analysis of the Effects of Company Culture, Education and Experience on Confirmation Bias Levels of Software Developers and Testers”. In: *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 2*. ICSE ’10. Cape Town, South Africa: ACM, 2010, pp. 187–190. ISBN: 978-1-60558-719-6. DOI: 10.1145/1810295.1810326. URL: http://doi.acm.org/10.1145/1810295.1810326.

[155] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. “Semantics derived automatically from language corpora contain human-like biases”. In: *Science* 356.6334 (2017), pp. 183–186.

[156] Scott Allen Cambo and Darren Gergle. “Model Positionality and Computational Reflexivity: Promoting Reflexivity in Data Science”. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–19.

[157] Alastair V Campbell, Jacqueline Chin, and Teck-Chuan Voo. “How can we know that ethics education produces ethical doctors?” In: *Medical teacher* 29.5 (2007), pp. 431–436.

[158] Ran Canetti, Aloni Cohen, Nishanth Dikkala, Govind Ramnarayan, Sarah Scheffler, and Adam Smith. “From soft classifiers to hard decisions: How fair can we be?” In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. 2019, pp. 309–318.

[159] Nicholas Carlini, A. Athalye, N. Papernot, W. Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. “On evaluating adversarial robustness”. In: *arXiv preprint arXiv:1902.06705* (2019).

[160] Nicholas Carlini and David Wagner. “Towards evaluating the robustness of neural networks”. In: *2017 IEEE symposium on security and privacy (sp)*. Ieee. 2017, pp. 39–57.

[161] Alexandra Carter. *Cathy O’Neil (2016) Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, New York, St. Martin’s Press and Virginia Eubanks (2018) Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor, New York, Broadway Books*. 2018.

[162] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. “Machine learning interpretability: A survey on methods and metrics”. In: *Electronics* 8.8 (2019), p. 832.

[163] Sergio Andrés Castaño-Pulgarin, Natalia Suárez-Betancur, Luz Magnolia Tilano Vega, and Harvey Mauricio Herrera López. “Internet, social media and online hate speech. Systematic review”. In: *Aggression and Violent Behavior* (2021), p. 101608.

[164] Dilek Cetindamar, Kirsty Kitto, Mengjia Wu, Yi Zhang, Babak Abedin, and Simon Knight. "Explicating AI Literacy of Employees at Digital Workplaces". In: *Trans. on Engineering Management* (2022).

[165] C. Chang, G. Adam, and A. Goldenberg. "Towards Robust Classification Model by Counterfactual and Invariant Data Generation". In: *2021 CVPR*. Los Alamitos, CA, USA: IEEE Computer Society, June 2021, pp. 15207–15216. DOI: 10.1109/CVPR46437.2021.01496. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01496.

[166] Kathy Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis*. sage, 2006.

[167] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali. "Mean Birds: Detecting Aggression and Bullying on Twitter". In: *Proceedings of the 2017 ACM on Web Science Conference*. WebSci '17. Troy, New York, USA: ACM, 2017, pp. 13–22. ISBN: 978-1-4503-4896-6. DOI: 10.1145/3091478.3091487. URL: http://doi.acm.org/10.1145/3091478.3091487.

[168] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali. "Measuring #GamerGate: A Tale of Hate, Sexism, and Bullying Despoina". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. New York, USA: ACM Press, 2017, pp. 1285–1290. ISBN: 9781450349147. DOI: 10.1145/3041021.3053890. URL: http://dl.acm.org/citation.cfm?doid=3041021.3053890.

[169] Alessandro Checco, Kevin Roitero, Eddy Maddalena, S Mizzaro, and G Demartini. "Let's agree to disagree: Fixing agreement measures for crowdsourcing". In: *Fifth AAAI Conf. on Human Computation and Crowdsourcing*. 2017.

[170] H. Chen, S. Mckeever, and S. J. Delany. "Presenting a Labelled Dataset for Real-time Detection of Abusive User Posts". In: *Proceedings of the International Conference on Web Intelligence*. WI '17. Leipzig, Germany: ACM, 2017, pp. 884–890. ISBN: 978-1-4503-4951-2. DOI: 10.1145/3106426.3106456. URL: http://doi.acm.org/10.1145/3106426.3106456.

[171] Hao Chen, Susan McKeever, and Sarah Jane Delany. "A comparison of classical versus deep learning techniques for abusive content detection on social media sites". In: *International Conference on Social Informatics*. Springer. 2018, pp. 117–133.

[172] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. *EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples*. 2017. DOI: 10.48550/ARXIV.1709.04114. URL: https://arxiv.org/abs/1709.04114.

[173] Shang-Tse Chen, C. Cornelius, J. Martin, and D. Horng Chau. "ShapeShifter: Robust Physical Adversarial Attack on Faster R-CNN Object Detector". In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 52–68. DOI: 10.1007/978-3-030-10925-7_4. URL: https://doi.org/10.1007%2F978-3-030-10925-7_4.

[174] Hao-Fei Cheng et al. "Explaining decision-making algorithms through UI: Strategies to help non-expert stakeholders". In: *CHI*. 2019, pp. 1–12.

[175] Hao-Fei Cheng, Logan Stapleton, Ruiqi Wang, Paige Bullock, Alexandra Chouldechova, Zhiwei Steven Steven Wu, and Haiyi Zhu. "Soliciting stakeholders' fairness notions in child maltreatment predictive systems". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–17.

[176] Lu Cheng, Ruocheng Guo, and Huan Liu. "Robust cyberbullying detection with causal interpretation". In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 169–175.

[177] Ruiqi Cheng, Kaiwei Wang, Jian Bai, and Zhijie Xu. "Unifying visual localization and scene recognition for people with visual impairment". In: *IEEE Access* 8 (2020), pp. 64284–64296.

[178] Naganna Chetty and Sreejith Alathur. "Hate speech review in the context of online social networks". In: *Aggression and violent behavior* 40 (2018), pp. 108–118.

[179] Ram Chillarese. *Software Testing Best Practices, IBM Research*. TR Patent RC21,457. 1999.

[180] Junghoo Cho, Sourashis Roy, and Robert Adams. "Page Quality: In Search of an Unbiased Web Ranking". In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*. 2005, pp. 551–562. DOI: 10.1145/1066157.1066220. URL: https://doi.org/10.1145/1066157.1066220.

[181] Shivang Chopra, Ramit Sawhney, Puneet Mathur, and Rajiv Ratn Shah. "Hindi-English Hate Speech Detection: Author Profiling, Debiasing, and Practical Perspectives". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 01. 2020, pp. 386–393.

[182] Alexandra Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[183] Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Abdelali, Soon-gyo Jung, Bernard J Jansen, and Joni Salminen. "A Multi-Platform Arabic News Comment Dataset for Offensive Language Detection". In: *Proceedings of The 12th Language Resources and Evaluation Conference*. 2020, pp. 6203–6212.

[184] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.

[185] Michael Chromik, Malin Eiband, Felicitas Buchner, Adrian Krüger, and Andreas Butz. "I Think I Get Your Point, AI! The Illusion of Explanatory Depth in Explainable AI". In: *26th Intl. Conf. on Intelligent User Interfaces*. 2021, pp. 307–317.

[186] Xu Chu, Ihab F Ilyas, and Paolo Papotti. "Holistic data cleaning: Putting violations into context". In: *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE. 2013, pp. 458–469.

[187] Don Clark. *Knowledge*. July 2012. URL: http://knowledgejump.com/knowledge/knowledge.html.

[188] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. *Certified Adversarial Robustness via Randomized Smoothing*. 2019. DOI: 10.48550/ARXIV.1902.02918. URL: https://arxiv.org/abs/1902.02918.

[189] Lucas Colusso, Cynthia L Bennett, Gary Hsieh, and Sean A Munson. "Translational resources: Reducing the gap between academic research and HCI practice". In: *Proceedings of the 2017 Conference on Designing Interactive Systems*. 2017, pp. 957–968.

[190] Lucas Colusso, Ridley Jones, Sean A Munson, and Gary Hsieh. "A translational science model for HCI". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–13.

[191] Luca Console, Daniele Theseider Dupre, and Pietro Torasso. "A Theory of Diagnosis for Incomplete Causal Models." In: *IJCAI*. 1989, pp. 1311–1317.

[192] A Feder Cooper, Ellen Abrams, and Na Na. "Emergent unfairness in algorithmic fairness-accuracy trade-off research". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 46–54.

[193] A Feder Cooper, Emanuel Moss, Benjamin Laufer, and Helen Nissenbaum. "Accountability in an algorithmic society: relationality, responsibility, and robustness in machine learning". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 864–876.

[194] Sam Corbett-Davies and Sharad Goel. "The measure and mismeasure of fairness: A critical review of fair machine learning". In: *arXiv preprint arXiv:1808.00023* (2018).

[195] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. "Algorithmic decision making and the cost of fairness". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2017, pp. 797–806.

[196] Sebastian Correa and Alberto Martin. "Linguistic Generalization of Slang Used in Mexican Tweets, Applied in Aggressiveness Detection". In: *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018*. Ed. by Paolo Rosso, Julio Gonzalo, Raquel Martinez, Soto Montalvo, and Jorge Carrillo de Albornoz. Vol. 2150. CEUR Workshop Proceedings. CEUR-WS.org, 2018, pp. 119–127. URL: http://ceur-ws.org/Vol-2150/MEX-A3T%5C_paper5.pdf.

[197] K. Cortis and S. Handschuh. "Analysis of cyberbullying tweets in trending world events". In: *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business - i-KNOW '15*. New York, USA: ACM Press, 2015, pp. 1–8. ISBN: 9781450337212. DOI: 10.1145/2809563.2809605. URL: http://dl.acm.org/citation.cfm?doid=2809563.2809605.

[198] G. Cowan and C. Hodge. "Judgments of hate speech: The effects of target group, publicness, and behavioral responses of the target". In: *Journal of Applied Social Psychology* 26.4 (1996), pp. 355–374.

[199] G. Cowan and D. Khatchadourian. “Empathy, ways of knowing, and interdependence as mediators of gender differences in attitudes toward hate speech and freedom of speech”. In: *Psychology of Women Quarterly* 27.4 (2003), pp. 300–308.

[200] G. Cowan and J. Mettrick. “The effects of target variables and setting on perceptions of hate speech1”. In: *Journal of Applied Social Psychology* 32.2 (2002), pp. 277–299.

[201] Kate Crawford and Trevor Paglen. “Excavating AI: The politics of images in machine learning training sets”. In: *Ai & Society* 36.4 (2021), pp. 1105–1116.

[202] John W Creswell. “Mixed-method research: Introduction and application”. In: *Handbook of educational policy*. Elsevier, 1999, pp. 455–472.

[203] G. B. Cunningham, M. Ferreira, and J. S. Fink. “Reactions to prejudicial statements: The influence of statement content and characteristics of the commenter.” In: *Group Dynamics: Theory, Research, and Practice* 13.1 (2009), p. 59.

[204] Alexander D’Amour, K Heller, D Moldovan, B Adlam, B Alipanahi, A Beutel, C Chen, J Deaton, J Eisenstein, M D Hoffman, et al. “Underspecification Presents Challenges for Credibility in Modern Machine Learning”. In: *arXiv preprint arXiv:2011.03395* (2020).

[205] Alexander D’Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, David Sculley, and Yoni Halpern. “Fairness is not static: deeper understanding of long term fairness via simulation studies”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 525–534.

[206] M Dadvar, FMG de Jong, R Ordelman, and D Trieschnigg. “Improved cyberbullying detection using gender information”. In: *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*. University of Ghent. 2012.

[207] M Dadvar, D Trieschnigg, and F de Jong. “Experts and machines against bullies: A hybrid approach to detect cyberbullies”. In: *Canadian Conference on Artificial Intelligence*. Springer. 2014, pp. 275–281.

[208] M Dadvar, D Trieschnigg, R Ordelman, and F de Jong. “Improving cyberbullying detection with user context”. In: *European Conference on Information Retrieval*. Springer. 2013, pp. 693–696.

[209] Maral Dadvar, R Ordelman, F de Jong, and D Trieschnigg. “Towards user modelling in the combat against cyberbullying”. In: *International Conference on Application of Natural Language to Information Systems*. Springer. 2012, pp. 277–283.

[210] Jared F Danker and John R Anderson. “The ghosts of brain states past: remembering reactivates the brain regions engaged during encoding.” In: *Psychological bulletin* 136.1 (2010), p. 87.

[211] Maitraye Das, Brent Hecht, and Darren Gergle. “The Gendered Geography of Contributions to OpenStreetMap: Complexities in Self-Focus Bias”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: ACM, 2019, 563:1–563:14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300793. URL: http://doi.acm.org/10.1145/3290605.3300793.

[212] Sumit Das, Aritra Dey, Akash Pal, and Nabamita Roy. “Applications of artificial intelligence in machine learning: review and prospect”. In: *International Journal of Computer Applications* 115.9 (2015).

[213] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. “Racial Bias in Hate Speech and Abusive Language Detection Datasets”. In: *Proceedings of the Third Workshop on Abusive Language Online*. 2019, pp. 25–35.

[214] Brittany Davis, Maria Glenski, William Sealy, and Dustin Arendt. “Measure utility, gain trust: practical advice for XAI researchers”. In: *2020 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. IEEE. 2020, pp. 1–8.

[215] T. De Smedt, G. De Pauw, and P. Van Ostaeyen. “Automatic Detection of Online Jihadist Hate Speech”. In: *Computational Linguistics & Psycholinguistics Technical Report Series, CTRS-007, FEBRUARY 2018* (Mar. 2018).

[216] Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. “Does object recognition work for everyone?” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 52–59.

[217] Laura P Del Bosque and Sara Elena Garza. "Aggressive text detection for cyberbullying". In: *Mexican International Conference on Artificial Intelligence*. Springer. 2014, pp. 221–232.

[218] Hervé Delseny, Christophe Gabreau, Adrien Gauffriau, Bernard Beaudouin, Ludovic Ponsolle, Lucian Alecu, Hugues Bonnin, Brice Beltran, Didier Duchel, Jean-Brice Ginestet, et al. "White paper machine learning in certified systems". In: *arXiv preprint arXiv:2103.10529* (2021).

[219] J Deng, W Dong, R Socher, L-J Li, K Li, and L Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. IEEE. 2009, pp. 248–255.

[220] Wesley Hanwen Deng, Manish Nagireddy, Michelle Seng Ah Lee, Jatinder Singh, Zhiwei Steven Wu, Kenneth Holstein, and Haiyi Zhu. "Exploring How Machine Learning Practitioners (Try To) Use Fairness Toolkits". In: (2022).

[221] N. Dennis-Gitari, Z. Zuping, H. Damien, and J. Long. "A Lexicon-based Approach for Hate Speech Detection". In: *International Journal of Multimedia and Ubiquitous Engineering* 10.4 (2015). ISSN: 1975-0080. DOI: 10.14257/ijmue.2015.10.4.21. URL: http://dx..

[222] Mark Diaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. "Addressing Age-Related Bias in Sentiment Analysis". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 412:1–412:14. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173986. URL: http://doi.acm.org/10.1145/3173574.3173986.

[223] K. R. Dickson. "All Prejudices are not Created Equal: Different Responses to Subtle versus Blatant Expressions of Prejudice". In: *Electronic Thesis and Dissertation Repository*. https://ir.lib.uwo.ca/etd/704 (2012).

[224] Edward Dillon, Jamie Macbeth, Robin Kowalski, Elizabeth Whittaker, and Juan E Gilbert. ""Is This Cyberbullying or Not?": Intertwining Computational Detection with Human Perception (A Case Study)". In: *Advances in Human Factors in Cybersecurity*. Springer, 2016, pp. 337–345.

[225] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard. "Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying". In: *ACM Transactions on Interactive Intelligent Systems* 2.3 (Sept. 2012), pp. 1–30. ISSN: 21606455. DOI: 10.1145/2362394.2362400. URL: http://dl.acm.org/citation.cfm?doid=2362394.2362400.

[226] K. Dinakar, R. Reichart, and H. Lieberman. "Modeling the detection of Textual Cyberbullying." In: *The Social Mobile Web* 11.02 (2011). URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/3841/4384.

[227] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. *Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack*. 2019. arXiv: 1908.06083 [cs.CL].

[228] Tilman Dingler, Ashris Choudhury, and Vassilis Kostakos. "Biased Bots: Conversational Agents to Overcome Polarization". In: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. UbiComp '18. Singapore, Singapore: ACM, 2018, pp. 1664–1668. ISBN: 978-1-4503-5966-5. DOI: 10.1145/3267305.3274189. URL: http://doi.acm.org/10.1145/3267305.3274189.

[229] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. *Generalization and Robustness Implications in Object-Centric Learning*. 2021. DOI: 10.48550/ARXIV.2107.00637. URL: https://arxiv.org/abs/2107.00637.

[230] Roel Dobbe, Thomas Krendl Gilbert, and Yonatan Mintz. "Hard choices in artificial intelligence". In: *Artificial Intelligence* 300 (2021), p. 103555.

[231] Roel IJ Dobbe, Thomas Krendl Gilbert, and Yonatan Mintz. "Hard Choices in Artificial Intelligence: Addressing Normative Uncertainty through Sociotechnical Commitments Proceedings of the AAAI". In: *ACM Conference on AI, Ethics, and Society*. 2020.

[232] Finale Doshi-Velez and Been Kim. "Considerations for evaluation and generalization in interpretable machine learning". In: *Explainable and interpretable models in computer vision and machine learning*. Springer, 2018, pp. 3–17.

[233] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017).

[234] Ravit Dotan and Smitha Milli. “Value-laden disciplinary shifts in machine learning”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 294–294.

[235] D. M. Downs and G. Cowan. “Predicting the importance of freedom of speech and the perceived harm of hate speech”. In: *Journal of applied social psychology* 42.6 (2012), pp. 1353–1375.

[236] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. “Robustness in Deep Learning for Computer Vision: Mind the gap?” In: *arXiv preprint arXiv:2112.00639* (2021).

[237] Alexey Drutsa, Valentina Fedorova, Dmitry Ustalov, Olga Megorskaya, Evfrosiniya Zerminova, and Daria Baidakova. “Crowdsourcing practice for efficient data labeling: Aggregation, incremental relabeling, and pricing”. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020, pp. 2623–2627.

[238] Chris Dulhanty and Alexander Wong. “Auditing imagenet: Towards a model-driven framework for annotating demographic attributes of large-scale image datasets”. In: *arXiv preprint arXiv:1905.01347* (2019).

[239] Sanghamitra Dutta, Dennis Wei, Hazar Yueksel, Pin-Yu Chen, Sijia Liu, and Kush Varshney. “Is there a trade-off between fairness and accuracy? a perspective using mismatched hypothesis testing”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 2803–2813.

[240] Yogesh K Dwivedi, Laurie Hughes, Elvira Ismagilova, Gert Aarts, Crispin Coombs, Tom Crick, Yanqing Duan, Rohita Dwivedi, John Edwards, Aled Eirug, et al. “Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy”. In: *International Journal of Information Management* 57 (2021), p. 101994.

[241] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. “Fairness through awareness”. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. ACM. 2012, pp. 214–226.

[242] Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. “At the Lower End of Language—Exploring the Vulgar and Obscene Side of German”. In: *Proceedings of the Third Workshop on Abusive Language Online*. 2019, pp. 119–128.

[243] Upol Ehsan, Q Vera Liao, Samir Passi, Mark O Riedl, and Hal Daume III. “Seamful XAI: Operationalizing Seamful Design in Explainable AI”. In: *arXiv preprint arXiv:2211.06753* (2022).

[244] Upol Ehsan and Mark O Riedl. “Human-centered explainable ai: Towards a reflective sociotechnical approach”. In: *International Conference on Human-Computer Interaction*. Springer. 2020, pp. 449–466.

[245] Shady Elbassuoni, Sihem Amer-Yahia, Christine El Atie, Ahmad Ghizzawi, and Bilel Oualha. “Exploring Fairness of Ranking in Online Job Marketplaces”. In: *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*. 2019, pp. 646–649. DOI: 10.5441/002/edbt.2019.77. URL: https://doi.org/10.5441/002/edbt.2019.77.

[246] Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. “Hate lingo: A target-based linguistic analysis of hate speech in social media”. In: *arXiv preprint arXiv:1804.04257* (2018).

[247] Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. “Peer to peer hate: Hate speech instigators and their targets”. In: *arXiv preprint arXiv:1804.04649* (2018).

[248] Anders Eriksson and Francisco Lacerda. “Charlatanry in forensic speech science: A problem to be taken seriously”. In: *International Journal of Speech, Language and the Law* 14.2 (2007), pp. 169–193.

[249] Eva Eriksson, Elisabet M Nilsson, Anne-Marie Hansen, and Tilde Bekker. “Teaching for Values in Human–Computer Interaction”. In: *Frontiers in Computer Science* 4 (2022).

[250] S. C. Eshan and M. S. Hasan. “An application of machine learning to detect abusive Bengali text”. In: *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, Dec. 2017. ISBN: 978-1-5386-1150-0. DOI: 10.1109/ICCITECHN.2017.8281787. URL: http://ieeexplore.ieee.org/document/8281787/.

[251] Council of Europe: European Commission against Racism and Intolerance (ECRI). *Hate speech and violence*. https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/hate-speech-and-violence, Last accessed on 2020-03-16.

[252] Golnoosh Farnadi, Behrouz Babaki, and Lise Getoor. "A Declarative Approach to Fairness in Relational Domains". In: *Data Engineering* (2019), p. 36.

[253] Golnoosh Farnadi, Behrouz Babaki, and Lise Getoor. "Fairness in Relational Domains". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '18. New Orleans, LA, USA: Association for Computing Machinery, 2018, pp. 108–114. ISBN: 9781450360128. DOI: 10.1145/3278721.3278733. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3278721.3278733.

[254] Sina Fazelpour and Zachary C Lipton. "Algorithmic fairness from a non-ideal perspective". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 57–63.

[255] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. "Certifying and Removing Disparate Impact". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. 2015, pp. 259–268. DOI: 10.1145/2783258.2783311. URL: https://doi.org/10.1145/2783258.2783311.

[256] Martinez-Plumed Fernando, Ferri Cèsar, Nieves David, and Hernández-Orallo José. "Missing the missing values: The ugly duckling of fairness in machine learning". In: *International Journal of Intelligent Systems* 36.7 (2021), pp. 3217–3258.

[257] Tobias Fiebig, Seda F. Gürses, Carlos Hernandez Gañán, Erna Kotkamp, Fernando Kuipers, Martina Lindorfer, Menghua Prisse, and Taritha Sari. "Heads in the Clouds: Measuring the Implications of Universities Migrating to Public Clouds". In: *CoRR* abs/2104.09462 (2021). arXiv: 2104.09462. URL: https://arxiv.org/abs/2104.09462.

[258] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. "Human model evaluation in interactive supervised learning". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2011, pp. 147–156.

[259] Ailbhe Finnerty, Pavel Kucherbaev, Stefano Tranquillini, and Gregorio Convertino. "Keep it simple: Reward and task design in crowdsourcing". In: *Italian Chapter of SIGCHI*. 2013, pp. 1–4.

[260] Jessie Finocchiaro, Roland Maio, Faidra Monachou, Gourab K Patro, Manish Raghavan, Ana-Andreea Stoica, and Stratis Tsirtsis. "Bridging machine learning and mechanism design towards algorithmic fairness". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 489–503.

[261] Luciano Floridi. "Establishing the rules for building trustworthy AI". In: *Nature Machine Intelligence* 1.6 (2019), pp. 261–262.

[262] Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. "Time of your hate: The challenge of time in hate speech detection on social media". In: *Applied Sciences* 10.12 (2020), p. 4180.

[263] P. Fortuna and S. Nunes. "A Survey on Automatic Detection of Hate Speech in Text". In: *ACM Comput. Surv.* 51.4 (July 2018), 85:1–85:30. ISSN: 0360-0300. DOI: 10.1145/3232676. URL: http://doi.acm.org/10.1145/3232676.

[264] Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. "A Unified Deep Learning Architecture for Abuse Detection". In: *Proceedings of the 10th ACM Conference on Web Science*. ACM. 2019, pp. 105–114.

[265] Gordon Fraser and JM Rojas. "Software Testing". In: *Handbook of Software Engineering*. Cham: Springer, 2019, pp. 123–192. ISBN: 978-3-030-00262-6. DOI: 10.1007/978-3-030-00262-6_4. URL: https://doi.org/10.1007/978-3-030-00262-6_4.

[266] Alex A Freitas. "Comprehensible classification models: a position paper". In: *ACM SIGKDD explorations newsletter* 15.1 (2014), pp. 1–10.

[267] Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. "The (im) possibility of fairness: Different value systems require different mechanisms for fair decision making". In: *Communications of the ACM* 64.4 (2021), pp. 136–143.

[268] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. "A comparative study of fairness-enhancing interventions in machine learning". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 329–338.

[269] Heidi Furey and Fred Martin. “AI education matters: a modular approach to AI ethics education”. In: *AI Matters* 4.4 (2019), pp. 13–15.

[270] Ajit G. Pillai, A Baki Kocaballi, Tuck Wah Leong, Rafael A. Calvo, Nassim Parvin, Katie Shilton, Jenny Waycott, Casey Fiesler, John C. Havens, and Naseem Ahmadpour. “Co-designing resources for ethics education in HCI”. In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–5.

[271] Ujwal Gadiraju and Jie Yang. “What can crowd computing do for the next generation of AI systems?” In: *2020 Crowd Science Workshop*. CEUR. 2020, pp. 7–13.

[272] Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. *Countering online hate speech*. Unesco Publishing, 2015.

[273] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. “Fairness Testing: Testing Software for Discrimination”. In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ESEC/FSE 2017. Paderborn, Germany: ACM, 2017, pp. 498–510. ISBN: 978-1-4503-5105-8. DOI: 10.1145/3106237.3106277. URL: http://doi.acm.org/10.1145/3106237.3106277.

[274] B. Gambäck and U. K. Sikdar. “Using Convolutional Neural Networks to Classify Hate-Speech”. In: *Proceedings of the First Workshop on Abusive Language Online*. 2017, pp. 85–90.

[275] L. Gao and R. Huang. “Detecting Online Hate Speech Using Context Aware Models”. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. 2017.

[276] L. Gao, A. Kuppersmith, and R. Huang. “Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach”. In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing*. Vol. 1. 2017, pp. 774–782.

[277] Tianyu Gao, Xingcheng Yao, and Danqi Chen. “SimCSE: Simple Contrastive Learning of Sentence Embeddings”. In: *EMNLP 2021*. ACL, 2021, pp. 6894–6910.

[278] Álvaro Garcia-Recuero, Jeffrey Burdges, and Christian Grothoff. “Privacy-preserving abuse detection in future decentralised online social networks”. In: *Data Privacy Management and Security Assurance*. Springer, 2016, pp. 78–93.

[279] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. “Word embeddings quantify 100 years of gender and ethnic stereotypes”. In: *Proceedings of the National Academy of Sciences* 115.16 (2018), E3635–E3644.

[280] Vahid Garousi, Gorkem Giray, Eray Tuzun, Cagatay Catal, and Michael Felderer. “Closing the gap between software engineering education and industrial needs”. In: *IEEE software* 37.2 (2019), pp. 68–77.

[281] Natalie Garrett, Nathan Beard, and Casey Fiesler. “More Than" If Time Allows" The Role of Ethics in AI Education”. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 272–278.

[282] William Gaver. “What should we expect from research through design?” In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 937–946.

[283] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. “Datasheets for datasets”. In: *Communications of the ACM* 64.12 (2021), pp. 86–92.

[284] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.

[285] Daniel M. German, Gregorio Robles, Germán Poo-Caamaño, Xin Yang, Hajimu Iida, and Katsuro Inoue. ““Was My Contribution Fairly Reviewed?": A Framework to Study the Perception of Fairness in Modern Code Reviews”. In: *Proceedings of the 40th International Conference on Software Engineering*. ICSE ’18. Gothenburg, Sweden: ACM, 2018, pp. 523–534. ISBN: 978-1-4503-5638-1. DOI: 10.1145/3180155.3180217. URL: http://doi.acm.org/10.1145/3180155.3180217.

[286] Amy Herstein Gervasio and Katy Ruckdeschel. “College Students' judgments of verbal sexual harassment 1”. In: *Journal of Applied Social Psychology* 22.3 (1992), pp. 190–211.

[287] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. "Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 2019, pp. 2221–2231. DOI: 10.1145/3292500.3330691. URL: https://doi.org/10.1145/3292500.3330691.

[288] Ahmad Ghizzawi, Julien Marinescu, Shady Elbassuoni, Sihem Amer-Yahia, and Gilles Bisson. "FaiRank: An Interactive System to Explore Fairness of Ranking in Online Job Marketplaces". In: *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*. 2019, pp. 582–585. DOI: 10.5441/002/edbt.2019.61. URL: https://doi.org/10.5441/002/edbt.2019.61.

[289] A Ghorbani and al. "Towards automatic concept-based explanations". In: *NeurIPS*. 2019.

[290] Sanjukta Ghosh, Rohan Shet, Peter Amon, Andreas Hutter, and André Kaup. "Robustness of Deep Convolutional Neural Networks for Image Degradations". In: *ICASSP*. 2018, pp. 2916–2920. DOI: 10.1109/ICASSP.2018.8461907.

[291] Fabio Giglietto and Yenn Lee. "To Be or Not to Be Charlie: Twitter hashtags as a discourse and counter-discourse in the aftermath of the 2015 Charlie Hebdo shooting in France". In: *Proceedings of the 5th Workshop on Making Sense of Microposts co-located with the 24th International World Wide Web Conference*. 2015, pp. 33–37.

[292] Yolanda Gil. "Teaching big data analytics skills with intelligent workflow systems". In: *AAAI*. Vol. 30. 1. 2016.

[293] Görkem Giray. "A software engineering perspective on engineering machine learning systems: State of the art and challenges". In: *Journal of Systems and Software* 180 (2021), p. 111031.

[294] Bruce Glymour and Jonathan Herington. "Measuring the Biases That Matter: The Ethical and Casual Foundations for Measures of Fairness in Algorithms". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. FAT* '19. Atlanta, GA, USA: ACM, 2019, pp. 269–278. ISBN: 978-1-4503-6125-5. DOI: 10.1145/3287560.3287573. URL: http://doi.acm.org/10.1145/3287560.3287573.

[295] Tejas Gokhale, Swaroop Mishra, Man Luo, Bhavdeep Singh Sachdeva, and Chitta Baral. "Generalized but not Robust? Comparing the Effects of Data Modification Methods on Out-of-Domain Generalization and Adversarial Robustness". In: *arXiv preprint arXiv:2203.07653* (2022).

[296] Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, et al. "A large labeled corpus for online harassment research". In: *Proceedings of the 2017 ACM on web science conference*. 2017, pp. 229–233.

[297] Sixue Gong, Xiaoming Liu, and Anil K Jain. "Jointly de-biasing face recognition and demographic attribute estimation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 330–347.

[298] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. "Making Machine Learning Robust against Adversarial Inputs". In: *Commun. ACM* 61.7 (June 2018), pp. 56–66. ISSN: 0001-0782. DOI: 10.1145/3134599. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3134599.

[299] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI magazine* 38.3 (2017), pp. 50–57.

[300] Leo A Goodman. "Snowball sampling". In: *The annals of mathematical statistics* (1961), pp. 148–170.

[301] Robert Gorwa, Reuben Binns, and Christian Katzenbach. "Algorithmic content moderation: Technical and political challenges in the automation of platform governance". In: *Big Data & Society* 7.1 (2020), p. 2053951719897945.

[302] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. "Counterfactual visual explanations". In: *ICML*. PMLR. 2019, pp. 2376–2384.

[303] Stefan Grafberger, Paul Groth, Julia Stoyanovich, and Sebastian Schelter. "Data distribution debugging in machine learning pipelines". In: *The VLDB Journal* (2022), pp. 1–24.

[304] Stefan Grafberger, Julia Stoyanovich, and Sebastian Schelter. "Lightweight Inspection of Data Preprocessing in Native Machine Learning Pipelines." In: *CIDR*. 2021.

[305] Corrado Grappiolo, Héctor P Martinez, and Georgios N Yannakakis. “Validating Generic Metrics of Fairness in Game-based Resource Allocation Scenarios with Crowdsourced Annotations”. In: *Transactions on Computational Intelligence XIII*. Springer, 2014, pp. 176–200.

[306] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech recognition with deep recurrent neural networks”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[307] Colin M Gray, Erik Stolterman, and Martin A Siegel. “Reprioritizing the relationship between HCI research and practice: bubble-up and trickle-down effects”. In: *Proceedings of the 2014 conference on Designing interactive systems*. 2014, pp. 725–734.

[308] Ben Green. “Escaping the" Impossibility of Fairness": From Formal to Substantive Algorithmic Fairness”. In: *arXiv preprint arXiv:2107.04642* (2021).

[309] Ben Green. “The contestation of tech ethics: A sociotechnical approach to technology ethics in practice”. In: *Journal of Social Computing* 2.3 (2021), pp. 209–225.

[310] Ben Green and Lily Hu. “The myth in the methodology: Towards a recontextualization of fairness in machine learning”. In: *Proceedings of the machine learning: the debates workshop*. 2018.

[311] Ben Green and Salomé Viljoen. “Algorithmic realism: expanding the boundaries of algorithmic thought”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 19–31.

[312] Daniel Greene, Anna Lauren Hoffmann, and Luke Stark. “Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning”. In: (2019).

[313] Nina Grgić-Hlača, Christoph Engel, and Krishna P Gummadi. “Human decision making with machine assistance: An experiment on bailing and jailing”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–25.

[314] Nina Grgić-Hlača, Gabriel Lima, Adrian Weller, and Elissa M Redmiles. “Dimensions of Diversity in Human Perceptions of Algorithmic Fairness”. In: *ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. ACM. 2022, pp. 1–12.

[315] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. “Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[316] Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. “All You Need is" Love" Evading Hate Speech Detection”. In: *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. 2018, pp. 2–12.

[317] Yifan Guan, Abolfazl Asudeh, Pranav Mayuram, H. V. Jagadish, Julia Stoyanovich, Gerome Miklau, and Gautam Das. “MithraRanking: A System for Responsible Ranking Design”. In: *Proceedings of the 2019 International Conference on Management of Data*. SIGMOD ’19. Amsterdam, Netherlands: ACM, 2019, pp. 1913–1916. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3320244. URL: http://doi.acm.org/10.1145/3299869.3320244.

[318] J. Guberman and L. Hemphill. “Challenges in modifying existing scales for detecting harassment in individual tweets”. In: *Proceedings of the 50th Hawaii International Conference on System Sciences*. 2017.

[319] Amos Guiora and Elizabeth A Park. “Hate speech on social media”. In: *Philosophia* 45.3 (2017), pp. 957–971.

[320] David Gundry and Sebastian Deterding. “Intrinsic elicitation: A model and design approach for games collecting human subject data”. In: *13th Intl. Conf. on the Foundations of Digital Games*. 2018, pp. 1–10.

[321] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. “Vizwiz grand challenge: Answering visual questions from blind people”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3608–3617.

[322] Seda Gurses and Joris Van Hoboken. “Privacy after the agile turn”. In: *SocArXiv* ().

[323] I. Guy and B. Shapira. "From Royals to Vegans: Characterizing Question Trolling on a Community Question Answering Website". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18*. New York, USA: ACM Press, 2018, pp. 835–844. ISBN: 9781450356572. DOI: 10.1145/3209978.3210058. URL: http://dl.acm.org/citation.cfm?doid=3209978.3210058.

[324] Bushr Haddad, Zoher Orabe, Anas Al-Abood, and Nada Ghneim. "Arabic Offensive Language Detection with Attention-based Deep Neural Networks". In: *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*. 2020.

[325] Thilo Hagendorff. "Forbidden knowledge in machine learning reflections on the limits of research and publication". In: *AI & SOCIETY* 36.3 (2021), pp. 767–781.

[326] Thilo Hagendorff. "The ethics of AI ethics: An evaluation of guidelines". In: *Minds and Machines* 30.1 (2020), pp. 99–120.

[327] Thilo Hagendorff and Katharina Wezel. "15 challenges for AI: or what AI (currently) can't do". In: *AI & SOCIETY* 35 (2020), pp. 355–365.

[328] Brent Hailpern and Padmanabhan Santhanam. "Software debugging, testing, and verification". In: *IBM Systems Journal* 41.1 (2002), pp. 4–12.

[329] Sara Hajian, Francesco Bonchi, and Carlos Castillo. "Algorithmic bias: From discrimination discovery to fairness-aware data mining". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 2125–2126.

[330] Sara Hajian and Josep Domingo-Ferrer. "A methodology for direct and indirect discrimination prevention in data mining". In: *IEEE transactions on knowledge and data engineering* 25.7 (2012), pp. 1445–1459.

[331] Sara Hajian, Josep Domingo-Ferrer, and Oriol Farràs. "Generalization-based privacy preservation and discrimination prevention in data publishing and mining". In: *Data Min. Knowl. Discov.* 28.5-6 (2014), pp. 1158–1188. DOI: 10.1007/s10618-014-0346-1. URL: https://doi.org/10.1007/s10618-014-0346-1.

[332] Sara Hajian, Josep Domingo-Ferrer, Anna Monreale, Dino Pedreschi, and Fosca Giannotti. "Discrimination- and privacy-aware patterns". In: *Data Min. Knowl. Discov.* 29.6 (2015), pp. 1733–1782. DOI: 10.1007/s10618-014-0393-7. URL: https://doi.org/10.1007/s10618-014-0393-7.

[333] Jerold L Hale, Brian J Householder, and Kathryn L Greene. "The theory of reasoned action". In: *The persuasion handbook: Developments in theory and practice* 14.2002 (2002), pp. 259–286.

[334] Lei Han, Xiao Dong, and Gianluca Demartini. "Iterative Human-in-the-Loop Discovery of Unknown Unknowns in Image Datasets". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 9. 2021, pp. 72–83.

[335] Alex Hanna, Emily Denton, Andrew Smart, and Jamila Smith-Loud. "Towards a critical race methodology in algorithmic fairness". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 501–512.

[336] Moritz Hardt, Eric Price, Nati Srebro, et al. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems*. 2016, pp. 3315–3323.

[337] Sharon L Harlan, David N Pellow, J Timmons Roberts, Shannon Elizabeth Bell, William G Holt, and Joane Nagel. "Climate justice and inequality". In: *Climate change and society: Sociological perspectives* (2015), pp. 127–163.

[338] Galen Harrison, Julia Hanson, Christine Jacinto, Julio Ramirez, and Blase Ur. "An empirical study on the perceived fairness of realistic, imperfect machine learning models". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 392–402.

[339] M. Hasanuzzaman, G. Dias, and A. Way. "Demographic Word Embeddings for Racism Detection on Twitter". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing* 1 (2017), pp. 926–936. URL: https://aclanthology.info/papers/I17-1093/i17-1093.

[340] Trevor J Hastie. "Generalized additive models". In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[341] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, et al. "Applied machine learning at facebook: A datacenter infrastructure perspective". In: *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2018, pp. 620–629.

[342] MEPS HC. "181: 2015 Full Year Consolidated Data File". In: *Agency for Healthcare Research and Quality* (2017).

[343] Gaole He, Agathe Balayn, Stefan Buijsman, Jie Yang, and Ujwal Gadiraju. "It Is Like Finding a Polar Bear in the Savannah! Concept-Level AI Explanations with Analogical Inference from Commonsense Knowledge". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. Vol. 10. 1. 2022, pp. 89–101.

[344] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. "Bag of tricks for image classification with convolutional neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 558–567.

[345] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.

[346] Amy Heger, Elizabeth B Marquis, Mihaela Vorvoreanu, Hanna Wallach, and Jennifer Wortman Vaughan. "Understanding Machine Learning Practitioners' Data Documentation Perceptions, Needs, Challenges, and Desiderata". In: *arXiv preprint arXiv:2206.02923* (2022).

[347] Hoda Heidari, Michele Loi, Krishna P Gummadi, and Andreas Krause. "A moral framework for understanding fair ml through economic models of equality of opportunity". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 181–190.

[348] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. "Women also snowboard: Overcoming bias in captioning models". In: *European Conference on Computer Vision*. Springer. 2018, pp. 793–811.

[349] Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. "Machine learning with a reject option: A survey". In: *arXiv preprint arXiv:2107.11277* (2021).

[350] Dan Hendrycks, Steven Basart, et al. "The many faces of robustness: A critical analysis of out-of-distribution generalization". In: *ICCV*. 2021, pp. 8340–8349.

[351] Dan Hendrycks and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations*. 2018.

[352] Gary T Henry. "Practical sampling". In: *The SAGE handbook of applied social research methods* 2 (2009), pp. 77–105.

[353] P.J. Henry, S. E. Butler, and M. J. Brandt. "The influence of target group status on the perception of the offensiveness of group-based slurs". In: *Journal of Experimental Social Psychology* 53 (2014).

[354] Amac Herdagdelen and Marco Baroni. "The concept game: Better commonsense knowledge extraction by combining text mining and a game with a purpose". In: *2010 AAAI Fall Symposium Series*. 2010.

[355] Jerónimo Hernández-González, Inaki Inza, and Jose A Lozano. "A note on the behavior of majority voting in multi-class domains with biased annotators". In: *IEEE Transactions on Knowledge and Data Engineering* 31.1 (2018), pp. 195–200.

[356] Luis Herranz, Shuqiang Jiang, and Xiangyang Li. "Scene recognition with CNNs: objects, scales and dataset bias". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 571–579.

[357] Sarah Hewitt, Thanassis Tiropanis, and C. Bokhove. "The problem of identifying misogynist language on Twitter (and other online social spaces)". In: *Proceedings of the 8th ACM Conference on Web Science, WebSci 2016, Hannover, Germany, May 22-25, 2016*. Ed. by Wolfgang Nejdl, Wendy Hall, Paolo Parigi, and Steffen Staab. ACM, 2016, pp. 333–335. ISBN: 978-1-4503-4208-7. DOI: 10.1145/2908131.2908183. URL: https://doi.org/10.1145/2908131.2908183.

[358] Johannes Himmelreich. "Against "Democratizing AI"". In: *AI & SOCIETY* (2022), pp. 1–14.

[359] P Hitzler and MK Sarker. "Human-Centered Concept Explanations for Neural Networks". In: *Neuro-Symbolic Artificial Intelligence: The State of the Art* 342.337 (2022), p. 2.

[360] Arthur E Hoerl and Robert W Kennard. "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1 (1970), pp. 55–67.

[361] Anna Lauren Hoffmann. "Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse". In: *Information, Communication & Society* 22.7 (2019), pp. 900–915.

[362] Anna Lauren Hoffmann and Katherine Alejandra Cross. "Teaching data ethics: Foundations and possibilities from engineering and computer science ethics education". In: (2021).

[363] Professor Dr. Hans Hofmann. *Statlog (German Credit Data) Data Set*. 1994. URL: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[364] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. "Knowledge graphs". In: *ACM Computing Surveys (CSUR)* 54.4 (2021), pp. 1–37.

[365] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. "Gamut: A design probe to understand how data scientists understand machine learning models". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–13.

[366] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng Polo Chau. "Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations". In: *Trans. on visualization and computer graphics* 26.1 (2019), pp. 1096–1106.

[367] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. "Understanding and visualizing data iteration in machine learning". In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–13.

[368] Derek Hoiem, Santosh K Divvala, and James H Hays. "Pascal VOC 2008 challenge". In: *PASCAL challenge workshop in ECCV*. Citeseer. 2009.

[369] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. "Improving fairness in machine learning systems: What do industry practitioners need?" In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–16.

[370] Tobias Holstein and Gordana Dodig-Crnkovic. "Avoiding the Intrinsic Unfairness of the Trolley Problem". In: *Proceedings of the International Workshop on Software Fairness*. FairWare '18. Gothenburg, Sweden: ACM, 2018, pp. 32–37. ISBN: 978-1-4503-5746-3. DOI: 10.1145/3194770.3194772. URL: http://doi.acm.org/10.1145/3194770.3194772.

[371] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. "Human factors in model interpretability: Industry practices, challenges, and needs". In: *ACM on Human-Computer Interaction* 4.CSCW1 (2020), pp. 1–26.

[372] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. "Characterising bias in compressed models". In: *arXiv preprint arXiv:2010.03058* (2020).

[373] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q Lv, and S. Mishra. "Detection of Cyberbullying Incidents on the Instagram Social Network". In: (Mar. 2015). arXiv: 1503.03909. URL: http://arxiv.org/abs/1503.03909.

[374] H. Hosseinmardi, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra. "Prediction of cyberbullying incidents in a media-based social network". In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, Aug. 2016, pp. 186–192. ISBN: 978-1-5090-2846-7. DOI: 10.1109/ASONAM.2016.7752233. URL: http://ieeexplore.ieee.org/document/7752233/.

[375] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. "Analyzing labeled cyberbullying incidents on the instagram social network". In: *International conference on social informatics*. Springer. 2015, pp. 49–66.

[376] Lotte Houwing. "Stop the Creep of Biometric Surveillance Technology". In: *Eur. Data Prot. L. Rev.* 6 (2020), p. 174.

[377] X Hu, H Wang, A Vegesana, and al. "Crowdsourcing Detection of Sampling Biases in Image Datasets". In: *Proc. of WWW*. 2020, pp. 2955–2961.

[378] Jin Huang and Charles X Ling. "Using AUC and accuracy in evaluating learning algorithms". In: *IEEE Transactions on knowledge and Data Engineering* 17.3 (2005), pp. 299–310.

[379] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. "Cosmos QA: Machine Reading Comprehension with Contextual Commonsense Reasoning". In: *2019 EMNLP-IJCNLP*. 2019, pp. 2391–2401.

[380] Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. "Cyber Bullying Detection Using Social and Textual Analysis". In: *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*. ACM. 2014, pp. 3–6.

[381] Christoph Hube, Besnik Fetahu, and Ujwal Gadiraju. "Understanding and mitigating worker biases in the crowdsourced collection of subjective judgments". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–12.

[382] Waqar Hussain, Davoud Mougouei, and Jon Whittle. "Integrating Social Values into Software Design Patterns". In: *Proceedings of the International Workshop on Software Fairness*. FairWare '18. Gothenburg, Sweden: ACM, 2018, pp. 8–14. ISBN: 978-1-4503-5746-3. DOI: 10.1145/3194770.3194777. URL: http://doi.acm.org/10.1145/3194770.3194777.

[383] Ben Hutchinson and Margaret Mitchell. "50 Years of Test (Un) fairness: Lessons for Machine Learning". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. 2019, pp. 49–58.

[384] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, et al. "Technology probes: inspiring design for and with families". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2003, pp. 17–24.

[385] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[386] M. O. Ibrohim and I. Budi. "A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media". In: *Procedia Computer Science* 135 (2018), pp. 222–229. ISSN: 18770509. DOI: 10.1016/j.procs.2018.08.169. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877050918314583.

[387] I. Iglezakis. "The Legal Regulation of Hate Speech on the Internet". In: *EU Internet Law*. Springer, 2017.

[388] Filip Ilievski, Pedro A. Szekely, Jingwei Cheng, Fu Zhang, and Ehsan Qasemi. "Consolidating Commonsense Knowledge". In: *CoRR* abs/2006.06114 (2020). arXiv: 2006.06114. URL: https://arxiv.org/abs/2006.06114.

[389] Filip Ilievski, Pedro A. Szekely, and Bin Zhang. "CSKG: The CommonSense Knowledge Graph". In: *The Semantic Web - 18th Intl. Conf., ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*. Vol. 12731. Lecture Notes in Computer Science. Springer, 2021, pp. 680–696.

[390] Nasif Imtiaz, Justin Middleton, Joymallya Chakraborty, Neill Robson, Gina Bai, and Emerson R. Murphy-Hill. "Investigating the effects of gender bias on GitHub". In: *Proceedings of the 41st International Conference on Software Engineering, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019*. 2019, pp. 700–711. DOI: 10.1109/ICSE.2019.00079. URL: https://doi.org/10.1109/ICSE.2019.00079.

[391] Oana Inel, Khalid Khamkham, Tatiana Cristea, Anca Dumitrache, Arne Rutjes, Jelle van der Ploeg, Lukasz Romaszko, Lora Aroyo, and Robert-Jan Sips. "Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data". In: *ISWC*. Springer. 2014, pp. 486–504.

[392] A. Ioannou, J. Blackburn, G. Siringhini, E. De Chrisiofaro, N. Kouriellis, M. Sirivianos, and P. Zaphiris. "From risk factors to detection and intervention: A metareview and practical proposal for research on cyberbullying". In: *2017 IST-Africa Week Conference (IST-Africa)*. IEEE. 2017, pp. 1–8.

[393] Lilly Irani. "Difference and dependence among digital workers: The case of Amazon Mechanical Turk". In: *South Atlantic Quarterly* 114.1 (2015), pp. 225–234.

[394] Lilly Irani. "Justice for data janitors". In: *Think in Public*. Columbia University Press, 2019, pp. 23–40.

[395] Fuyuki Ishikawa and Nobukazu Yoshioka. "How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey". In: *CESI and SER&IP workshops*. IEEE. 2019, pp. 2–9.

[396] Md Johirul Islam, Giang Nguyen, Rangeet Pan, and Hridesh Rajan. "A comprehensive study on deep learning bug characteristics". In: *ESEC and Foundations of Software Engineering*. 2019, pp. 510–520.

[397] Rashidul Islam, Shimei Pan, and James R Foulds. "Can We Obtain Fairness For Free?" In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 586–596.

[398] Vladimir Ivanov, Alan Rogers, Giancarlo Succi, Jooyong Yi, and Vasilii Zorin. "What do software engineers care about? gaps between research and practice". In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. 2017, pp. 890–895.

[399] Hanen Jabnoun, Faouzi Benzarti, and Hamid Amiri. "Visual scene prediction for blind people based on object recognition". In: *2017 14th International Conference on Computer Graphics, Imaging and Visualization*. IEEE. 2017, pp. 21–26.

[400] Michael Jackson. "The world and the machine". In: *Proceedings of the 17th international conference on Software engineering*. 1995, pp. 283–292.

[401] Abigail Z Jacobs and Hanna Wallach. "Measurement and fairness". In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 375–385.

[402] HV Jagadish, Francesco Bonchi, Tina Eliassi-Rad, Lise Getoor, Krishna Gummadi, and Julia Stoyanovich. "The Responsibility Challenge for Data". In: *Proceedings of the 2019 International Conference on Management of Data*. 2019, pp. 412–414.

[403] James Vincent. *Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day*. https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist (Feb. 2020).

[404] Rabih Jamil and Yanick Noiseux. "Shake that moneymaker: insights from Montreal's Uber drivers". In: *Revue Interventions Économiques. Papers in Political Economy* 60 (2018).

[405] Yeonju Jang, Seongyune Choi, and Hyeoncheol Kim. "Development and validation of an instrument to measure undergraduate students' attitudes toward the ethics of artificial intelligence (AT-EAI) and analysis of its difference by gender and experience of AI education". In: *Education and Information Technologies* (2022), pp. 1–33.

[406] Dietmar Jannach, Iman Kamehkhosh, and Geoffray Bonnin. "Biases in Automated Music Playlist Generation: A Comparison of Next-Track Recommending Techniques". In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. UMAP '16. Halifax, Nova Scotia, Canada: ACM, 2016, pp. 281–285. ISBN: 978-1-4503-4368-8. DOI: 10.1145/2930238.2930283. URL: http://doi.acm.org/10.1145/2930238.2930283.

[407] Nathalie Japkowicz. "Why question machine learning evaluation methods". In: *AAAI workshop on evaluation methods for machine learning*. 2006, pp. 6–11.

[408] Jeffrey Dastin. *Amazon scraps secret AI recruiting tool that showed bias against women*. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-airecruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G (Jan. 2020).

[409] Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, P Saleiro, P Bizarro, and J Gama. "How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations". In: *2021 ACM FAccT*. 2021, pp. 805–815.

[410] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. "Is bert really robust? a strong baseline for natural language attack on text classification and entailment". In: *AAAI*. Vol. 34. 05. 2020, pp. 8018–8025.

[411] Zhongjun Jin, Mengjing Xu, Chenkai Sun, Abolfazl Asudeh, and H. V. Jagadish. "MithraCoverage: A System for Investigating Population Bias for Intersectional Fairness". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD '20. Portland, OR, USA: Association for Computing Machinery, 2020, pp. 2721–2724. ISBN: 9781450367356. DOI: 10.1145/3318464.3384689. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3318464.3384689.

[412] Anna Jobin, Marcello Ienca, and Effy Vayena. "The global landscape of AI ethics guidelines". In: *Nature Machine Intelligence* 1.9 (2019), pp. 389–399.

[413] Adam J Johs, Denise E Agosto, and Rosina O Weber. "Explainable artificial intelligence and social science: Further insights for qualitative investigation". In: *Applied AI Letters* 3.1 (2022), e64.

[414] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[415] David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. “A Just and Comprehensive Strategy for Using NLP to Address Online Abuse”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 3658–3666.

[416] Singh V. K., Huang Q., and Atrey P. K. “Cyberbullying detection using probabilistic socio-textual information fusion”. In: *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, Aug. 2016, pp. 884–887. ISBN: 978-1-5090-2846-7. DOI: 10.1109/ASONAM.2016.7752342. URL: http://ieeexplore.ieee.org/document/7752342/.

[417] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. “Model-based and actual independence for fairness-aware classification”. In: *Data Min. Knowl. Discov.* 32.1 (2018), pp. 258–286. DOI: 10.1007/s10618-017-0534-x. URL: https://doi.org/10.1007/s10618-017-0534-x.

[418] Byunggu Kang and Sishi Wu. “False positives vs. false negatives: public opinion on the cost ratio in criminal justice risk assessment”. In: *Journal of Experimental Criminology* (2022), pp. 1–23.

[419] Daniel Kang, Deepti Raghavan, Peter Bailis, and Matei Zaharia. “Model assertions for debugging machine learning”. In: *NeurIPS MLSys Workshop*. 2018.

[420] Shivani Kapania, Oliver Siy, Gabe Clapper, Azhagu Meena SP, and Nithya Sambasivan. “” Because AI is 100% right and safe”: User Attitudes and Sources of AI Authority in India”. In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–18.

[421] Raghav Kapoor, Yaman Kumar, Kshitij Rajput, Rajiv Ratn Shah, Ponnurangam Kumaraguru, and Roger Zimmermann. “Mind your language: Abuse and offense detection for code-switched languages”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 9951–9952.

[422] Chen Karako and Putra Manggala. “Using Image Fairness Representations in Diversity-Based Reranking for Recommendations”. In: *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. UMAP ’18. Singapore, Singapore: ACM, 2018, pp. 23–28. ISBN: 978-1-4503-5784-5. DOI: 10.1145/3213586.3226206. URL: http://doi.acm.org/10.1145/3213586.3226206.

[423] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. “A survey of algorithmic recourse: contrastive explanations and consequential recommendations”. In: *ACM Computing Surveys* 55.5 (2022), pp. 1–29.

[424] Andrej Karpathy. *A Recipe for Training Neural Networks*. Apr. 2019. URL: http://karpathy.github.io/2019/04/25/recipe/.

[425] Maria Kasinidou, Styliani Kleanthous, Pınar Barlas, and Jahna Otterbacher. “I agree with the decision, but they didn’t deserve this: Future Developers’ Perception of Fairness in Algorithmic Decisions”. In: *Proceedings of the 2021 acm conference on fairness, accountability, and transparency*. 2021, pp. 690–700.

[426] Masahiro Kato, Zhenghang Cui, and Yoshihiro Fukuhara. “Atro: Adversarial training with a rejection option”. In: *arXiv preprint arXiv:2010.12905* (2020).

[427] Harmanpreet Kaur et al. “Interpreting Interpretability: Understanding Data Scientists’ Use of Interpretability Tools for Machine Learning”. In: *CHI*. 2020, pp. 1–14.

[428] Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. “Key challenges for delivering clinical impact with artificial intelligence”. In: *BMC medicine* 17.1 (2019), pp. 1–9.

[429] Brendan Kennedy, X Jin, A M Davani, M Dehghani, and X Ren. “Contextualizing hate speech classifiers with post-hoc explanation”. In: *arXiv preprint arXiv:2005.02439* (2020).

[430] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of NAACL-HLT*. 2019, pp. 4171–4186.

[431] Daniel Kerrigan, Jessica Hullman, and Enrico Bertini. “A survey of domain knowledge elicitation in applied machine learning”. In: *Multimodal Technologies and Interaction* 5.12 (2021), p. 73.

[432] Os Keyes, Jevan Hutson, and Meredith Durbin. "A mulching proposal: Analysing and improving an algorithmic system for turning the elderly into high-nutrient slurry". In: *Extended abstracts of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–11.

[433] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. "A survey of the recent architectures of deep convolutional neural networks". In: *Artificial intelligence review* 53.8 (2020), pp. 5455–5516.

[434] Roli Khanna, J. Dodge, A. Anderson, R. Dikkala, Jed Irvine, Zeyad Shureih, Kin-ho Lam, Caleb R Matthews, Zhengxian Lin, Minsuk Kahng, et al. "Finding AI's faults with AAR/AI: An empirical study". In: *ACM TiiS* 12.1 (2022), pp. 1–33.

[435] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. "Undoing the damage of dataset bias". In: *European Conference on Computer Vision*. Springer. 2012, pp. 158–171.

[436] Sountongnoma Martial Anicet Kiemde and Ahmed Dooguy Kora. "Towards an ethics of AI in Africa: rule of education". In: *AI and Ethics* (2021), pp. 1–6.

[437] B Kim, M Wattenberg, and al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors". In: *ICML*. 2018.

[438] Been Kim, Oluwasanmi Koyejo, Rajiv Khanna, et al. "Examples are not enough, learn to criticize! Criticism for Interpretability." In: *NIPS*. 2016, pp. 2280–2288.

[439] Jae Yeon Kim, Carlos Ortiz, Sarah Nam, Sarah Santiago, and Vivek Datta. "Intersectional Bias in Hate Speech and Abusive Language Datasets". In: *arXiv preprint arXiv:2005.05921* (2020).

[440] Jieun Kim, Hokyoung Ryu, and Hyeonah Kim. "To Be Biased or Not to Be: Choosing Between Design Fixation and Design Intentionality". In: *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '13. Paris, France: ACM, 2013, pp. 349–354. ISBN: 978-1-4503-1952-2. DOI: 10.1145/2468356.2468418. URL: http://doi.acm.org/10.1145/2468356.2468418.

[441] Styliani Kleanthous, Maria Kasinidou, Pınar Barlas, and Jahna Otterbacher. "Perception of fairness in algorithmic decisions: Future developers' perspective". In: *Patterns* 3.1 (2022), p. 100380.

[442] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. "Inherent Trade-Offs in the Fair Determination of Risk Scores". In: *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Vol. 67. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 2017, p. 43.

[443] Kate Klonick. "The New Governors: The People, Rules and Processes Governing Online Speech". In: *Harvard Law Review* 131 (2018), p. 1598.

[444] F. Klubička and R. Fernández. "Examining a hate speech corpus for hate speech detection and popularity prediction". In: *Proceedings of 4REAL Workshop, Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language* (Miyazaki, Japan). 2018.

[445] Ari Kobren, Barna Saha, and Andrew McCallum. "Paper Matching with Local Fairness Constraints". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 2019, pp. 1247–1257. DOI: 10.1145/3292500.3330899. URL: https://doi.org/10.1145/3292500.3330899.

[446] Ansgar Koene, Liz Dowthwaite, and Suchana Seth. "IEEE P7003&Trade; Standard for Algorithmic Bias Considerations: Work in Progress Paper". In: *Proceedings of the International Workshop on Software Fairness*. FairWare '18. Gothenburg, Sweden: ACM, 2018, pp. 38–41. ISBN: 978-1-4503-5746-3. DOI: 10.1145/3194770.3194773. URL: http://doi.acm.org/10.1145/3194770.3194773.

[447] P W Koh and P Liang. "Understanding black-box predictions via influence functions". In: *arXiv preprint arXiv:1703.04730* (2017).

[448] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. "Wilds: A benchmark of in-the-wild distribution shifts". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5637–5664.

[449] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. "Analyzing and Reducing the Damage of Dataset Bias to Face Recognition With Synthetic Data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[450] Blagovesta Kostova, Seda Gürses, and Carmela Troncoso. "Privacy engineering meets software engineering. On the challenges of engineering privacy ByDesign". In: *arXiv preprint arXiv:2007.08613* (2020).

[451] Alicia Krebs, Alessandro Lenci, and Denis Paperno. "Semeval-2018 task 10: Capturing discriminative attributes". In: *12th Intl.workshop on semantic evaluation*. 2018, pp. 732–740.

[452] Dilip Krishna, Nancy Albinson, Yang Chu, and J Burdis. "Managing algorithmic risks—Safeguarding the use of complex algorithms and machine learning". In: *Deloitte Risk and Financial Advisory* (2017).

[453] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[454] Sean Kross and Philip Guo. "Orienting, framing, bridging, magic, and counseling: How data scientists navigate the outer loop of client collaborations in industry and academia". In: *ACM on Human-Computer Interaction* 5.CSCW2 (2021), pp. 1–28.

[455] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. "Principles of explanatory debugging to personalize interactive machine learning". In: *Proceedings of the 20th international conference on intelligent user interfaces*. 2015, pp. 126–137.

[456] Bogdan Kulynych, David Madras, Smitha Milli, Inioluwa Deborah Raji, Angela Zhou, and Richard Zemel. *Participatory Approaches to Machine Learning*. International Conference on Machine Learning Workshop. 2020.

[457] Bogdan Kulynych, Rebekah Overdorf, Carmela Troncoso, and Seda Gürses. "POTs: protective optimization technologies". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 177–188.

[458] Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. "Evaluating Aggression Identification in Social Media". In: *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. 2020, pp. 1–5.

[459] Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. "Aggression-annotated Corpus of Hindi-English Code-mixed Data". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. Ed. by Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. European Language Resources Association (ELRA), 2018. URL: http://www.lrec-conf.org/proceedings/lrec2018/summaries/861.html.

[460] MJ Kusner, J Loftus, Christopher Russell, and R Silva. "Counterfactual Fairness". In: *Advances in Neural Information Processing Systems 30 (NIPS 2017) pre-proceedings* 30 (2017).

[461] Irene Kwok and Yuzhou Wang. "Locate the hate: detecting tweets against blacks". In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press. 2013, pp. 1621–1622.

[462] Emanuele La Malfa and Marta Kwiatkowska. "The king is naked: on the notion of robustness for natural language processing". In: *AAAI*. Vol. 36. 10. 2022, pp. 11047–11057.

[463] Emanuele La Malfa, Min Wu, L. Laurenti, B. Wang, A. Hartshorn, and Marta Kwiatkowska. "Assessing Robustness of Text Classification through Maximal Safe Radius Computation". In: *EMNLP*. ACL, Nov. 2020, pp. 2949–2968. DOI: 10.18653/v1/2020.findings-emnlp.266. URL: https://aclanthology.org/2020.findings-emnlp.266.

[464] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. "Quantifying the carbon emissions of machine learning". In: *arXiv preprint arXiv:1910.09700* (2019).

[465] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. "iFair: Learning Individually Fair Data Representations for Algorithmic Decision Making". In: *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. 2019, pp. 1334–1345. DOI: 10.1109/ICDE.2019.00121. URL: https://doi.org/10.1109/ICDE.2019.00121.

[466] Jennifer L Lambe. "Who wants to censor pornography and hate speech?" In: *Mass Communication & Society* 7.3 (2004), pp. 279–299.

[467] Markus Langer, Tim Hunsicker, Tina Feldkamp, Cornelius J König, and Nina Grgić-Hlača. ""Look! it'sa computer program! it's an algorithm! it's ai!": does terminology affect human perceptions and evaluations of algorithmic decision-making systems?" In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–28.

[468] J. Langham and K. Gosha. "The Classification of Aggressive Dialogue in Social Media Platforms". In: *Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research*. ACM. 2018.

[469] Kyle Langvardt. "Regulating Online Content Moderation". In: *Georgetown Law Journal* 106.5 (2018), pp. 1353–1389.

[470] Issie Lapowsky. *Mark Zuckerberg and the Tale of Two Hearings*. https://www.wired.com/story/mark-zuckerberg-congress-day-two/, Last accessed on 2020-03-16. 2018.

[471] Theodoros Lappas and Evimaria Terzi. "Toward a Fair Review-Management System". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II*. 2011, pp. 293–309. DOI: 10.1007/978-3-642-23783-6_19. URL: https://doi.org/10.1007/978-3-642-23783-6%5C_19.

[472] Alfred Laugros, Alice Caplier, and Matthieu Ospici. *Addressing Neural Network Robustness with Mixup and Targeted Labeling Adversarial Training*. 2020. DOI: 10.48550/ARXIV.2008.08384. URL: https://arxiv.org/abs/2008.08384.

[473] Hady W Lauw, Ee-Peng Lim, and Ke Wang. "Bias and controversy in evaluation systems". In: *IEEE Transactions on Knowledge and Data Engineering* 20.11 (2008), pp. 1490–1504.

[474] Niklas Lavesson. "Learning machine learning: a case study". In: *Trans. on Education* 53.4 (2010), pp. 672–676.

[475] Lucas Layman, Madeline Diep, Meiyappan Nagappan, Janice Singer, Robert Deline, and Gina Venolia. "Debugging revisited: Toward understanding the debugging needs of contemporary software developers". In: *2013 ACM/IEEE international symposium on empirical software engineering and measurement*. IEEE. 2013, pp. 383–392.

[476] Susan Leavy. "Gender Bias in Artificial Intelligence: The Need for Diversity and Gender Theory in Machine Learning". In: *2018 IEEE/ACM 1st International Workshop on Gender Equality in Software Engineering, GE@ICSE, Gothenburg, Sweden, May 28, 2018*. 2018, pp. 14–16. URL: http://ieeexplore.ieee.org/document/8452744.

[477] Derek Leben. "Normative principles for evaluating fairness in machine learning". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 86–92.

[478] H-S. Lee, H-R. Lee, J-U. Park, and Y-S. Han. "An abusive text detection system based on enhanced abusive and non-abusive word lists". In: *Decision Support Systems* 113 (2018), pp. 22–31.

[479] Jihyun Lee, Sungwon Kang, and Danhyung Lee. "Survey on software testing practices". In: *IET software* 6.3 (2012), pp. 275–282.

[480] Michelle Seng Ah Lee, Luciano Floridi, and Jatinder Singh. "Formalising trade-offs beyond algorithmic fairness: lessons from ethical philosophy and welfare economics". In: *AI and Ethics* 1.4 (2021), pp. 529–544.

[481] Michelle Seng Ah Lee and Jat Singh. "The landscape and gaps in open source fairness toolkits". In: *Proceedings of the 2021 CHI conference on human factors in computing systems*. 2021, pp. 1–13.

[482] Min Kyung Lee and Su Baykal. "Algorithmic Mediation in Group Decisions: Fairness Perceptions of Algorithmically Mediated vs. Discussion-Based Social Division". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW '17. Portland, Oregon, USA: ACM, 2017, pp. 1035–1048. ISBN: 978-1-4503-4335-0. DOI: 10.1145/2998181.2998230. URL: http://doi.acm.org/10.1145/2998181.2998230.

[483] Min Kyung Lee, Ji Tae Kim, and Leah Lizarondo. "A Human-Centered Approach to Algorithmic Services: Considerations for Fair and Motivating Smart Community Service Management That Allocates Donations to Non-Profit Organizations". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 3365–3376. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025884. URL: http://doi.acm.org/10.1145/3025453.3025884.

[484] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. "WeBuildAI: Participatory framework for algorithmic governance". In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–35.

[485] Wonhee Lee, Samuel Sangkon Lee, Seungjong Chung, and Dongun An. "Harmful contents classification using the harmful word filtering and SVM". In: *International Conference on Computational Science*. Springer. 2007, pp. 18–25.

[486] Roselyn J Lee-Won, Tiffany N White, Hyunjin Song, Ji Young Lee, and Mikhail R Smith. "Source magnification of cyberhate: affective and cognitive effects of multiple-source hate messages on target group members". In: *Media Psychology* (2019), pp. 1–22.

[487] Roxanne Leitão and Filip Jakobsen. "A Survey on User-Interface Design Strategies to Address Online Bias". In: *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI EA '18. Montreal QC, Canada: ACM, 2018, LBW084:1–LBW084:6. ISBN: 978-1-4503-5621-3. DOI: 10.1145/3170427.3188567. URL: http://doi.acm.org/10.1145/3170427.3188567.

[488] Maurizio Leotta, Dario Olianas, and Filippo Ricca. "A large experimentation to analyze the effects of implementation bugs in machine learning algorithms". In: *Future Generation Computer Systems* 133 (2022), pp. 184–200.

[489] Piyawat Lertvittayakumjorn and Francesca Toni. "Explanation-Based Human Debugging of NLP Models: A Survey". In: *Framework* 3 (), p. 2.

[490] David Leslie. "Understanding Artificial Intelligence Ethics and Safety: A Guide for the Responsible Design and Implementation of AI Systems in the Public Sector". In: *Available at SSRN 3403301* (2019).

[491] Sam Levin. *Google to hire thousands of moderators after outcry over YouTube abuse videos*. https://www.theguardian.com/technology/2017/dec/04/google-youtube-hire-moderators-child-abuse-videos, Last accessed on 2020-03-16. 2017.

[492] Ariel Levy, Monica Agrawal, Arvind Satyanarayan, and David Sontag. "Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–13.

[493] Yi Li and Nuno Vasconcelos. "Repair: Removing representation bias by dataset resampling". In: *ICCV*. 2019, pp. 9572–9581.

[494] Yingwei Li, Yi Li, and Nuno Vasconcelos. "RESOUND: Towards action recognition without representation bias". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 513–528.

[495] Z. Li, J. Kawamoto, Y. Feng, and K. Sakurai. "Cyberbullying detection using parent-child relationship between comments". In: *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services - iiWAS '16*. New York, USA: ACM Press, 2016. ISBN: 9781450348072. DOI: 10.1145/3011141.3011182. URL: http://dl.acm.org/citation.cfm?doid=3011141.3011182.

[496] Konstantinos G Liakos, Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. "Machine learning in agriculture: A review". In: *Sensors* 18.8 (2018), p. 2674.

[497] Q Vera Liao, Daniel Gruen, and Sarah Miller. "Questioning the AI: informing design practices for explainable AI user experiences". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–15.

[498] Q Vera Liao, Milena Pribić, Jaesik Han, Sarah Miller, and Daby Sow. "Question-Driven Design Process for Explainable AI User Experiences". In: *arXiv preprint arXiv:2104.03483* (2021).

[499] Q. Vera Liao, Wai-Tat Fu, and Markus Strohmaier. "#Snowden: Understanding Biases Introduced by Behavioral Differences of Opinion Groups on Social Media". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 3352–3363. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858422. URL: http://doi.acm.org/10.1145/2858036.2858422.

[500] Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. “Are we learning yet? a meta review of evaluation failures across machine learning”. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[501] Horst Lichter, Matthias Schneider-Hufschmidt, and Heinz Zullighoven. “Prototyping in industrial software projects-bridging the gap between theory and practice”. In: *IEEE transactions on software engineering* 20.11 (1994), pp. 825–832.

[502] Henry Lieberman, Dustin Smith, and Alea Teeters. “Common Consensus: a web-based game for collecting commonsense goals”. In: *ACM Workshop on Common Sense for Intelligent Interfaces*. 2007.

[503] Brian Y Lim, Qian Yang, Ashraf M Abdul, and Danding Wang. “Why these Explanations? Selecting Intelligibility Types for Explanation Goals.” In: *IUI Workshops*. 2019.

[504] Phoebe Lin and Jessica Van Brummelen. “Engaging Teachers to Co-Design Integrated AI Curriculum for K-12 Classrooms”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–12.

[505] Yin Lin, Yifan Guan, Abolfazl Asudeh, and HV Jagadish. “Identifying insufficient data coverage in databases with multiple relations”. In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 2229–2242.

[506] Anthony Liu, Santiago Guerra, Isaac Fung, Gabriel Matute, Ece Kamar, and Walter Lasecki. “Towards hybrid human-ai workflows for unknown unknown detection”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 2432–2442.

[507] Hugo Liu and Push Singh. “ConceptNet—a practical commonsense reasoning tool-kit”. In: *BT technology journal* 22.4 (2004), pp. 211–226.

[508] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. “Delayed impact of fair machine learning”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3150–3158.

[509] S. Liu and T. Forss. “Text Classification Models for Web Content Filtering and Online Safety”. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, Nov. 2015, pp. 961–968. ISBN: 978-1-4673-8493-3. DOI: 10.1109/ICDMW.2015.143. URL: http://ieeexplore.ieee.org/document/7395771/.

[510] X Liu, H Zhao, M Tian, L Sheng, J Shao, S Yi, J Yan, and al. “Hydraplus-net: Attentive deep features for pedestrian analysis”. In: *Proc. of IEEE ICCV*. 2017, pp. 350–359.

[511] Xiaoxuan Liu, Ben Glocker, Melissa M McCradden, Marzyeh Ghassemi, Alastair K Denniston, and Lauren Oakden-Rayner. “The medical algorithmic audit”. In: *The Lancet Digital Health* (2022).

[512] Duri Long and Brian Magerko. “What is AI literacy? Competencies and design considerations”. In: *2020 CHI*. 2020, pp. 1–16.

[513] Gabi Löschper, Amélie Mummendey, Volker Linneweber, and Manfred Bornewasser. “The judgement of behaviour as aggressive and sanctionable”. In: *European Journal of Social Psychology* 14.4 (1984), pp. 391–404.

[514] Raoni Lourenço, Juliana Freire, and Dennis Shasha. “Debugging machine learning pipelines”. In: *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning*. 2019, pp. 1–10.

[515] Alan Lundgard. “Measuring justice in machine learning”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 680–680.

[516] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. “k-NN as an implementation of situation testing for discrimination discovery and prevention”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 502–510.

[517] Henrietta Lyons, Eduardo Velloso, and Tim Miller. “Conceptualising contestability: Perspectives on contesting algorithmic decisions”. In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–25.

[518] Estevez Almenzar M, Fernandez Llorca D, Gomez Gutierrez E, and Martinez Plumed F. *Glossary of human-centric artificial intelligence*. Scientific analysis or review, Technical guidance KJ-NA-31113-EN-N (online). Luxembourg (Luxembourg), 2022. DOI: 10.2760/860665(online).

[519] Shiqing Ma, Yingqi Liu, Wen-Chuan Lee, Xiangyu Zhang, and Ananth Grama. "MODE: automated neural network model debugging via state differential analysis and input selection". In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2018, pp. 175–186.

[520] Wendy E Mackay and Anne-Laure Fayard. "HCI, natural science and design: a framework for triangulation across disciplines". In: *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*. 1997, pp. 223–234.

[521] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. "Assessing the Fairness of AI Systems: AI Practitioners' Processes, Challenges, and Needs for Support". In: *ACM on Human-Computer Interaction* 6.CSCW1 (2022), pp. 1–26.

[522] Michael Madaio, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. "Assessing the Fairness of AI Systems: AI Practitioners' Processes, Challenges, and Needs for Support". In: *Proc. ACM Hum.-Comput. Interact.* 6.CSCW1 (Apr. 2022). DOI: 10.1145/3512899. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3512899.

[523] Michael A Madaio, Luke Stark, Jennifer Wortman Vaughan, and Hanna Wallach. "Co-designing checklists to understand organizational challenges and opportunities around fairness in AI". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14.

[524] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. "Fairness through causal awareness: Learning causal latent-variable models for biased data". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. 2019, pp. 349–358.

[525] Rijul Magu and Jiebo Luo. "Determining code words in euphemistic hate speech using word embedding networks". In: *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. 2018.

[526] Subha Maity, Debarghya Mukherjee, Mikhail Yurochkin, and Yuekai Sun. "There is no trade-off: enforcing fairness can improve accuracy". In: *stat* 1050 (2020), p. 6.

[527] Taro Makino, Stanisław Jastrzębski, Witold Oleszkiewicz, Celin Chacko, Robin Ehrenpreis, Naziya Samreen, Chloe Chhor, Eric Kim, Jiyon Lee, Kristine Pysarenko, et al. "Differences between human and machine perception in medical diagnosis". In: *Scientific reports* 12.1 (2022), pp. 1–13.

[528] V. Mal and A. J. Agrawal. "Removing Flaming Problems from Social Networking Sites Using Semi-Supervised Learning Approach". In: *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. ICTCS '16. Udaipur, India: ACM, 2016. ISBN: 978-1-4503-3962-9. DOI: 10.1145/2905055.2905338. URL: http://doi.acm.org/10.1145/2905055.2905338.

[529] Nicolas Maleve. "An Introduction to Image Datasets". In: (2019). URL: https://unthinking.photography/articles/an-introduction-to-image-datasets.

[530] Shervin Malmasi and Marcos Zampieri. "Challenges in discriminating profanity from hate speech". In: *Journal of Experimental & Theoretical Artificial Intelligence* 30.2 (2018), pp. 187–202.

[531] A. Mangaonkar, A. Hayrapetian, and R. Raje. "Collaborative detection of cyberbullying behavior in Twitter data". In: *2015 IEEE International Conference on Electro/Information Technology (EIT)*. May 2015. ISBN: 978-1-4799-8802-0. DOI: 10.1109/EIT.2015.7293405. URL: http://ieeexplore.ieee.org/document/7293405/.

[532] Gary Marcus. "The next decade in ai: four steps towards robust artificial intelligence". In: *arXiv preprint arXiv:2002.06177* (2020).

[533] Panos Ipeirotis Margarita Boyarskaya. "Fair Payments in Adaptive Voting". In: *Proceedings of HCOMP 2019*. 2019.

[534] Eric Margolis and Stephen Laurence. "The ontology of concepts-abstract objects or mental representations?" In: *Noûs* 41.4 (2007), pp. 561–593.

[535] M Lynne Markus, Marco Marabelli, and Christina Zhu. "POETs and quants: Ethics education for data scientists and managers". In: *Marco and Zhu, Xiaolin (Christina), POETs and Quants: Ethics Education for Data Scientists and Managers (November 19, 2019)* (2019).

[536] Afra Mashhadi, Annuska Zolyomi, and Jay Quedado. “A Case Study of Integrating Fairness Visualization Tools in Machine Learning Education”. In: *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 2022, pp. 1–7.

[537] Ariadna Matamoros-Fernández and Johan Farkas. “Racism, Hate Speech, and Social Media: A Systematic Review and Critique”. In: *Television & New Media* 22.2 (2021), pp. 205–224.

[538] J Nathan Matias. “Preventing harassment and increasing group participation through social norms in 2,190 online science discussions”. In: *Proceedings of the National Academy of Sciences* 116.20 (2019), pp. 9785–9789.

[539] Yasuyuki Matsushita, Stephen Lin, Sing Bing Kang, and Heung-Yeung Shum. “Estimating intrinsic images from image sequences with biased illumination”. In: *European Conference on Computer Vision*. Springer. 2004, pp. 274–286.

[540] Peter Mattson, Christine Cheng, Gregory Diamos, Cody Coleman, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bittorf, et al. “Mlperf training benchmark”. In: *Proceedings of Machine Learning and Systems* 2 (2020), pp. 336–349.

[541] Anneliese von Mayrhauser and A Marie Vans. “Program understanding behavior during debugging of large scale software”. In: *Papers presented at the seventh workshop on Empirical studies of programmers*. 1997, pp. 157–179.

[542] Renee McCauley, Sue Fitzgerald, Gary Lewandowski, Laurie Murphy, Beth Simon, Lynda Thomas, and Carol Zander. “Debugging: a review of the literature from an educational perspective”. In: *Computer Science Education* 18.2 (2008), pp. 67–92.

[543] Melissa D McCradden, Shalmali Joshi, Mjaye Mazwi, and James A Anderson. “Ethical limitations of algorithmic fairness solutions in health care machine learning”. In: *The Lancet Digital Health* 2.5 (2020), e221–e223.

[544] Nora McDonald and Shimei Pan. “Intersectional AI: A Study of How Information Science Students Think about Ethics and Their Impact”. In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW2 (2020), pp. 1–19.

[545] M. L. McHugh. “Interrater reliability: the kappa statistic”. In: *Biochemia medica: Biochemia medica* 22.3 (2012), pp. 276–282.

[546] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. “A survey on bias and fairness in machine learning”. In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–35.

[547] Jacob Metcalf and Kate Crawford. “Where are human subjects in big data research? The emerging ethics divide”. In: *Big Data & Society* 3.1 (2016), p. 2053951716650211.

[548] Milagros Miceli, Tianling Yang, Laurens Naudts, Martin Schuessler, Diana Serbanescu, and Alex Hanna. “Documenting computer vision datasets: an invitation to reflexive data practices”. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 161–172.

[549] Tilman Michaeli and Ralf Romeike. “Improving debugging skills in the classroom: The effects of teaching a systematic debugging process”. In: *14th workshop in primary and secondary computing education*. 2019, pp. 1–7.

[550] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[551] Brad Miller, Alex Kantchelian, Sadia Afroz, Rekha Bachwani, E. Dauber, L. Huang, M. C. Tschantz, A. D. Joseph, and J Doug Tygar. “Adversarial active learning”. In: *Workshop on Artificial Intelligent and Security*. 2014, pp. 3–14.

[552] David J Miller, Xinyi Hu, Zhicong Qiu, and George Kesidis. “Adversarial learning: a critical review and active learning study”. In: *Intl. Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2017, pp. 1–6.

[553] Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. “The social cost of strategic classification”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 2019, pp. 230–239.

[554] A. Mishra and R. Rastogi. "Semi-supervised correction of biased comment ratings". In: *Proceedings of the 21st international conference on World Wide Web - WWW '12*. New York, USA: ACM Press, 2012. ISBN: 9781450312295. DOI: 10.1145/2187836.2187862. URL: http://dl.acm.org/citation.cfm?doid=2187836.2187862.

[555] Swati Mishra and Jeffrey M Rzeszotarski. "Crowdsourcing and Evaluating Concept-driven Explanations of Machine Learning Models". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–26.

[556] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. "Model cards for model reporting". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 220–229.

[557] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. "Algorithmic fairness: Choices, assumptions, and definitions". In: *Annual Review of Statistics and Its Application* 8 (2021), pp. 141–163.

[558] Petra Molnar. "Technological Testing Grounds and Surveillance Sandboxes: Migration and Border Technology at the Frontiers". In: *Fletcher F. World Aff.* 45 (2021), p. 109.

[559] M. Mondal, L. A. Silva, and F. Benevenuto. "A Measurement Study of Hate Speech in Social Media". In: *Proceedings of the 28th ACM Conference on Hypertext and Social Media - HT '17*. New York, USA: ACM Press, 2017, pp. 85–94. ISBN: 9781450347082. DOI: 10.1145/3078714.3078723. URL: http://dl.acm.org/citation.cfm?doid=3078714.3078723.

[560] Vijay S. Mookerjee. "Debiasing training data for inductive expert system construction". In: *IEEE Transactions on Knowledge and Data Engineering* 13.3 (2001), pp. 497–512.

[561] Phoebe V Moore and Jamie Woodcock. *Augmented exploitation: Artificial intelligence, automation, and work*. Pluto Press, 2021.

[562] Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. *Divide, Denoise, and Defend against Adversarial Attacks*. 2018. DOI: 10.48550/ARXIV.1802.06806. URL: https://arxiv.org/abs/1802.06806.

[563] Jonathan Scott Morgan, Cliff Lampe, and Muhammad Zubair Shafiq. "Is News Sharing on Twitter Ideologically Biased?" In: *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. CSCW '13. San Antonio, Texas, USA: ACM, 2013, pp. 887–896. ISBN: 978-1-4503-1331-5. DOI: 10.1145/2441776.2441877. URL: http://doi.acm.org/10.1145/2441776.2441877.

[564] Jessica Morley, Libby Kinsey, Anat Elhalal, Francesca Garcia, Marta Ziosi, and Luciano Floridi. "Operationalising AI ethics: barriers, enablers and next steps". In: *AI & SOCIETY* (2021), pp. 1–13.

[565] Yuval Moskovitch and HV Jagadish. "COUNTATA: dataset labeling using pattern counts". In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 2829–2832.

[566] Hussein Mouzannar, Mesrob I Ohannessian, and Nathan Srebro. "From fair decision making to social equality". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. 2019, pp. 359–368.

[567] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. "Hate speech detection and racial bias mitigation in social media based on BERT model". In: *PloS one* 15.8 (2020), e0237861.

[568] Norman Mu and Justin Gilmer. "Mnist-c: A robustness benchmark for computer vision". In: *arXiv preprint arXiv:1906.02337* (2019).

[569] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q Vera Liao, Casey Dugan, and Thomas Erickson. "How data science workers work with data: Discovery, capture, curation, design, creation". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–15.

[570] Michael Muller and Angelika Strohmayer. "Forgetting Practices in the Data Sciences". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517644. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3491102.3517644.

[571] Deirdre K Mulligan, Joshua A Kroll, Nitin Kohli, and Richmond Y Wong. "This thing called fairness: Disciplinary confusion realizing a value in technology". In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–36.

[572] Amelie Mummendey and Michael Wenzel. "Social discrimination and tolerance in intergroup relations: Reactions to intergroup difference". In: *Personality and Social Psychology Review* 3.2 (1999), pp. 158–174.

[573] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences* 116.44 (2019), pp. 22071–22080.

[574] Vedant Nanda, Till Speicher, John P. Dickerson, Krishna P. Gummadi, and Muhammad Bilal Zafar. *Unifying Model Explainability and Robustness via Machine-Checkable Concepts*. 2020. DOI: 10.48550/ARXIV.2007.00251. URL: https://arxiv.org/abs/2007.00251.

[575] B. S. Nandhini and J. I. Sheeba. "Cyberbullying Detection and Classification Using Information Retrieval Algorithm". In: *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*. New York, USA: ACM Press, 2015, pp. 1–5. ISBN: 9781450334419. DOI: 10.1145/2743065.2743085. URL: http://dl.acm.org/citation.cfm?doid=2743065.2743085%20http://dx.doi.org/10.1145/2743065.2743085.

[576] Luca Nannini, Agathe Balayn, and Adam Leon Smith. "Explainability in AI Policies: A Critical Review of Communications, Reports, Regulations, and Standards in the EU, US, and UK". In: *2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2023.

[577] UP Narendra, BS Pradeep, and M Prabhakar. "Externalization of tacit knowledge in a knowledge management system using chat bots". In: *2017 3rd Intl. Conf. on Science in Information Technology (ICSITech)*. IEEE. 2017, pp. 613–617.

[578] Shweta Narkar, Yunfeng Zhang, Q Vera Liao, Dakuo Wang, and Justin D Weisz. "Model LineUpper: Supporting Interactive Model Comparison at Multiple Levels for AutoML". In: *26th International Conference on Intelligent User Interfaces*. 2021, pp. 170–174.

[579] Vishwajeet Narwal, Mohamed Hashim Salih, Jose Angel Lopez, Angel Ortega, John O'Donovan, Tobias Höllerer, and Saiph Savage. "Automated Assistants to Identify and Prompt Action on Visual News Bias". In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: ACM, 2017, pp. 2796–2801. ISBN: 978-1-4503-4656-6. DOI: 10.1145/3027063.3053227. URL: http://doi.acm.org/10.1145/3027063.3053227.

[580] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. *Exploring Generalization in Deep Learning*. 2017. DOI: 10.48550/ARXIV.1706.08947. URL: https://arxiv.org/abs/1706.08947.

[581] Kun-Peng Ning, Lue Tao, Songcan Chen, and Sheng-Jun Huang. "Improving Model Robustness by Adaptively Correcting Perturbation Levels with Active Queries". In: *EAAI*. AAAI Press, 2021, pp. 9161–9169. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17106.

[582] T. Nitta, F. Masui, M. Ptaszynski, Y. Kimura, Rzepka R., and K. Araki. *Detecting Cyberbullying Entries on Informal School Websites Based on Category Relevance Maximization*. Tech. rep. 2013. URL: http://mecab.sourceforge.net/.

[583] Ardavan Salehi Nobandegani, Kevin da Silva Castanheira, Timothy O'Donnell, and Thomas R Shultz. "On Robustness: An Undervalued Dimension of Human Rationality." In: *CogSci*. 2019, p. 3327.

[584] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. "Abusive Language Detection in Online User Content". In: *Proceedings of the 25th International Conference on World Wide Web*. New York, USA: ACM Press, 2016. ISBN: 9781450341431. DOI: 10.1145/2872427.2883062. URL: http://dx.doi.org/10.1145/2872427.2883062%20http://dl.acm.org/citation.cfm?doid=2872427.2883062.

[585] Ikujirō Nonaka and Hirotaka Takeuchi. "The knowledge-creating company". In: *Harvard business review* 85.7/8 (2007), p. 162.

[586] Donald A Norman. "The research-Practice Gap: The need for translational developers". In: *interactions* 17.4 (2010), pp. 9–12.

[587] Mahsan Nourani, Samia Kabir, Sina Mohseni, and Eric D Ragan. "The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 2019, pp. 97–105.

[588] Noviantho, S. M. Isa, and L. Ashianti. "Cyberbullying classification using text mining". In: *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, Nov. 2017, pp. 241–246. ISBN: 978-1-5386-0903-3. DOI: 10.1109/ICICOS.2017.8276369. URL: http://ieeexplore.ieee.org/document/8276369/.

[589] Ika Nurfarida and Laudetta Dianne Fitri. "Mapping and Defining Hate Speech in Instagram's Comments: A Study of Language Use in Social Media". In: *SENABASTRA*| *8* (), p. 105.

[590] Besmira Nushi, Ece Kamar, Eric Horvitz, and Donald Kossmann. "On human intellect and machine failures: Troubleshooting integrative machine learning systems". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[591] C. J. O'Dea, S. S. Miller, E. B. Andres, M. H. Ray, D. F. Till, and D. A. Saucier. "Out of bounds: factors affecting the perceived offensiveness of racial slurs". In: *Language Sciences* 52 (2015), pp. 155–164.

[592] Heather L O'Brien, Paul Cairns, and Mark Hall. "A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form". In: *Intl.Journal of Human-Computer Studies* 112 (2018), pp. 28–39.

[593] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464 (2019), pp. 447–453.

[594] Damla Oguz and Kaya Oguz. "Perspectives on the gap between the software industry and the software engineering education". In: *IEEE Access* 7 (2019), pp. 117527–117543.

[595] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. "Feature visualization". In: *Distill* 2.11 (2017), e7.

[596] Gary M Olson, Sylvia Sheppard, and Elliot Soloway. *Empirical studies of programmers: second workshop*. Vol. 2. Intellect Books, 1987.

[597] A. Olteanu, K. Talamadupula, and K. R. Varshney. "The Limits of Abstract Evaluation Metrics: The Case of Hate Speech Detection". In: *Proceedings of the 2017 ACM on Web Science Conference*. WebSci '17. Troy, New York, USA: ACM, 2017, pp. 405–406. ISBN: 978-1-4503-4896-6. DOI: 10.1145/3091478.3098871. URL: http://doi.acm.org/10.1145/3091478.3098871.

[598] Laurel Orr, Samuel Ainsworth, Walter Cai, Kevin Jamieson, Magda Balazinska, and Dan Suciu. "Mosaic: A Sample-Based Database System for Open World Query Processing". In: *CIDR*. 2020.

[599] Laurel Orr, Magdalena Balazinska, and Dan Suciu. "Sample debiasing in the themis open world database system". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020, pp. 257–268.

[600] Jahna Otterbacher. "Crowdsourcing Stereotypes: Linguistic Bias in Metadata Generated via GWAP". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 1955–1964. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702151. URL: http://doi.acm.org/10.1145/2702123.2702151.

[601] Jahna Otterbacher. "Social cues, social biases: stereotypes in annotations on people images". In: *Sixth AAAI Conference on Human Computation and Crowdsourcing*. 2018.

[602] Jahna Otterbacher, Pinar Barlas, Styliani Kleanthous, and Kyriakos Kyriakou. "How Do We Talk About Other People? Group (Un)Fairness in Natural Language Image Descriptions". In: *HCOMP 2019*. 2019.

[603] Rebekah Overdorf, Bogdan Kulynych, Ero Balsa, Carmela Troncoso, and Seda Gürses. "Questioning the assumptions behind fairness solutions". In: *arXiv preprint arXiv:1811.11293* (2018).

[604] S. Ozawa, S. Yoshida, J. Kitazono, T. Sugawara, and T. Haga. "A sentiment polarity prediction model using transfer learning and its application to SNS flaming event detection". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Dec. 2016, pp. 1–7. ISBN: 978-1-5090-4240-1. DOI: 10.1109/SSCI.2016.7849868. URL: http://ieeexplore.ieee.org/document/7849868/.

[605] S. A. Ozel, E. Sarac, S. Akdemir, and H. Aksu. "Detection of cyberbullying on social media messages in Turkish". In: *2017 International Conference on Computer Science and Engineering*. IEEE, Oct. 2017. ISBN: 978-1-5386-0930-9. DOI: 10.1109/UBMK.2017.8093411. URL: http://ieeexplore.ieee.org/document/8093411/.

[606] Stefan Palan and Christian Schitter. "Prolific. ac—A subject pool for online experiments". In: *Journal of Behavioral and Experimental Finance* 17 (2018), pp. 22–27.

[607] Rameswar Panda, Jianming Zhang, Haoxiang Li, Joon-Young Lee, Xin Lu, and Amit K Roy-Chowdhury. "Contemplating Visual Emotions: Understanding and Overcoming Dataset Bias". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 579–595.

[608] Tianyu Pang, Huishuai Zhang, Di He, Yinpeng Dong, Hang Su, Wei Chen, Jun Zhu, and Tie-Yan Liu. "Two Coupled Rejection Metrics Can Tell Adversarial Examples Apart". In: *CVPR*. 2022, pp. 15223–15233.

[609] Andrea Papenmeier, Dagmar Kern, Daniel Hienert, Yvonne Kammerer, and Christin Seifert. "How Accurate Does It Feel?–Human Perception of Different Types of Classification Mistakes". In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–13.

[610] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. "Towards the science of security and privacy in machine learning". In: *arXiv preprint arXiv:1611.03814* (2016).

[611] Praveen Paritosh, Panos Ipeirotis, Matt Cooper, and Siddharth Suri. "The computer is the new sewing machine: benefits and perils of crowdsourcing". In: *Proceedings of the 20th international conference companion on World wide web*. 2011, pp. 325–326.

[612] Hyanghee Park, Daehwan Ahn, Kartik Hosanagar, and Joonhwan Lee. "Designing Fair AI in Human Resource Management: Understanding Tensions Surrounding Algorithmic Evaluation and Envisioning Stakeholder-Centered Solutions". In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–22.

[613] J. H. Park, J. Shin, and P. Fung. "Reducing Gender Bias in Abusive Language Detection". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 2799–2804.

[614] Ji Ho Park and Pascale Fung. "One-step and Two-step Classification for Abusive Language Detection on Twitter". In: *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*. Ed. by Zeerak Waseem, Wendy Hui Kyong Chung, Dirk Hovy, and Joel R. Tetreault. Association for Computational Linguistics, 2017, pp. 41–45. ISBN: 978-1-945626-66-1. DOI: 10.18653/v1/w17-3006. URL: https://doi.org/10.18653/v1/w17-3006.

[615] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. "A slow algorithm improves users' assessments of the algorithm's accuracy". In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–15.

[616] Souneil Park, Seungwoo Kang, Sangyoung Chung, and Junehwa Song. "NewsCube: Delivering Multiple Aspects of News to Mitigate Media Bias". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 443–452. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518772. URL: http://doi.acm.org/10.1145/1518701.1518772.

[617] Magdalini Paschali, Sailesh Conjeti, Fernando Navarro, and Nassir Navab. *Generalizability vs. Robustness: Adversarial Examples for Medical Imaging*. 2018. DOI: 10.48550/ARXIV.1804.00504. URL: https://arxiv.org/abs/1804.00504.

[618] Demetris Paschalides, Dimosthenis Stephanidis, Andreas Andreou, Kalia Orphanou, George Pallis, Marios D Dikaiakos, and Evangelos Markatos. "MANDOLA: A big-data processing and visualization platform for monitoring and detecting online hate speech". In: *ACM Transactions on Internet Technology (TOIT)* 20.2 (2020), pp. 1–21.

[619] Samir Passi and Solon Barocas. "Problem formulation and fairness". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. 2019, pp. 39–48.

[620] Samir Passi and Steven J Jackson. "Trust in data science: Collaboration, translation, and accountability in corporate data science projects". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), pp. 1–28.

[621] Samir Passi and Phoebe Sengers. “Making data science systems work”. In: *Big Data & Society* 7.2 (2020), p. 2053951720939605. DOI: 10.1177/2053951720939605. eprint: https://doi.org/10.1177/2053951720939605. URL: https://doi.org/10.1177/2053951720939605.

[622] Kayur Patel, Naomi Bancroft, Steven M Drucker, James Fogarty, Amy J Ko, and James Landay. “Gestalt: integrated support for implementation and analysis in machine learning”. In: *23nd annual ACM symposium on User interface software and technology*. 2010, pp. 37–46.

[623] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. “Data and its (dis) contents: A survey of dataset development and use in machine learning research”. In: *arXiv preprint arXiv:2012.05345* (2020).

[624] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. “Deeper Attention to Abusive User Content Moderation”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 1125–1135.

[625] John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. “Improved Abusive Comment Moderation with User Embeddings”. In: *Proceedings of the 2017 Workshop: Natural Language Processing meets Journalism, NLPmJ@EMNLP, Copenhagen, Denmark, September 7, 2017*. Ed. by Octavian Popescu and Carlo Strapparava. Association for Computational Linguistics, 2017, pp. 51–55. ISBN: 978-1-945626-88-3. DOI: 10.18653/v1/w17-4209. URL: https://doi.org/10.18653/v1/w17-4209.

[626] John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. “Toxicity Detection: Does Context Really Matter?” In: *arXiv preprint arXiv:2006.00998* (2020).

[627] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. “Discrimination-aware data mining”. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2008, pp. 560–568.

[628] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. “Towards practical verification of machine learning: The case of computer vision systems”. In: *arXiv preprint arXiv:1712.01785* (2017).

[629] Seeta Peña Gangadharan and Jędrzej Niklas. “Decentering technology in discourse on discrimination”. In: *Information, Communication & Society* 22.7 (2019), pp. 882–899.

[630] Andi Peng, Besmira Nushi, Emre Kıcıman, Kori Inkpen, Siddharth Suri, and Ece Kamar. “What you see is what you get? the impact of representation criteria on human bias in hiring”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 2019, pp. 125–134.

[631] Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-Garcia, Jordi Munoz-Mari, Luis Gómez-Chova, and Gustau Camps-Valls. “Fair Kernel Learning”. In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part I*. 2017, pp. 339–355. DOI: 10.1007/978-3-319-71249-9_21. URL: https://doi.org/10.1007/978-3-319-71249-9%5C_21.

[632] Billy Perrigo. *Facebook Says It’s Removing More Hate Speech Than Ever Before. But There’s a Catch*. https://time.com/5739688/facebook-hate-speech-languages/, Last accessed on 2020-03-16. 2019.

[633] Michael Perscheid, Benjamin Siegmund, Marcel Taeumel, and Robert Hirschfeld. “Studying the advancement in debugging practice of professional software developers”. In: *Software Quality Journal* 25.1 (2017), pp. 83–110.

[634] Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. *Human uncertainty makes classification more robust*. 2019. DOI: 10.48550/ARXIV.1908.07086. URL: https://arxiv.org/abs/1908.07086.

[635] Jean S Phinney, Tanya Madden, and Lorena J Santos. “Psychological variables as predictors of perceived ethnic discrimination among minority and immigrant adolescents 1”. In: *Journal of Applied Social Psychology* 28.11 (1998), pp. 937–953.

[636] Emma Pierson. “Demographics and discussion influence views on algorithmic fairness”. In: *arXiv preprint arXiv:1712.09124* (2017).

[637] David Piorkowski, Soya Park, April Yi Wang, Dakuo Wang, Michael Muller, and Felix Portnoy. "How ai developers overcome communication challenges in a multidisciplinary team: A case study". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–25.

[638] G. K. Pitsilis, H. Ramampiaro, and H. Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks". In: *Applied Intelligence* (July 2018). ISSN: 0924-669X. DOI: 10.1007/s10489-018-1242-y. URL: http://link.springer.com/10.1007/s10489-018-1242-y.

[639] Karyn M Plumm, Cheryl A Terrance, and Adam Austin. "Not all hate crimes are created equal: An examination of the roles of ambiguity and expectations in perceptions of hate crimes". In: *Current Psychology* 33.3 (2014), pp. 321–364.

[640] Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. "Resources and benchmark corpora for hate speech detection: a systematic review". In: *Language Resources and Evaluation* (2020), pp. 1–47.

[641] Fabio Poletto, Marco Stranisci, Manuela Sanguinetti, Viviana Patti, and Cristina Bosco. "Hate speech annotation: Analysis of an italian twitter corpus". In: *4th Italian Conference on Computational Linguistics, CLiC-it 2017*. Vol. 2006. CEUR-WS. 2017, pp. 1–6.

[642] N. Potha and M. Maragoudakis. "Cyberbullying Detection using Time Series Modeling". In: *2014 IEEE International Conference on Data Mining Workshop*. IEEE, Dec. 2014, pp. 373–382. ISBN: 978-1-4799-4274-9. DOI: 10.1109/ICDMW.2014.170. URL: http://ieeexplore.ieee.org/document/7022621/.

[643] Daniel J Power. *Decision support systems: concepts and resources for managers*. Greenwood Publishing Group, 2002.

[644] Duncan Pritchard. *What is this thing called knowledge?* Routledge, 2013.

[645] Michal Ptaszynski, Pawel Dybala, Tatsuaki Matsuba, Fumito Masui, Rafal Rzepka, and Kenji Araki. "Machine learning and affect analysis against cyber-bullying". In: *the 36th AISB* (2010), pp. 7–16.

[646] David Pujol, Ryan McKenna, Satya Kuppam, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. "Fair decision making using privacy-protected data". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 189–199.

[647] Jing Qian, Mai ElSherief, Elizabeth M. Belding, and William Yang Wang. "Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, 2018, pp. 118–123. ISBN: 978-1-948087-29-2. DOI: 10.18653/v1/n18-2019. URL: https://doi.org/10.18653/v1/n18-2019.

[648] Shangshu Qian, Viet Hung Pham, Thibaud Lutellier, Zeou Hu, Jungwon Kim, Lin Tan, Yaoliang Yu, Jiahao Chen, and Sameena Shah. "Are my deep learning systems fair? An empirical study of fixed-seed training". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 30211–30227.

[649] Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. "Discovering fair representations in the data domain". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8227–8236.

[650] Giovanni Quattrone, Licia Capra, and Pasquale De Meo. "There's No Such Thing As the Perfect Map: Quantifying Bias in Spatial Crowd-sourcing Datasets". In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW '15. Vancouver, BC, Canada: ACM, 2015, pp. 1021–1032. ISBN: 978-1-4503-2922-4. DOI: 10.1145/2675133.2675235. URL: http://doi.acm.org/10.1145/2675133.2675235.

[651] R. I. Rafiq, H. Hosseinmardi, R. Han, Q. Lv, and S. Mishra. "Scalable and timely detection of cyberbullying in online social networks". In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18*. New York, USA: ACM Press, 2018, pp. 1738–1747. ISBN: 9781450351911. DOI: 10.1145/3167132.3167317. URL: http://dl.acm.org/citation.cfm?doid=3167132.3167317.

[652] R. I. Rafiq, H. Hosseinmardi, R. Han, Q. Lv, S. Mishra, and S. A. Mattson. "Careful what you share in six seconds: Detecting cyberbullying instances in Vine". In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*. New York, USA: ACM Press, 2015, pp. 617–622. ISBN: 9781450338547. DOI: 10.1145/2808797.2809381. URL: http://dl.acm.org/citation.cfm?doid=2808797.2809381.

[653] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. "Mitigating bias in algorithmic hiring: Evaluating claims and practices". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 469–481.

[654] Foyzur Rahman, Daryl Posnett, Israel Herraiz, and Premkumar Devanbu. "Sample Size vs. Bias in Defect Prediction". In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. ESEC/FSE 2013. Saint Petersburg, Russia: ACM, 2013, pp. 147–157. ISBN: 978-1-4503-2237-9. DOI: 10.1145/2491411.2491418. URL: http://doi.acm.org/10.1145/2491411.2491418.

[655] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, Serkan Kiranyaz, Saad Bin Abul Kashem, Mohammad Tariqul Islam, Somaya Al Maadeed, Susu M Zughaier, Muhammad Salman Khan, et al. "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images". In: *Computers in biology and medicine* 132 (2021), p. 104319.

[656] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. "Machine behaviour". In: *Nature* 568.7753 (2019), pp. 477–486.

[657] E. Raisi and B. Huang. "Weakly supervised cyberbullying detection with participant-vocabulary consistency". In: *Social Network Analysis and Mining* 8.1 (Dec. 2018), p. 38. ISSN: 1869-5450. DOI: 10.1007/s13278-018-0517-y. URL: http://link.springer.com/10.1007/s13278-018-0517-y.

[658] Inioluwa Deborah Raji and Joy Buolamwini. "Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products". In: *AAAI/ACM Conf. on AI Ethics and Society*. Vol. 1. 2019.

[659] Inioluwa Deborah Raji, Emily Denton, Emily M Bender, Alex Hanna, and Amandalynne Paullada. "AI and the Everything in the Whole Wide World Benchmark". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[660] Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. "Saving face: Investigating the ethical concerns of facial recognition auditing". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 145–151.

[661] Inioluwa Deborah Raji, I Elizabeth Kumar, Aaron Horowitz, and Andrew Selbst. "The fallacy of AI functionality". In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 959–972.

[662] Inioluwa Deborah Raji, Morgan Klaus Scheuerman, and Razvan Amironesei. "You can't sit with us: Exclusionary pedagogy in ai ethics education". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 515–525.

[663] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 33–44.

[664] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. "Machine learning in medicine". In: *New England Journal of Medicine* 380.14 (2019), pp. 1347–1358.

[665] Bogdana Rakova, Jingying Yang, Henriette Cramer, and Rumman Chowdhury. "Where Responsible AI Meets Reality: Practitioner Perspectives on Enablers for Shifting Organizational Practices". In: *Proc. ACM Hum.-Comput. Interact.* 5.CSCW1 (Apr. 2021). DOI: 10.1145/3449081. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3449081.

[666] Bogdana Rakova, Jingying Yang, Henriette Cramer, and Rumman Chowdhury. "Where responsible AI meets reality: Practitioner perspectives on enablers for shifting organizational practices". In: *Proceedings of the ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–23.

[667] Arvind Ramanathan, Laura Pullum, Zubir Husein, Sunny Raj, Neslisah Torosdagli, Sumanta Pattanaik, and Sumit K Jha. "Adversarial attacks on computer vision algorithms using natural perturbations". In: *2017 Tenth International Conference on Contemporary Computing (IC3)*. IEEE. 2017, pp. 1–6.

[668] Bing Ran and P R Duimering. “Conceptual combination: Models, theories and controversies”. In: *Intl. Journal of Cognitive Linguistics* 1.1 (2010), pp. 65–90.

[669] Ayushi Rastogi. “Do Biases Related to Geographical Location Influence Work-related Decisions in GitHub?” In: *Proceedings of the 38th International Conference on Software Engineering Companion*. ICSE ’16. Austin, Texas: ACM, 2016, pp. 665–667. ISBN: 978-1-4503-4205-6. DOI: 10.1145/2889160.2891035. URL: http://doi.acm.org/10.1145/2889160.2891035.

[670] Jonas Rauber, Wieland Brendel, and Matthias Bethge. “Foolbox: A python toolbox to benchmark the robustness of machine learning models”. In: *arXiv preprint arXiv:1707.04131* (2017).

[671] Nathalie Rauschmayr, Vikas Kumar, Rahul Huilgol, Andrea Olgiati, Satadal Bhattacharjee, et al. “Amazon SageMaker Debugger: A System for Real-Time Insights into Machine Learning Model Training”. In: *Proceedings of Machine Learning and Systems* 3 (2021).

[672] V C Raykar, S Yu, and al. “Learning from crowds”. In: *JMLR* 11.Apr (2010).

[673] Donghao Ren, Saleema Amershi, Bongshin Lee, Jina Suh, and Jason D Williams. “Squares: Supporting interactive performance analysis for multiclass classifiers”. In: *IEEE transactions on visualization and computer graphics* 23.1 (2016), pp. 61–70.

[674] Ashkan Rezaei, Anqi Liu, Omid Memarrast, and Brian D Ziebart. “Robust fairness under covariate shift”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 11. 2021, pp. 9419–9427.

[675] M. Rezvan, S. Shekarpour, L. Balasuriya, K. Thirunarayan, V. L. Shalin, and A. Sheth. “A Quality Type-aware Annotated Corpus and Lexicon for Harassment Research”. In: *Proceedings of the 10th ACM Conference on Web Science - WebSci ’18*. New York, USA: ACM Press, 2018, pp. 33–36. ISBN: 9781450355636. DOI: 10.1145/3201064.3201103. URL: http://dl.acm.org/citation.cfm?doid=3201064.3201103.

[676] M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. F. Almeida, and W. Meira. “"Like Sheep Among Wolves": Characterizing Hateful Users on Twitter”. In: *Proceedings of WSDM workshop on Misinformation and Misbehavior Mining on the Web (MIS2)* (New York, NY, USA). ACM, 2018. DOI: https://doi.org/10.475/123_4.

[677] Marco Tulio Ribeiro et al. “Why should i trust you? Explaining the predictions of any classifier”. In: *SIGKDD*. 2016, pp. 1135–1144.

[678] John Richards, David Piorkowski, Michael Hind, Stephanie Houde, and Aleksandra Mojsilović. “A methodology for creating AI FactSheets”. In: *arXiv preprint arXiv:2006.13796* (2020).

[679] Brianna Richardson, Jean Garcia-Gathright, Samuel F Way, Jennifer Thom, and Henriette Cramer. “Towards Fairness in Practice: A Practitioner-Oriented Rubric for Evaluating Fair ML Toolkits”. In: *CHI*. 2021, pp. 1–13.

[680] Matthew Richardson and Pedro Domingos. “Markov logic networks”. In: *Machine learning* 62.1 (2006), pp. 107–136.

[681] Julian Risch, Robin Ruff, and Ralf Krestel. “Offensive Language Detection Explained”. In: *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. 2020, pp. 137–143.

[682] Ronald E. Robertson, Shan Jiang, Kenneth Joseph, Lisa Friedland, David Lazer, and Christo Wilson. “Auditing Partisan Audience Bias Within Google Search”. In: *Proc. ACM Hum.-Comput. Interact.* 2.CSCW (Nov. 2018), 148:1–148:22. ISSN: 2573-0142. DOI: 10.1145/3274417. URL: http://doi.acm.org/10.1145/3274417.

[683] Christos Rodosthenous and Loizos Michael. “A hybrid approach to commonsense knowledge acquisition”. In: *STAIRS 2016*. IOS Press, 2016, pp. 111–122.

[684] Nestor Rodriguez and Sergio Rojas-Galeano. “Fighting Adversarial Attacks on Online Abusive Language Moderation”. In: *Workshop on Engineering Applications*. Springer. 2018, pp. 480–493.

[685] Andrea Romei and Salvatore Ruggieri. “A multidisciplinary survey on discrimination analysis”. In: *The Knowledge Engineering Review* 29.5 (2014), pp. 582–638.

[686] H. Rosa, J. P. Carvalho, P. Calado, B. Martins, R. Ribeiro, and L. Coheur. “Using Fuzzy Fingerprints for Cyberbullying Detection in Social Networks”. In: *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, July 2018, pp. 1–7. ISBN: 978-1-5090-6020-7. DOI: 10.1109/FUZZ-IEEE.2018.8491557. URL: https://ieeexplore.ieee.org/document/8491557/.

[687] Drew Roselli, Jeanna Matthews, and Nisha Talagala. “Managing bias in AI”. In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 539–544.

[688] A S Ross, M C Hughes, and F Doshi-V. “Right for the right reasons: training differentiable models by constraining their explanations”. In: *IJCAI*. 2017, pp. 2662–2670.

[689] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki. “Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis”. In: *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer Mediated Communication, vol. 17* (Bochum, Germany). Ed. by Beisswenger M., Wojatzki M., and Zesch T. Bochumer Linguistischer Arbeitsberichte, 2016, pp. 6–9.

[690] Joel Ross, Lilly Irani, M Six Silberman, Andrew Zaldivar, and Bill Tomlinson. “Who are the crowdworkers? Shifting demographics in Mechanical Turk”. In: *CHI’10 extended abstracts on Human factors in computing systems*. 2010, pp. 2863–2872.

[691] Nirmal Roy, Agathe Balayn, David Maxwell, and Claudia Hauff. “Hear Me Out: A Study on the Use of the Voice Modality for Crowdsourced Relevance Assessments”. In: *Special Interest Group on Information Retrieval (SIGIR)*. 2023.

[692] T. Roy, J. McClendon, and L. Hodges. “Analyzing Abusive Text Messages to Detect Digital Dating Abuse”. In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, June 2018, pp. 284–293. ISBN: 978-1-5386-5377-7. DOI: 10.1109/ICHI.2018.00039. URL: https://ieeexplore.ieee.org/document/8419372/.

[693] Giovanni Rubeis, Keerthi Dubbala, and Ingrid Metzler. “‘Democratization’ in the context of Artificial intelligence and healthcare: Mapping an elusive term”. In: *Frontiers in Genetics* (2022), p. 2144.

[694] Bonnie Ruberg and Spencer Ruelos. “Data for queer lives: How LGBTQ gender and sexuality identities challenge norms of demographics”. In: *Big Data & Society* 7.1 (2020), p. 2053951720933286.

[695] Salvatore Ruggieri, Sara Hajian, Faisal Kamiran, and Xiangliang Zhang. “Anti-discrimination analysis using privacy attack strategies”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2014, pp. 694–710.

[696] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. “Data mining for discrimination discovery”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 4.2 (2010), p. 9.

[697] Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. “DCUBE: discrimination discovery in databases”. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*. 2010, pp. 1127–1130. DOI: 10.1145/1807167.1807298. URL: https://doi.org/10.1145/1807167.1807298.

[698] Salvatore Ruggieri and Franco Turini. “A KDD process for discrimination discovery”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2016, pp. 249–253.

[699] O Russakovsky, J Deng, H Su, J Krause, and al. “Imagenet large scale visual recognition challenge”. In: *IJCV* 115.3 (2015), pp. 211–252.

[700] Maciej Rybinski, William Miller, Javier Del Ser, Miren Nekane Bilbao, and José F Aldana-Montes. “On the Design and Tuning of Machine Learning Models for Language Toxicity Classification in Online Platforms”. In: *International Symposium on Intelligent and Distributed Computing*. Springer. 2018, pp. 329–343.

[701] Aastha Sahni and Naveen Raja. “Analyzation and Detection of Cyberbullying: A Twitter Based Indian Case Study”. In: *International Conference on Recent Developments in Science, Engineering and Technology*. Springer. 2017, pp. 484–497.

[702] S. Salawu, Y. He, and J. Lumsden. “Approaches to Automated Detection of Cyberbullying: A Survey”. In: *IEEE Transactions on Affective Computing* 1 (2017), pp. 1–1.

[703] H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths. “A Web of Hate: Tackling Hateful Speech in Online Social Spaces”. In: *First Workshop on Text Analytics for Cybersecurity and Online Safety at LREC 2016*. 2016.

[704] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. “Aequitas: A bias and fairness audit toolkit”. In: *arXiv preprint arXiv:1811.05577* (2018).

[705] Babak Salimi, Corey Cole, Peter Li, Johannes Gehrke, and Dan Suciu. "HypDB: a demonstration of detecting, explaining and resolving bias in OLAP queries". In: *Proceedings of the VLDB Endowment* 11.12 (2018), pp. 2062–2065.

[706] Babak Salimi, Johannes Gehrke, and Dan Suciu. "Bias in OLAP Queries: Detection, Explanation, and Removal". In: *Proceedings of the 2018 International Conference on Management of Data*. SIGMOD '18. Houston, TX, USA: ACM, 2018, pp. 1021–1035. ISBN: 978-1-4503-4703-7. DOI: 10.1145/3183713.3196914. URL: http://doi.acm.org/10.1145/3183713.3196914.

[707] Babak Salimi, Bill Howe, and Dan Suciu. "Data Management for Causal Algorithmic Fairness". In: *arXiv preprint arXiv:1908.07924* (2019).

[708] Babak Salimi, Bill Howe, and Dan Suciu. "Database Repair Meets Algorithmic Fairness". In: *SIGMOD Rec.* 49.1 (Sept. 2020), pp. 34–41. ISSN: 0163-5808. DOI: 10.1145/3422648.3422657. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3422648.3422657.

[709] Babak Salimi, Harsh Parikh, Moe Kayali, Lise Getoor, Sudeepa Roy, and Dan Suciu. "Causal Relational Learning". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD '20. Portland, OR, USA: Association for Computing Machinery, 2020, pp. 241–256. ISBN: 9781450367356. DOI: 10.1145/3318464.3389759. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3318464.3389759.

[710] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. "Interventional Fairness: Causal Database Repair for Algorithmic Fairness". In: *Proceedings of the 2019 International Conference on Management of Data*. SIGMOD '19. Amsterdam, Netherlands: ACM, 2019, pp. 793–810. ISBN: 978-1-4503-5643-5. DOI: 10.1145/3299869.3319901. URL: http://doi.acm.org/10.1145/3299869.3319901.

[711] Iflaah Salman. "Cognitive Biases in Software Quality and Testing". In: *Proceedings of the 38th International Conference on Software Engineering Companion*. ICSE '16. Austin, Texas: ACM, 2016, pp. 823–826. ISBN: 978-1-4503-4205-6. DOI: 10.1145/2889160.2889265. URL: http://doi.acm.org/10.1145/2889160.2889265.

[712] Joni Salminen, Soon-Gyo Jung, and Bernard J. Jansen. "Detecting Demographic Bias in Automatically Generated Personas". In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI EA '19. Glasgow, Scotland Uk: ACM, 2019, LBW0122:1–LBW0122:6. ISBN: 978-1-4503-5971-9. DOI: 10.1145/3290607.3313034. URL: http://doi.acm.org/10.1145/3290607.3313034.

[713] P. Salunkhe, S. Bharne, and P. Padiya. "Filtering unwanted messages from OSN walls". In: *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*. IEEE, Feb. 2016. ISBN: 978-1-5090-2084-3. DOI: 10.1109/ICICCS.2016.7542319. URL: http://ieeexplore.ieee.org/document/7542319/.

[714] Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. ""Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI". In: *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–15.

[715] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature, 2019.

[716] Wojciech Samek and Klaus-Robert Müller. "Towards explainable artificial intelligence". In: *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, 2019, pp. 5–22.

[717] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models". In: *arXiv preprint arXiv:1708.08296* (2017).

[718] Javier Sánchez-Monedero, Lina Dencik, and Lilian Edwards. "What does it mean to'solve'the problem of discrimination in hiring? Social, technical and legal perspectives from the UK on automated hiring systems". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 458–468.

[719] Conrad Sanderson, David Douglas, Qinghua Lu, Emma Schleiger, Jon Whittle, Justine Lacey, Glenn Newnham, Stefan Hajkowicz, Cathy Robinson, and David Hansen. "AI ethics principles in practice: Perspectives of designers and developers". In: *arXiv preprint arXiv:2112.07467* (2021).

[720] Filippo Santoni de Sio. "The European Commission report on ethics of connected and automated vehicles and the future of ethics of transportation". In: *Ethics and Information Technology* 23.4 (2021), pp. 713–726.

[721] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. "The risk of racial bias in hate speech detection". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 1668–1678.

[722] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. "SocialIQA: Commonsense Reasoning about Social Interactions". In: *Conf. on Empirical Methods in Natural Language Processing*. 2019.

[723] Edward Sapir. *Language: An introduction to the study of speech*. Courier Corporation, 2004.

[724] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C Parkes, and Yang Liu. "How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 99–106.

[725] Burcu Sayin, Fabio Casati, Andrea Passerini, Jie Yang, and Xinyue Chen. "Rethinking and Recomputing the Value of ML Models". In: *arXiv preprint arXiv:2209.15157* (2022).

[726] Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. "FairPrep: Promoting Data to a First-Class Citizen in Studies on Fairness-Enhancing Interventions". In: *Proceedings of the 23nd International Conference on Extending Database Technology, EDBT, 2020*. 2020.

[727] A. Schmidt and M. Wiegand. "A survey on hate speech detection using natural language processing". In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics, Valencia, Spain*. 2017, pp. 1–10.

[728] Frank Schneider, Felix Dangel, and Philipp Hennig. "Cockpit: A Practical Debugging Tool for Training Deep Neural Networks". In: (2021).

[729] Eldon Schoop, Forrest Huang, and Björn Hartmann. "UMLAUT: Debugging Deep Learning Programs using Program Structure and Model Behavior". In: (2021).

[730] Tina Schuh and Stephan Dreiseitl. "Evaluating novel features for aggressive language detection". In: *International Conference on Speech and Computer*. Springer. 2018, pp. 585–595.

[731] Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. "Fairness and abstraction in sociotechnical systems". In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 59–68.

[732] R R Selvaraju, M Cogswell, A Das, and al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proc. of ICCV*. 2017, pp. 618–626.

[733] Suin Seo and Sung-Bea Cho. "Offensive sentence classification using character-level CNN and transfer learning with fake sentences". In: *International Conference on Neural Information Processing*. Springer. 2017, pp. 532–539.

[734] Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. "Assisting the Human Fact-Checkers: Detecting All Previously Fact-Checked Claims in a Document". In: *arXiv preprint arXiv:2109.07410* (2021).

[735] Shreya Shankar, Rolando Garcia, Joseph M Hellerstein, and Aditya G Parameswaran. "Operationalizing Machine Learning: An Interview Study". In: *arXiv preprint arXiv:2209.09125* (2022).

[736] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D Sculley. "No classification without representation: Assessing geodiversity issues in open data sets for the developing world". In: *arXiv preprint arXiv:1711.08536* (2017).

[737] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. *Do Image Classifiers Generalize Across Time?* 2019. DOI: 10.48550/ARXIV.1906.02168. URL: https://arxiv.org/abs/1906.02168.

[738] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. "Evaluating machine accuracy on imagenet". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8634–8644.

[739] R Benjamin Shapiro, Rebecca Fiebrink, and Peter Norvig. “How machine learning impacts the undergraduate computing curriculum”. In: *Communications of the ACM* 61.11 (2018), pp. 27–29.

[740] Shahin Sharifi, Sihang Qiu, Burcu Sayin, Agathe Balayn, Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. “Perspective, Leveraging Human Understanding for Identifying and Characterizing Image Atypicality”. In: *Conference on Intelligent User Interface (IUI)*. 2023.

[741] Shahin Sharifi Noorian, Sihang Qiu, Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. “What Should You Know? A Human-In-the-Loop Approach to Unknown Unknowns Characterization in Image Recognition”. In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 882–892.

[742] Sima Sharifirad and Stan Matwin. “Using Attention-based Bidirectional LSTM to Identify Different Categories of Offensive Language Directed Toward Female Celebrities”. In: *Proceedings of the 2019 Workshop on Widening NLP*. 2019, pp. 46–48.

[743] Hong Shen, Wesley H Deng, Aditi Chattopadhyay, Zhiwei Steven Wu, Xu Wang, and Haiyi Zhu. “Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation”. In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 850–861.

[744] Max W Shen. “Trust in AI: Interpretability is not necessary or sufficient, while black-box interaction is necessary and sufficient”. In: *arXiv preprint arXiv:2202.05302* (2022).

[745] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. “Towards out-of-distribution generalization: A survey”. In: *arXiv preprint arXiv:2108.13624* (2021).

[746] Victor S Sheng and Jing Zhang. “Machine learning with crowdsourcing: A brief summary of the past research and future directions”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 9837–9843.

[747] Ben Shneiderman. “Human-centered artificial intelligence: Reliable, safe & trustworthy”. In: *International Journal of Human–Computer Interaction* 36.6 (2020), pp. 495–504.

[748] Reza Shokri, Martin Strobel, and Yair Zick. “On the privacy risks of model explanations”. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 231–241.

[749] Nischal Shrestha, Titus Barik, and Chris Parnin. “Remote, but Connected: How# TidyTuesday Provides an Online Community of Practice for Data Scientists.” In: *ACM on Human-Computer Interaction* 5.CSCW1 (2021), pp. 1–31.

[750] A Shrikumar, P G, and A Kundaje. “Learning Important Features Through Propagating Activation Differences”. In: *ICML*. 2017, pp. 3145–3153.

[751] Dule Shu, Nandi O Leslie, Charles A Kamhoua, and Conrad S Tucker. “Generative adversarial attacks against intrusion detection systems using active learning”. In: *Workshop on Wireless Security and Machine Learning*. 2020, pp. 1–6.

[752] Jake Silberg and James Manyika. “Notes from the AI frontier: Tackling bias in AI (and in humans)”. In: *McKinsey Global Institute* 1.6 (2019).

[753] Leandro Silva, Mainack Mondal, Denzil Correa, Fabricio Benevenuto, and Ingmar Weber. “Analyzing the Targets of Hate in Online Social Media”. In: *10th International AAAI Conference on Web and Social Media*. AAAI. 2016, pp. 687–690.

[754] K Simonyan, A Vedaldi, and A Zisserman. “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: *ICLR*. 2014.

[755] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[756] Elena Simperl, Maribel Acosta, and Fabian Flöck. “Knowledge engineering via human computation”. In: *Handbook of Human Computation*. Springer, 2013, pp. 131–151.

[757] Ashudeep Singh and Thorsten Joachims. “Fairness of Exposure in Rankings”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 2018, pp. 2219–2228. DOI: 10.1145/3219819.3220088. URL: https://doi.org/10.1145/3219819.3220088.

[758] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. “Boosting Robustness Certification of Neural Networks”. In: *ICLR*. 2019.

[759] Harvineet Singh, Rina Singh, Vishwali Mhasawade, and Rumi Chunara. “Fairness violations and mitigation under covariate shift”. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 3–13.

[760] Push Singh et al. “The public acquisition of commonsense knowledge”. In: *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. 2002.

[761] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. “Open mind common sense: Knowledge acquisition from the general public”. In: *OTM Confederated Intl. Conf. "On the Move to Meaningful Internet Systems"*. Springer. 2002, pp. 1223–1237.

[762] Vivek K Singh and Connor Hofenbitzer. “Fairness across network positions in cyberbullying detection algorithms”. In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2019, pp. 557–559.

[763] Sahil Singla, Besmira Nushi, S Shah, E Kamar, and E Horvitz. “Understanding Failures of Deep Networks via Robust Feature Extraction”. In: (2020).

[764] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. “Variational adversarial active learning”. In: *ICCV*. 2019, pp. 5972–5981.

[765] Sanchit Sinha, Mohit Agarwal, Mayank Vatsa, Richa Singh, and Saket Anand. “Exploring Bias in Primate Face Detection and Recognition”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0.

[766] Leon Sixt, Maximilian Granz, and Tim Landgraf. “When Explanations Lie: Why Many Modified BP Attributions Fail”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9046–9057.

[767] Dylan Slack, Sorelle A Friedler, and Emile Givental. “Fairness warnings and Fair-MAML: learning fairly with minimal data”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 200–209.

[768] M Sloane, E Moss, O Awomolo, and L Forlano. “Participation is not a Design Fix for Machine Learning”. In: *Participatory Approaches to Machine Learning* (2020).

[769] D Smilkov and al. “SmoothGrad: removing noise by adding noise”. In: (2017).

[770] Carol J Smith. “Designing trustworthy AI: A human-machine teaming framework to guide development”. In: *arXiv preprint arXiv:1910.03515* (2019).

[771] Jessie J Smith, Saleema Amershi, Solon Barocas, Hanna Wallach, and Jennifer Wortman Vaughan. “Real ml: Recognizing, exploring, and articulating limitations of machine learning research”. In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2022, pp. 587–597.

[772] Nathalie A Smuha. “The EU approach to ethics guidelines for trustworthy artificial intelligence”. In: *Computer Law Review International* 20.4 (2019), pp. 97–106.

[773] Kacper Sokol and Peter Flach. “Explainability fact sheets: a framework for systematic assessment of explainable approaches”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 2020, pp. 56–67.

[774] Jacob Solomon. “Customization Bias in Decision Support Systems”. In: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’14. Toronto, Ontario, Canada: ACM, 2014, pp. 3065–3074. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557211. URL: http://doi.acm.org/10.1145/2556288.2557211.

[775] Olivia Solon. *Facebook is hiring moderators. But is the job too gruesome to handle?* https://www.theguardian.com/technology/2017/may/04/facebook-content-moderators-ptsd-psychological-dangers, Last accessed on 2020-03-16. 2017.

[776] Nasim Sonboli and Robin Burke. “Localized Fairness in Recommender Systems”. In: *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. UMAP’19 Adjunct. Larnaca, Cyprus: ACM, 2019, pp. 295–300. ISBN: 978-1-4503-6711-0. DOI: 10.1145/3314183.3323845. URL: http://doi.acm.org/10.1145/3314183.3323845.

[777] S. O. Sood, J. Antin, and E. F. Churchill. “Using Crowdsourcing to Improve Profanity Detection.” In: *AAAI Spring Symposium: Wisdom of the Crowd*. Vol. 12. 2012, p. 06.

[778] S. O. Sood, E. F. Churchill, and J. Antin. “Automatic identification of personal insults on social news sites”. In: *Journal of the Association for Information Science and Technology* 63.2 (2012), pp. 270–285.

[779] Sara Owsley Sood, Judd Antin, and Elizabeth F. Churchill. “Profanity use in online communities”. In: *CHI Conference on Human Factors in Computing Systems, CHI ’12, Austin, TX, USA - May 05 - 10, 2012*. Ed. by Joseph A. Konstan, Ed H. Chi, and Kristina Höök. ACM, 2012, pp. 1481–1490. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208610. URL: https://doi.org/10.1145/2207676.2208610.

[780] Robyn Speer, Joshua Chin, and Catherine Havasi. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”. In: *Thirty-First AAAI Conf. on Artificial Intelligence*. AAAI Press, 2017, pp. 4444–4451.

[781] Robyn Speer, Jayant Krishnamurthy, Catherine Havasi, Dustin Smith, Henry Lieberman, and Kenneth Arnold. “An interface for targeted collection of common sense knowledge using a mixture model”. In: *14th Intl. Conf. on intelligent user interfaces*. 2009, pp. 137–146.

[782] Brendan Spillane, Séamus Lawless, and Vincent Wade. “Measuring Bias in News Websites, Towards a Model for Personalization”. In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. UMAP ’17. Bratislava, Slovakia: ACM, 2017, pp. 387–388. ISBN: 978-1-4503-4635-1. DOI: 10.1145/3079628.3079647. URL: http://doi.acm.org/10.1145/3079628.3079647.

[783] Aaron Springer, Jean Garcia-Gathright, and Henriette Cramer. “Assessing and Addressing Algorithmic Bias-But Before We Get There...” In: *2018 AAAI Spring Symposium Series*. 2018.

[784] Anna Squicciarini, Sarah Rajtmajer, Y Liu, and Christopher Griffin. “Identification and characterization of cyberbullying dynamics in an online social network”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 280–285.

[785] Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. “Robustness to spurious correlations via human annotations”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9109–9119.

[786] Megha Srivastava, Hoda Heidari, and Andreas Krause. “Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2459–2468.

[787] Joe Stacey, Yonatan Belinkov, and Marek Rei. “Supervising model attention with human explanations for robust natural language inference”. In: *AAAI*. Vol. 36. 10. 2022, pp. 11349–11357.

[788] Thilo Stadelmann, Julian Keuzenkamp, Helmut Grabner, and Christoph Würsch. “The AI-atlas: didactics for teaching AI and machine learning on-site, online, and hybrid”. In: *Education Sciences* 11.7 (2021), p. 318.

[789] Pieter Jan Stappers and Elisa Giaccardi. “Research through design”. In: *The encyclopedia of human-computer interaction*. The Interaction Design Foundation, 2017, pp. 1–94.

[790] Luke Stark and Jesse Hoey. “The ethics of emotion in artificial intelligence systems”. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 782–793.

[791] Christopher Starke, Janine Baleis, Birte Keller, and Frank Marcinkowski. “Fairness perceptions of algorithmic decision-making: A systematic review of the empirical literature”. In: *arXiv preprint arXiv:2103.12016* (2021).

[792] Marc Steen. “Co-design as a process of joint inquiry and imagination”. In: *Design Issues* 29.2 (2013), pp. 16–28.

[793] Pierre Stock and Moustapha Cisse. “Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 498–512.

[794] Julia Stoyanovich and Bill Howe. “Nutritional Labels for Data and Models”. In: *Data Engineering* (2019), p. 13.

[795] Julia Stoyanovich, Bill Howe, Serge Abiteboul, Gerome Miklau, Arnaud Sahuguet, and Gerhard Weikum. "Fides: Towards a Platform for Responsible Data Science". In: *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. SSDBM '17. Chicago, IL, USA: ACM, 2017, 26:1–26:6. ISBN: 978-1-4503-5282-6. DOI: 10.1145/3085504.3085530. URL: http://doi.acm.org/10.1145/3085504.3085530.

[796] Julia Stoyanovich, Bill Howe, and HV Jagadish. "Responsible data management". In: *Proceedings of the VLDB Endowment* 13.12 (2020), pp. 3474–3488.

[797] Julia Stoyanovich, Ke Yang, and H. V. Jagadish. "Online Set Selection with Fairness and Diversity Constraints". In: *Proceedings of the 21th International Conference on Extending Database Technology, EDBT 2018, Vienna, Austria, March 26-29, 2018*. 2018, pp. 241–252. DOI: 10.5441/002/edbt.2018.22. URL: https://doi.org/10.5441/002/edbt.2018.22.

[798] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. "Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records". In: *BioMed research international* 2014 (2014).

[799] Anselm L Strauss. *Qualitative analysis for social scientists*. Cambridge university press, 1987.

[800] E Štrumbelj and I Kononenko. "Explaining prediction models and individual predictions with feature contributions". In: *Knowledge and information systems* (2014).

[801] David Stutz, Matthias Hein, and Bernt Schiele. "Confidence-calibrated adversarial training: Generalizing to unseen attacks". In: *ICML*. PMLR. 2020, pp. 9155–9166.

[802] Dong Su, H. Zhang, H. Chen, J. Yi, Pin-Yu Chen, and Yupeng Gao. "Is Robustness the Cost of Accuracy? A Comprehensive Study on the Robustness of 18 Deep Image Classification Models". In: *ECCV*. Cham: Springer, 2018, pp. 644–661.

[803] Hariharan Subramonyam, Jane Im, Colleen Seifert, and Eytan Adar. "Solving Separation-of-Concerns Problems in Collaborative Design of Human-AI Systems through Leaky Abstractions". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517537. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3491102.3517537.

[804] Tom Sühr, Asia J. Biega, Meike Zehlike, Krishna P. Gummadi, and Abhijnan Chakraborty. "Two-Sided Fairness for Repeated Matchings in Two-Sided Markets: A Case Study of a Ride-Hailing Platform". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 2019, pp. 3082–3092. DOI: 10.1145/3292500.3330793. URL: https://doi.org/10.1145/3292500.3330793.

[805] Elisabeth Sulmont, Elizabeth Patitsas, and Jeremy R Cooperstock. "What is hard about teaching machine learning to non-majors? Insights from classifying instructors' learning goals". In: *Trans. on Computing Education (TOCE)* 19.4 (2019), pp. 1–16.

[806] Chenkai Sun, Abolfazl Asudeh, HV Jagadish, Bill Howe, and Julia Stoyanovich. "Mithralabel: Flexible dataset nutritional labels for responsible data science". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2019, pp. 2893–2896.

[807] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. "Mitigating Gender Bias in Natural Language Processing: Literature Review". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 1630–1640.

[808] Xiaobing Sun, Tianchi Zhou, Gengjie Li, Jiajun Hu, Hui Yang, and Bin Li. "An empirical study on real bugs for machine learning programs". In: *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE. 2017, pp. 348–357.

[809] M Sundararajan and al. "Axiomatic Attribution for Deep Networks". In: *ICML*. 2017.

[810] Harini Suresh, Steven R Gomez, Kevin K Nam, and A Satyanarayan. "Beyond Expertise and Roles: A Framework to Characterize the Stakeholders of Interpretable Machine Learning and their Needs". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–16.

[811] Harini Suresh and John Guttag. “A framework for understanding sources of harm throughout the machine learning life cycle”. In: *Equity and access in algorithms, mechanisms, and optimization*. 2021, pp. 1–9.

[812] Harini Suresh and John V. Guttag. “A Framework for Understanding Unintended Consequences of Machine Learning”. In: *CoRR* abs/1901.10002 (2019). arXiv: 1901.10002. URL: http://arxiv.org/abs/1901.10002.

[813] C Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. “Rethinking the inception architecture for computer vision”. In: *Proc. of the IEEE CVPR*. 2016, pp. 2818–2826.

[814] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. *Intriguing properties of neural networks*. 2013. DOI: 10.48550/ARXIV.1312.6199. URL: https://arxiv.org/abs/1312.6199.

[815] Ki Hyun Tae, Yuji Roh, Young Hun Oh, Hyunsu Kim, and Steven Euijong Whang. “Data Cleaning for Accurate, Fair, and Robust Models: A Big Data - AI Integration Approach”. In: *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning*. DEEM’19. Amsterdam, Netherlands: ACM, 2019, 5:1–5:4. ISBN: 978-1-4503-6797-4. DOI: 10.1145/3329486.3329493. URL: http://doi.acm.org/10.1145/3329486.3329493.

[816] N. Tahmasbi and A. Fuchsberger. “Challenges and Future Directions of Automated Cyberbullying Detection”. In: *AMCIS 2018 Proceedings* (Aug. 2018). URL: https://aisel.aisnet.org/amcis2018/SocialComputing/Presentations/10.

[817] N. Tahmasbi and E. Rastegari. “A Socio-contextual Approach in Automated Detection of Cyberbullying”. In: *Hawaii International Conference on System Sciences 2018 (HICSS-51)* (2018). URL: https://aisel.aisnet.org/hicss-51/dsm/social%7B%5C_%7Dmedia%7B%5C_%7Dculture/3.

[818] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. “CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge”. In: *2019 NAACL-HLT , Minneapolis, MN, USA, June 2-7, 2019, Volume 1*. ACL, 2019, pp. 4149–4158.

[819] Damian A Tamburri. “Sustainable mlops: Trends and challenges”. In: *2020 22nd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC)*. IEEE. 2020, pp. 17–23.

[820] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. “Measuring Robustness to Natural Distribution Shifts in Image Classification”. In: *NeurIPS*. Vol. 33. Curran Associates, 2020, pp. 18583–18599. URL: https://proceedings.neurips.cc/paper/2020/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

[821] Jasmine Tata. “The structure and phenomenon of sexual harassment: Impact of category of sexually harassing behavior, gender, and hierarchical level”. In: *Journal of Applied Social Psychology* 23.3 (1993), pp. 199–211.

[822] P. L. Teh, C-B. Cheng, and W. M. Chee. “Identifying and Categorising Profane Words in Hate Speech”. In: *Proceedings of the 2nd International Conference on Compute and Data Analysis - ICCDA 2018*. New York, USA: ACM Press, 2018. ISBN: 9781450363594. DOI: 10.1145/3193077.3193078. URL: http://dl.acm.org/citation.cfm?doid=3193077.3193078.

[823] Ferdian Thung, Shaowei Wang, David Lo, and Lingxiao Jiang. “An empirical study of bugs in machine learning systems”. In: *2012 IEEE 23rd International Symposium on Software Reliability Engineering*. IEEE. 2012, pp. 271–280.

[824] Andrea Tocchetti and Marco Brambilla. “The role of human knowledge in explainable ai”. In: *Data* 7.7 (2022), p. 93.

[825] Andrea Tocchetti, Lorenzo Corti, Agathe Balayn, Mireia Yurrita, Philip Lippmann, Marco Brambilla, and Jie Yang. “AI Robustness: a Human-Centered Perspective on Technological Challenges and Opportunities”. In: *arXiv preprint arXiv:2210.08906* (2022).

[826] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. “A deeper look at dataset bias”. In: *Domain adaptation in computer vision applications*. Springer, 2017, pp. 37–55.

[827] Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. “What clinicians want: contextualizing explainable machine learning for clinical end use”. In: *Machine learning for healthcare conference*. PMLR. 2019, pp. 359–380.

[828] Antonio Torralba, Alexei A Efros, et al. “Unbiased look at dataset bias.” In: *CVPR*. Vol. 1. 2. Citeseer. 2011, p. 7.

[829] A. Tsesis. “Hate in cyberspace: Regulating hate speech on the Internet”. In: *San Diego L. Rev.* 38 (2001), p. 817.

[830] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. “From ImageNet to Image Classification: Contextualizing Progress on Benchmarks”. In: *arXiv preprint arXiv:2005.11295* (2020).

[831] Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. “A dictionary-based approach to racism detection in Dutch social media”. In: *Proceedings of the first Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS 2016)/Daelemans*. 2016, pp. 1–7.

[832] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. “Automated Directed Fairness Testing”. In: *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. ASE 2018. Montpellier, France: ACM, 2018, pp. 98–108. ISBN: 978-1-4503-5937-5. DOI: 10.1145/3238147.3238165. URL: http://doi.acm.org/10.1145/3238147.3238165.

[833] Stefanie Ullmann and Marcus Tomalin. “Quarantining online hate speech: technical and ethical perspectives”. In: *Ethics and Information Technology* 22.1 (2020), pp. 69–80.

[834] Berk Ustun, Alexander Spangher, and Yang Liu. “Actionable recourse in linear classification”. In: *Proceedings of the conference on fairness, accountability, and transparency*. 2019, pp. 10–19.

[835] Mohammad Moeen Valipoor and Angélica de Antonio. “Recent trends in computer vision-driven scene understanding for VI/blind users: a systematic mapping”. In: *Universal Access in the Information Society* (2022), pp. 1–23.

[836] Niels Van Berkel, Jorge Goncalves, Danula Hettiachchi, Senuri Wijenayake, Ryan M Kelly, and Vassilis Kostakos. “Crowdsourcing perceptions of fair predictors for machine learning: A recidivism case study”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–21.

[837] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[838] Niels Van Doorn, Fabian Ferrari, and Mark Graham. “Migration and migrant labour in the gig economy: An intervention”. In: *Work, Employment and Society* (2020), p. 09500170221096581.

[839] Cynthia Van Hee, Gilles Jacobs, Chris Emmery, Bart Desmet, Els Lefever, Ben Verhoeven, Guy De Pauw, Walter Daelemans, and Véronique Hoste. “Automatic detection of cyberbullying in social media text”. In: *PloS one* 13.10 (2018).

[840] Colin Vandenhof. “A Hybrid Approach to Identifying Unknown Unknowns of Predictive Models”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*. Vol. 7. 1. 2019, pp. 180–187.

[841] Marisa Vasconcelos, Carlos Cardonha, and Bernardo Gonçalves. “Modeling Epistemological Principles for Bias Mitigation in AI Systems: An Illustration in Hiring Decisions”. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. AIES ’18. New Orleans, LA, USA: ACM, 2018, pp. 323–329. ISBN: 978-1-4503-6012-8. DOI: 10.1145/3278721.3278751. URL: http://doi.acm.org/10.1145/3278721.3278751.

[842] Sriram Vasudevan and Krishnaram Kenthapadi. “Lift: A scalable framework for measuring fairness in ml applications”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2020, pp. 2773–2780.

[843] Jennifer Wortman Vaughan. “Making better use of the crowd: How crowdsourcing can advance machine learning research”. In: *JMLR* 18.193 (2018), pp. 1–46.

[844] Michael Veale, Reuben Binns, and Lilian Edwards. “Algorithms that remember: model inversion attacks and data protection law”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2133 (2018), p. 20180083.

[845] Michael Veale and Frederik Zuiderveen Borgesius. “Demystifying the Draft EU Artificial Intelligence Act—Analysing the good, the bad, and the unclear elements of the proposed approach”. In: *Computer Law Review International* 22.4 (2021), pp. 97–112.

[846] Michael Veale, Max Van Kleek, and Reuben Binns. “Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making”. In: *Proceedings of the 2018 chi conference on human factors in computing systems*. 2018, pp. 1–14.

[847] Raphael Velt, Steve Benford, and Stuart Reeves. “Translations and boundaries in the gap between HCI theory and design practice”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 27.4 (2020), pp. 1–28.

[848] Sahil Verma and Julia Rubin. “Fairness definitions explained”. In: *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE. 2018, pp. 1–7.

[849] Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. “Challenges and frontiers in abusive content detection”. In: *Proceedings of the Third Workshop on Abusive Language Online*. 2019, pp. 80–93.

[850] Bertie Vidgen and Taha Yasseri. “Detecting weak and strong Islamophobic hate speech on social media”. In: *Journal of Information Technology & Politics* 17.1 (2020), pp. 66–78.

[851] Luis Von Ahn. “Games with a purpose”. In: *Computer* 39.6 (2006), pp. 92–94.

[852] Luis Von Ahn and Laura Dabbish. “ESP: Labeling Images with a Computer Game.” In: *AAAI spring symposium: Knowledge collection from volunteer contributors*. Vol. 2. 2005.

[853] Luis Von Ahn, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. “Improving accessibility of the web with a computer game”. In: *SIGCHI*. 2006, pp. 79–82.

[854] Luis Von Ahn, M. Kedia, and M. Blum. “Verbosity: a game for collecting common-sense facts”. In: *SIGCHI*. 2006, pp. 75–78.

[855] Luis Von Ahn, Ruoran Liu, and Manuel Blum. “Peekaboom: a game for locating objects in images”. In: *SIGCHI*. 2006, pp. 55–64.

[856] Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, and Margaret Burnett. “From Gender Biases to Gender-Inclusive Design: An Empirical Investigation”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: ACM, 2019, 53:1–53:14. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300283. URL: http://doi.acm.org/10.1145/3290605.3300283.

[857] Jeremy Waldron. *The harm in hate speech*. Harvard University Press, 2012.

[858] Zhiyuan Wan, Xin Xia, David Lo, and Gail C Murphy. “How does machine learning change software development practices?” In: *Trans. on Software Engineering* 47.9 (2019), pp. 1857–1871.

[859] Angelina Wang, Alexander Liu, Ryan Zhang, Anat Kleiman, Leslie Kim, Dora Zhao, Iroha Shirai, Arvind Narayanan, and Olga Russakovsky. “REVISE: A tool for measuring and mitigating bias in visual datasets”. In: *International Journal of Computer Vision* (2022), pp. 1–21.

[860] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. “From human-human collaboration to Human-AI collaboration: Designing AI systems that can work together with people”. In: *Extended abstracts of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–6.

[861] Dakuo Wang, Q Vera Liao, Yunfeng Zhang, Udayan Khurana, Horst Samulowitz, Soya Park, Michael Muller, and Lisa Amini. “How much automation does a data scientist want?” In: *arXiv preprint arXiv:2101.03970* (2021).

[862] Dakuo Wang, Justin D Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. “Human-AI collaboration in data science: Exploring data scientists’ perceptions of automated AI”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–24.

[863] Jiakai Wang, Zixin Yin, Pengfei Hu, Aishan Liu, Renshuai Tao, Haotong Qin, Xianglong Liu, and Dacheng Tao. “Defensive Patches for Robust Recognition in the Physical World”. In: *CVPR*. 2022, pp. 2456–2465.

[864] Ruotong Wang, F Maxwell Harper, and Haiyi Zhu. “Factors influencing perceived fairness in algorithmic decision-making: Algorithm outcomes, development procedures, and individual differences”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–14.

[865] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. "Cursing in English on twitter". In: *Computer Supported Cooperative Work, CSCW '14, Baltimore, MD, USA, February 15-19, 2014*. Ed. by Susan R. Fussell, Wayne G. Lutters, Meredith Ringel Morris, and Madhu Reddy. ACM, 2014, pp. 415–425. ISBN: 978-1-4503-2540-0. DOI: 10.1145/2531602.2531734. URL: https://doi.org/10.1145/2531602.2531734.

[866] Yi Wang and David F. Redmiles. "Implicit gender biases in professional software development: an empirical study". In: *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society, ICSE 2019, Montreal, QC, Canada, May 25-31, 2019*. 2019, pp. 1–10. DOI: 10.1109/ICSE-SEIS.2019.00009. URL: https://doi.org/10.1109/ICSE-SEIS.2019.00009.

[867] Z. Waseem. "Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter". In: *Proceedings of the first workshop on NLP and computational social science*. 2016.

[868] Z. Waseem and D. Hovy. "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter." In: *SRW@ HLT-NAACL*. 2016, pp. 88–93.

[869] Thomas Way, Mary-Angela Papalaskari, Lillian Cassel, Paula Matuszek, Carol Weiss, and Yamini Praveena Tella. "Machine learning modules for all disciplines". In: *2017 acm Conf. on innovation and technology in computer science education*. 2017, pp. 84–85.

[870] Hilde Weerts, Lambèr Royakkers, and Mykola Pechenizkiy. "Does the End Justify the Means? On the Moral Justification of Fairness-Aware Machine Learning". In: *arXiv preprint arXiv:2202.08536* (2022).

[871] Lindsay Wells and Tomasz Bednarz. "Explainable ai and reinforcement learning—a systematic review of current approaches and trends". In: *Frontiers in artificial intelligence* 4 (2021), p. 550030.

[872] Chris Welty, Praveen Paritosh, and Lora Aroyo. "Metrology for AI: From Benchmarks to Instruments". In: *arXiv preprint arXiv:1911.01875* (2019).

[873] Mike Wendling. *2015: The year that angry won the internet*. https://www.bbc.com/news/blogs-trending-35111707, Last accessed on 2020-03-16. 2015.

[874] Jianshu Weng, Zhiqi Shen, Chunyan Miao, Angela Goh, and Cyril Leung. "Credibility: How agents can handle unfair third-party testimonies in computational trust models". In: *IEEE Transactions on Knowledge and Data Engineering* 22.9 (2009), pp. 1286–1298.

[875] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. "The what-if tool: Interactive probing of machine learning models". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 56–65.

[876] Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. "Inducing a Lexicon of Abusive Words–a Feature-Based Approach". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 1046–1056.

[877] A. Williams, C. Oliver, K. Aumer, and C. Meyers. "Racial microaggressions and perceptions of Internet memes". In: *Computers in Human Behavior* 63 (2016), pp. 424–432.

[878] Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. "Building and Auditing Fair Algorithms: A Case Study in Candidate Screening". In: *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*. Ed. by Madeleine Clare Elish, William Isaac, and Richard S. Zemel. ACM, 2021, pp. 666–677. DOI: 10.1145/3442188.3445928. URL: https://doi.org/10.1145/3442188.3445928.

[879] Michael J Witbrock, Cynthia Matuszek, Antoine Brusseau, Robert C Kahlert, C Bruce Fraser, and Douglas B Lenat. "Knowledge Begets Knowledge: Steps towards Assisted Knowledge Acquisition in Cyc." In: *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*. 2005, pp. 99–105.

[880] Michael Wojatzki, Tobias Horsmann, Darina Gold, and Torsten Zesch. "Do Women Perceive Hate Differently: Examining the Relationship Between Hate Speech, Gender, and Agreement Judgments". In: *Proceedings of the 14th Conference on Natural Language Processing, KONVENS 2018, Vienna, Austria, September 19-21, 2018*. Ed. by Adrien Barbaresi, Hanno Biber, Friedrich Neubarth, and Rainer Osswald. Österreichische Akademie der Wissenschaften, 2018, pp. 110–120. URL: https://www.oeaw.ac.at/fileadmin/subsites/academiaecorpora/PDF/konvens18%5C_13.pdf.

[881] Eric Wong and J. Zico Kolter. *Learning perturbation sets for robust machine learning*. 2020. DOI: 10.48550/ARXIV.2007.08450. URL: https://arxiv.org/abs/2007.08450.

[882] Richmond Y Wong, Michael A Madaio, and Nick Merrill. "Seeing Like a Toolkit: How Toolkits Envision the Work of AI Ethics". In: *arXiv preprint arXiv:2202.08792* (2022).

[883] Allison Woodruff, Sarah E. Fox, Steven Rousso-Schindler, and Jeffrey Warshaw. "A Qualitative Exploration of Perceptions of Algorithmic Fairness". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: ACM, 2018, 656:1–656:14. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3174230. URL: http://doi.acm.org/10.1145/3173574.3174230.

[884] Julie A Woodzicka, Robyn K Mallett, Shelbi Hendricks, and Astrid V Pruitt. "It's just a (sexist) joke: Comparing reactions to sexist versus racist communications". In: *Humor* 28.2 (2015), pp. 289–309.

[885] E. Wulczyn, N. Thain, and L. Dixon. "Ex machina: Personal attacks seen at scale". In: *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2017, pp. 1391–1399.

[886] Tomer Wullach, Amir Adler, and Einat Minkov. "Towards Hate Speech Detection at Large via Deep Generative Modeling". In: *arXiv preprint arXiv:2005.06370* (2020).

[887] Huichuan Xia, Yang Wang, Yun Huang, and Anuj Shah. ""Our Privacy Needs to be Protected at All Costs" Crowd Workers' Privacy Experiences on Amazon Mechanical Turk". In: *Proceedings of the ACM on human-computer interaction* 1.CSCW (2017), pp. 1–22.

[888] Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. "Demoting Racial Bias in Hate Speech Detection". In: *arXiv* (2020), arXiv–2005.

[889] Yao Xie, Melody Chen, David Kao, Ge Gao, and Xiang'Anthony' Chen. "CheXplain: Enabling Physicians to Explore and Understand Data-Driven, AI-Enabled Medical Imaging Analysis". In: *2020 CHI*. 2020, pp. 1–13.

[890] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. "Whither automl? understanding the role of automation in machine learning workflows". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–16.

[891] Pulei Xiong, Scott Buffett, Shahrear Iqbal, Philippe Lamontagne, Mohammad Mamun, and Heather Molyneaux. "Towards a robust and trustworthy machine learning system development: An engineering perspective". In: *Journal of Information Security and Applications* 65 (2022), p. 103121.

[892] Catherina Xu, Christina Greer, Manasi N Joshi, and Tulsee Doshi. "Fairness Indicators Demo: Scalable Infrastructure for Fair ML Systems". In: (2020).

[893] Depeng Xu, Shuhan Yuan, and Xintao Wu. "Achieving differential privacy and fairness in logistic regression". In: *Companion proceedings of The 2019 world wide web conference*. 2019, pp. 594–599.

[894] K Xu, J Ba, R Kiros, K Cho, A Courville, and al. "Show, attend and tell: Neural image caption generation with visual attention". In: *ICML*. 2015, pp. 2048–2057.

[895] Makoto Yamada, Leonid Sigal, and Michalis Raptis. "No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation". In: *European Conference on Computer Vision*. Springer. 2012, pp. 674–687.

[896] An Yan and Bill Howe. "Fairness in Practice: A Survey on Equity in Urban Mobility". In: *Data Engineering* (2019), p. 49.

[897] Y Yan, G M Fung, and al. "Active learning from crowds". In: *ICML*. 2011, pp. 1161–1168.

[898] C Yang, A Rangarajan, and S Ranka. "Global model interpretation via recursive partitioning". In: *IEEE HPCC/SmartCity/DSS*. IEEE. 2018, pp. 1563–1570.

[899] J Yang, A Smirnova, and al. "Scalpel-cd: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data". In: *WWW*. 2019, pp. 2158–2168.

[900] Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. "Leveraging crowdsourcing data for deep active learning an application: Learning intents in alexa". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 23–32.

[901] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. "Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* '20. Barcelona, Spain: Association for Computing Machinery, 2020, pp. 547–558. ISBN: 9781450369367. DOI: 10.1145/3351095.3375709. URL: https://doi.org/10.1145/3351095.3375709.

[902] Ke Yang, Biao Huang, Julia Stoyanovich, and Sebastian Schelter. "Fairness-aware instrumentation of preprocessing pipelines for machine learning". In: *HILDA workshop at SIGMOD*. 2020.

[903] Ke Yang and Julia Stoyanovich. "Measuring Fairness in Ranked Outputs". In: *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. SSDBM '17. Chicago, IL, USA: ACM, 2017, 22:1–22:6. ISBN: 978-1-4503-5282-6. DOI: 10.1145/3085504.3085526. URL: http://doi.acm.org/10.1145/3085504.3085526.

[904] M Yang and B Kim. "Benchmarking Attribution Methods with Relative Feature Importance". In: (2019).

[905] Pengfei Yang, J. Li, J. Liu, C-C. Huang, R. Li, L. Chen, X. Huang, and Lijun Zhang. "Enhancing Robustness Verification for Deep Neural Networks via Symbolic Propagation". In: *Formal Aspects of Computing* 33 (June 2021). DOI: 10.1007/s00165-021-00548-1.

[906] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. "Grounding interactive machine learning tool design in how non-experts actually build models". In: *2018 designing interactive systems Conf.* 2018, pp. 573–584.

[907] Adrienne Yapo and Joseph Weiss. "Ethical Implications of Bias in Machine Learning". In: *Proceedings of the 51st Hawaii International Conference on System Sciences*. 2018.

[908] I-Cheng Yeh. *default of credit card clients Data Set*. 2016. URL: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.

[909] Ming Yin, Siddharth Suri, and Mary L Gray. "Running out of time: The impact and value of flexibility in on-demand crowdwork". In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–11.

[910] S. Yoshida, J. Kitazono, S. Ozawa, T. Sugawara, T. Haga, and S. Nakamura. "Sentiment analysis for various SNS media using Naive Bayes classifier and its application to flaming detection". In: *2014 IEEE Symposium on Computational Intelligence in Big Data (CIBD)*. IEEE, Dec. 2014, pp. 1–6. ISBN: 978-1-4799-4540-5. DOI: 10.1109/CIBD.2014.7011523. URL: http://ieeexplore.ieee.org/document/7011523/.

[911] W. D. Yu, M. Gole, N. Prabhuswamy, S. Prakash, and V. G. Shankaramurthy. "An Approach to Design and Analyze the Framework for Preventing Cyberbullying". In: *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, June 2016, pp. 864–867. ISBN: 978-1-5090-2628-9. DOI: 10.1109/SCC.2016.125. URL: http://ieeexplore.ieee.org/document/7557547/.

[912] Shuhan Yuan, Xintao Wu, and Yang Xiang. "A Two Phase Deep Learning Model for Identifying Discrimination from Tweets". In: *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*. 2016, pp. 696–697. DOI: 10.5441/002/edbt.2016.92. URL: https://doi.org/10.5441/002/edbt.2016.92.

[913] Mireia Yurrita, Agathe Balayn, and Ujwal Gadiraju. "Generating Process-Centric Explanations to Enable Contestability in Algorithmic Decision Making: Challenges and Opportunities". In: *HCXAI workshop (CHI'23)*. 2023.

[914] Mireia Yurrita, Tim Draws, Agathe Balayn, Dave Murray-Rust, Nava Tintarev, and Alessandro Bozzon. "Disentangling Fairness Perceptions in Algorithmic Decision-Making: the Effects of Explanations, Human Oversight, and Contestability". In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2023, pp. 1–21.

[915] Mireia Yurrita, Dave Murray-Rust, Agathe Balayn, and Alessandro Bozzon. "Towards a multi-stakeholder value-based assessment framework for algorithmic systems". In: *CM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2022, pp. 535–563.

[916] Alexey Zagalsky, Dov Te'eni, Inbal Yahav, David G Schwartz, Gahl Silverman, Daniel Cohen, Yossi Mann, and Dafna Lewinsky. "The design of reciprocal learning between human and artificial intelligence". In: *ACM on Human-Computer Interaction* 5.CSCW2 (2021), pp. 1–36.

[917] Liang-Jun Zang, Cong Cao, Ya-Nan Cao, Yu-Ming Wu, and CAO Cun-Gen. “A survey of commonsense knowledge acquisition”. In: *Journal of Computer Science and Technology* 28.4 (2013), pp. 689–719.

[918] M D. Zeiler and R Fergus. “Visualizing and Understanding Convolutional Networks”. In: *ECCV*. 2014, pp. 818–833.

[919] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. *Adversarially Robust Generalization Just Requires More Unlabeled Data*. 2019. DOI: 10.48550/ARXIV.1906.00555. URL: https://arxiv.org/abs/1906.00555.

[920] Amy X Zhang, Michael Muller, and Dakuo Wang. “How do data science workers collaborate? roles, workflows, and tools”. In: *ACM on Human-Computer Interaction* 4.CSCW1 (2020), pp. 1–23.

[921] Chongzhi Zhang, Aishan Liu, Xianglong Liu, Yitao Xu, Hang Yu, Yuqing Ma, and Tianlin Li. “Interpreting and Improving Adversarial Robustness of Deep Neural Networks With Neuron Sensitivity”. In: *IEEE Trans. on Image Processing* 30 (2021), pp. 1291–1304. DOI: 10.1109/tip.2020.3042083. URL: https://doi.org/10.1109%2Ftip.2020.3042083.

[922] Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. “Demographics Should Not Be the Reason of Toxicity: Mitigating Discrimination in Text Classifications with Instance Weighting”. In: *arXiv preprint arXiv:2004.14088* (2020).

[923] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. “Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models”. In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 364–373.

[924] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. “Machine learning testing: Survey, landscapes and horizons”. In: *IEEE Transactions on Software Engineering* (2020).

[925] Jing Zhang, Xindong Wu, and Victor S Sheng. “Imbalanced multiple noisy labeling”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2014), pp. 489–503.

[926] Lu Zhang, Yongkai Wu, and Xintao Wu. “Achieving Non-Discrimination in Data Release”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. 2017, pp. 1335–1344. DOI: 10.1145/3097983.3098167. URL: https://doi.org/10.1145/3097983.3098167.

[927] Lu Zhang, Yongkai Wu, and Xintao Wu. “Causal modeling-based discrimination discovery and removal: criteria, bounds, and algorithms”. In: *IEEE Transactions on Knowledge and Data Engineering* (2018).

[928] Q Zhang and al. “Interpretable convolutional neural networks”. In: *CVPR*. 2018.

[929] Qiaoning Zhang, Matthew L Lee, and Scott Carter. “You Complete Me: Human-AI Teams and Complementary Expertise”. In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–28.

[930] Ru Zhang, Wencong Xiao, Hongyu Zhang, Yu Liu, Haoxiang Lin, and Mao Yang. “An empirical study on program failures of deep learning jobs”. In: *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE. 2020, pp. 1159–1170.

[931] Wencan Zhang and Brian Y Lim. “Towards Relatable Explainable AI with the Perceptual Process”. In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–24.

[932] Yunfeng Zhang, Rachel K.E. Bellamy, and Wendy A. Kellogg. “Designing Information for Remediating Cognitive Biases in Decision-Making”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Seoul, Republic of Korea: ACM, 2015, pp. 2211–2220. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702239. URL: http://doi.acm.org/10.1145/2702123.2702239.

[933] Z. Zhang, D. Robinson, and J. Tepper. “Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network”. In: *The Semantic Web*. Ed. by Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam. Cham: Springer International Publishing, 2018, pp. 745–760. ISBN: 978-3-319-93417-4.

[934] Zijian Zhang, Jaspreet Singh, Ujwal Gadiraju, and Avishek Anand. “Dissonance between human and machine understanding”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–23.

[935] Ziqi Zhang and Lei Luo. “Hate speech detection: A solved problem? the challenging case of long tail on twitter”. In: *Semantic Web* 10.5 (2019), pp. 925–945.

[936] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. “Gender Bias in Contextualized Word Embeddings”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 629–634.

[937] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. *Maximum-Entropy Adversarial Data Augmentation for Improved Generalization and Robustness*. 2020. DOI: 10.48550/ARXIV.2010.08001. URL: https://arxiv.org/abs/2010.08001.

[938] Peng Zhao, Yu-Jie Zhang, and Zhi-Hua Zhou. “Exploratory machine learning with unknown unknowns”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 10999–11006.

[939] R. Zhao and K. Mao. “Cyberbullying Detection Based on Semantic-Enhanced Marginalized Denoising Auto-Encoder”. In: *IEEE Transactions on Affective Computing* 8.3 (July 2017), pp. 328–339. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2016.2531682. URL: http://ieeexplore.ieee.org/document/7412690/.

[940] Qinkai Zheng, Xu Zou, Yuxiao Dong, Yukuo Cen, Da Yin, Jiarong Xu, Yang Yang, and Jie Tang. “Graph Robustness Benchmark: Benchmarking the Adversarial Robustness of Graph Machine Learning”. In: *NeurIPS*. 2021. URL: https://openreview.net/forum?id=NxWUnvwFV4.

[941] Yong Zheng, Tanaya Dave, Neha Mishra, and Harshit Kumar. “Fairness In Reciprocal Recommendations: A Speed-Dating Study”. In: *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. UMAP ’18. Singapore, Singapore: ACM, 2018, pp. 29–34. ISBN: 978-1-4503-5784-5. DOI: 10.1145/3213586.3226207. URL: http://doi.acm.org/10.1145/3213586.3226207.

[942] H. Zhong, H. Li, A. Squicciarini, S. Rajtmajer, C. Griffin, D. Miller, and C. Caragea. “Content-driven detection of cyberbullying on the instagram social network”. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press. 2016, pp. 3952–3958.

[943] Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. “Evaluating the quality of machine learning explanations: A survey on methods and metrics”. In: *Electronics* 10.5 (2021), p. 593.

[944] Minhaz Fahim Zibran. “Chi-squared test of independence”. In: (2007).

[945] Thomas Zielke. “Is Artificial Intelligence Ready for Standardization?” In: *European Conference on Software Process Improvement*. Springer. 2020, pp. 259–274.

[946] Indre Zliobaite. “Measuring discrimination in algorithmic decision making”. In: *Data Min. Knowl. Discov.* 31.4 (2017), pp. 1060–1089. DOI: 10.1007/s10618-017-0506-1. URL: https://doi.org/10.1007/s10618-017-0506-1.

[947] D-S. Zois, A. Kapodistria, M. Yao, and C. Chelmis. “Optimal Online Cyberbullying Detection”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2018. ISBN: 978-1-5386-4658-8. DOI: 10.1109/ICASSP.2018.8462092. URL: https://ieeexplore.ieee.org/document/8462092/.

[948] Kathryn Zyskowski, Meredith Ringel Morris, Jeffrey P. Bigham, Mary L. Gray, and Shaun K. Kane. “Accessible Crowdwork?: Understanding the Value in and Challenge of Microtask Employment for People with Disabilities”. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW 2015, Vancouver, BC, Canada, March 14 - 18, 2015*. Ed. by Dan Cosley, Andrea Forte, Luigina Ciolfi, and David McDonald. ACM, 2015, pp. 1682–1693. DOI: 10.1145/2675133.2675158. URL: https://doi.org/10.1145/2675133.2675158.

APPENDIX

SUMMARY

Machine learning (ML) is an artificial intelligence technology that has a great potential for being adopted in various sectors of activities. Yet, it is now also increasingly recognized as a hazardous technology. Failures in the outputs of an ML system might cause physical or social harms. Besides, the development and deployment of an ML system itself are also argued to be harmful in certain contexts.

Surprisingly, these hazards persist in applications where ML technology has been deployed, despite the increasing amount of research performed by the ML research community. In this thesis, we task ourselves with the challenges of understanding the reasons for the subsistence of hazardous system's output failures and of hazardous development and deployment processes in practice, and of developing solutions to further diagnose these hazardous failures (especially in the system's outputs). For that, we investigate further the nature of the potential gap between research and the practices of those developers who build and deploy the systems. To do so, we survey major related ML research directions, surface developers practices and challenges, and search for types of (mis)alignement between theory and practices. There, among others, we find a lack of technical support for ML developers to identify the potential failures of their systems. Hence, we then tackle the development and evaluation of a human-in-the-loop, explainability-based, failure diagnosis method and user-interface for computer vision systems.

In terms of current ML research directions in relation to hazardous failures, we find that these directions revolve around: the characterization of harms caused by ML systems and their causes within the ML development lifecycle; the development of technical solutions for measuring and increasing the robustness of ML systems and algorithmic fairness in the outputs of ML systems (sometimes research on fairness and robustness is intertwined); and the critical characterization of proposed solutions in terms of the harms they might leave out or inaccurately reflect. We also find a great lack of understanding of developers practices in handling harms of their ML systems and in using proposed technical solutions around robustness and algorithmic fairness.

In order to characterize the research / practice gap, we then investigate practices and compare the above research directions and research insights to the challenges faced by developers. We find three types of gaps. *Technical gap:* The current technical research directions can be used to support certain needs of developers, but these developers also have additional needs that are not yet answered by any existing (technical or not) solution. *Social gap:* In comparison to existing research literature, developers might lack awareness on certain types of hazardous failures and on available ways to handle them, they sometimes mis-interpret and mis-use available tools to support the failure handling process, and they do not necessarily reflect on the limitations of existing technical solutions and tools and of their own conceptions and approaches of the failures. Note however that there is no best way to envision and handle hazardous failures as it is a

subjective concept, for which no clear solution exists. *Organisational gap:* Finally, developers might also face a number of contextual, organisation-related obstacles, that might hinder them in handling harms.

These three types of gaps call for diverse solutions that we briefly discuss: the development of technical methods; the design of developer-friendly tools that rely on the technical methods; the creation of education process for the new tasks that developers face in the context of ML and harms; and the proposition of new policies to further regulate the development and deployment of ML systems.

Finally, we tackle one of the technical solutions that revealed important to conduct a rich model diagnosis based on a model's learned features. We design a new explainability method that could support developers in further diagnosing hazardous failures of their ML systems by identifying human-interpretable features learned by the model; we develop a game with a purpose to collect knowledge potentially needed for the diagnosis process and especially for assessing the soundness of the model's learned features; and we build a developer-friendly user-interface that gathers the outputs of the explainability method (model learned features) and of the game (domain information for assessing model features) and prior explainability methods for supporting developers in their diagnosis process. We then evaluate these two methods and this artifact. We find that the involvement of crowd workers and their knowledge is useful to collect rich information useful for diagnosing hazardous failures. We also show that with this information, we can support developers in identifying a more comprehensive set of hazardous failures.

This work paves the way for a breadth of technical, design, social, organisational, and policy efforts to bridge the research / practice gap with the aim of developing less hazardous ML systems. This work also constitutes one of the first in-depth reports of an instance of a full-circle, mixed-method, research work in the context of ML hazardous-failure diagnosis, on which we extract lessons learned and recommendations for future research efforts.

SAMENVATTING

Machine learning (ML) is een technologie voor kunstmatige intelligentie die een groot potentieel heeft om te worden toegepast in verschillende sectoren. Toch wordt ML nu ook steeds meer erkend als een gevaarlijke technologie. Fouten in de output van een ML-systeem kunnen fysieke of sociale schade veroorzaken. Daarnaast wordt ook gesteld dat de ontwikkeling en inzet van een ML-systeem zelf schadelijk kan zijn in bepaalde contexten.

Verrassend genoeg blijven deze gevaren bestaan in toepassingen waar ML-technologie is ingezet, ondanks de toenemende hoeveelheid onderzoek uitgevoerd door de ML-onderzoeksgemeenschap. In dit proefschrift stellen we onszelf voor de uitdaging om de redenen te begrijpen voor het voortbestaan van gevaarlijke systeemuitvoerfouten en van gevaarlijke ontwikkel- en implementatieprocessen in de praktijk, en om oplossingen te ontwikkelen om deze gevaarlijke fouten verder te diagnosticeren (vooral in de uitvoer van het systeem). Daarvoor onderzoeken we de aard van de potentiële kloof tussen onderzoek en de praktijk van de ontwikkelaars die de systemen bouwen en inzetten. Om dit te doen, inventariseren we de belangrijkste gerelateerde ML-onderzoeksrichtingen, brengen we ontwikkelaarspraktijken en -uitdagingen in kaart, en zoeken we naar soorten (mis)afstemming tussen theorie en praktijk. Daarbij zien we onder andere een gebrek aan technische ondersteuning voor ML-ontwikkelaars om de potentiële fouten van hun systemen te identificeren. Daarom gaan we vervolgens in op de ontwikkeling en evaluatie van een op verklaarbaarheid gebaseerde human-in-the-loop foutdiagnosemethode en gebruikersinterface voor computervisiesystemen.

In termen van huidige ML onderzoeksrichtingen in relatie tot gevaarlijke fouten, vinden we dat deze richtingen draaien om de volgende aspecten: de karakterisering van schade veroorzaakt door ML systemen en hun oorzaken binnen de ML ontwikkelingslevenscyclus; de ontwikkeling van technische oplossingen voor het meten en verhogen van de robuustheid van ML systemen en algoritmische eerlijkheid in de output van ML systemen (soms is onderzoek naar eerlijkheid en robuustheid met elkaar verweven); en de kritische karakterisering van voorgestelde oplossingen in termen van de schade die ze mogelijk weglaten of onnauwkeurig weergeven. We vinden ook een groot gebrek aan inzicht in de praktijk van ontwikkelaars in het omgaan met schade van hun ML-systemen en in het gebruik van voorgestelde technische oplossingen rond robuustheid en algoritmische eerlijkheid.

Om de kloof tussen onderzoek en praktijk te karakteriseren, onderzoeken we vervolgens de praktijk en vergelijken we de bovenstaande onderzoeksrichtingen en onderzoeksinzichten met de uitdagingen waarmee ontwikkelaars worden geconfronteerd. We vinden drie soorten hiaten. Technische kloof: De huidige technische onderzoeksrichtingen kunnen worden gebruikt om bepaalde behoeften van ontwikkelaars te ondersteunen, maar deze ontwikkelaars hebben ook aanvullende behoeften die nog niet worden beantwoord door een bestaande (al dan niet technische) oplossing. Sociale kloof: In ver-

gelijking met de bestaande onderzoeksliteratuur zijn ontwikkelaars zich misschien niet voldoende bewust van bepaalde soorten gevaarlijke storingen en de beschikbare manieren om ermee om te gaan, ze interpreteren en gebruiken de beschikbare hulpmiddelen ter ondersteuning van het proces van storingsbehandeling soms verkeerd, en ze denken niet noodzakelijk na over de beperkingen van de bestaande technische oplossingen en hulpmiddelen en van hun eigen opvattingen over en benaderingen van storingen. Ook denken ze niet noodzakelijk na over de beperkingen van bestaande technische oplossingen en hulpmiddelen en van hun eigen opvattingen over en benaderingen van storingen. Merk echter op dat er geen beste manier is om gevaarlijke storingen in te schatten en te behandelen, aangezien het een subjectief concept is waarvoor geen duidelijke oplossing bestaat. Organisatorische kloof: Tot slot kunnen ontwikkelaars ook te maken krijgen met een aantal contextuele, organisatiegerelateerde obstakels die hen kunnen hinderen bij het omgaan met schade.

Deze drie soorten hiaten vragen om verschillende oplossingen die we kort bespreken: de ontwikkeling van technische methoden; het ontwerp van ontwikkelaarvriendelijke gereedschappen die vertrouwen op de technische methoden; het creëren van een onderwijsproces voor de nieuwe taken waarmee ontwikkelaars te maken krijgen in de context van ML en schade; en het voorstellen van nieuw beleid om de ontwikkeling en inzet van ML-systemen verder te reguleren.

Tot slot pakken we een van de technische oplossingen aan die belangrijk zijn gebleken om een uitgebreide modeldiagnose uit te voeren op basis van de geleerde kenmerken van een model. We ontwerpen een nieuwe verklaarbaarheidsmethode die ontwikkelaars kan ondersteunen bij het verder diagnosticeren van gevaarlijke fouten in hun ML-systemen door het identificeren van door mensen interpreteerbare eigenschappen die door het model zijn geleerd; we ontwikkelen een spel met als doel kennis te verzamelen die mogelijk nodig is voor het diagnoseproces en in het bijzonder voor het beoordelen van de deugdelijkheid van de door het model geleerde eigenschappen; en we bouwen een ontwikkelaarsvriendelijke gebruikersinterface die de output van de verklaarbaarheidsmethode (door het model geleerde eigenschappen) en van het spel (domeininformatie voor het beoordelen van modeleigenschappen) en eerdere verklaarbaarheidsmethoden verzamelt om ontwikkelaars te ondersteunen bij hun diagnoseproces. Vervolgens evalueren we deze twee methoden en dit artefact. We vinden dat de betrokkenheid van crowdworkers en hun kennis nuttig is om rijke informatie te verzamelen die nuttig is voor het diagnosticeren van gevaarlijke fouten. We laten ook zien dat we met deze informatie ontwikkelaars kunnen ondersteunen bij het identificeren van een uitgebreidere set van gevaarlijke fouten.

Dit werk maakt de weg vrij voor een breed scala aan technische, ontwerp-, sociale, organisatorische en beleidsinspanningen om de kloof tussen onderzoek en praktijk te overbruggen met als doel minder gevaarlijke ML-systemen te ontwikkelen. Dit werk is ook een van de eerste diepgaande verslagen van een volledig, gemengd methodologisch onderzoek in de context van ML foutdiagnose, waaruit we lessen trekken en aanbevelingen doen voor toekomstige onderzoeksinspanningen.

ACKNOWLEDGEMENTS

This extensive document and the corresponding publications might come across as the culmination of a short, four-year, solitary, PhD trajectory. Yet, this document represents far more than that: this body of work holds a significance far beyond its material and temporal confines. The outcomes of my PhD trajectory are manifold. Of course, I have evolved into an *independent* researcher with all the disputable connotations attached to the term. Along the way, I made many, new, exciting connections. However, these accomplishments extend beyond the academic realm. I have also developed a new understanding of the world illustrated by the thesis cover, a burgeoning set of interests that extend far beyond research, and personal transformations of my character, beliefs, etc. Importantly, these outcomes are not the result of my sole individual efforts during these four assiduous years of work. Rather, they are the culmination of a much lengthier process, during which I benefited immensely from the assistance, support, and guidance of numerous individuals. Thanking all these individuals would be too big of an endeavor, that would risk an encyclopedic stretch of this document. Let me just thank some of them here, to start expressing my gratitude.

First and foremost, I want to thank the enablers of my PhD at TU Delft. Among them, I wish to offer special recognition to Alessandro Bozzon and Geert-Jan Houben. They offered me the opportunity to embark on the PhD journey, regularly followed my progress, and sought various support and resources so as to successfully conclude various research projects. Our, many, demanding, discussions played a pivotal role in my personal and professional growth during this journey. I would also like to express my appreciation to Jie Yang and Ujwal Gadiraju. Our countless discussions and common efforts to design studies, formulate research questions, develop experimental setups, and lay on paper all our exciting findings have greatly contributed to the successful realization of my thesis. Their contribution extends beyond the thesis, and I will keep great memories of all the fun moments we shared together. I reserve a special thank you to Seda Gürses. I met her at the end of my first PhD year, and her support has been unwavering since then. With her many inputs, I have learned to reflect critically on my research, and to adopt new, exploratory, and rigorous research methodologies. Her insightful mentorship has extended beyond research projects. She has provided me with plenty of opportunities to grow outside the constraints of a scientific publication, which exposed me to another side of academic research and to other avenues to make a positive impact on the world. I must not forget to thank the various TU Delft support teams for facilitating all the administrative aspects of research within academia. I am particularly grateful to Daphne Stephan for skillfully juggling all the emails I sent her and all the calendar invites she had to share with me. Finally, I am thankful to all my PhD committee members, Prof. Maaike Kleinsmann, Prof. Irina Shklovski, Prof. Somaya Ben Allouch, Prof. Philippe Cudré-Mauroux, and Prof. Gerd Kortuem, for accepting the daunting task of reviewing my lengthy thesis.

Another wave of appreciation goes towards all those individuals with whom I interacted directly or anonymously, online or in-person, once or frequently, during the PhD, to successfully conduct research projects. Especially, I am grateful to all the talented students at TU Delft. Our interactions taught me about myself and about our research. I am thinking about the students who attended the few lectures I gave and projects I TA-ed, but also about those with whom we worked on bachelor and master theses and other projects, such as Panagiotis Soilis, Ziad Nawar, Simran Karnani, Shreyan Biswas, Niels de Bruin, Natasa Rikalo, Andy Hu, Manisha Sethia, and many more. In addition to students, I want to bring attention to all the crowd-workers who took part in the quantitative experiments I conducted. I also appreciate the contributions of the 90+ machine learning practitioners who generously devoted their time to answer my floods of questions about their practices. A big thank you also comes to all the anonymous reviewers of my conference submission, whose feedback was invaluable in refining my work.

My acknowledgments extend to my colleagues and friends, who greatly contributed to making my PhD experience enjoyable, even though they might not have been involved directly in the research. Thank you to the WIS group: Avishek, Asterios, Christoph, Claudia, Rihan, Sole, Andrea, David, Jurek, Lixia, Marios, Mesut, Oana, Pavel, Tahir, Venky, Alisa, Andra, Arthur, Christos, Danning, Esra, Felipe, Gaole, Garrett, Georgios, Gustavo, Petros, Kyriakos, Lijun, Lorenzo, Manuel, Nirmal, Peide, Philip, Sara, Sepideh, Sihang, Shabnam, Shahin, Tim, and Ziyu. Thank you to my other co-authors, colleagues, and friends, including Burcu, Corinne, Donald, Luca, Machiel, Thijmen, and Zoltan. Special thanks to my paranymphs, Jiwon and Mireia, for taking the role of mental defenders on the final graduation day, but also for the tons of fun and less fun chats we've had all along my PhD. Thank you to David, who acted as a great sparing partner and Photoshop expert in ideating and finalizing my thesis cover. Thank you as well to those that I met during academic events, who supported me in my various endeavors: Charu, Cynthia, Daphne, Evgeny, Frederik, Gunay, Hilde, Himanshu, Jill, Laura, Leon, Marvin, Pawel, Piotr, Roel, Sarah (and EDRi), Siddharth, Sonja, Stefan, Wiebke, and many more.

I want to thank my friends in the Netherlands and abroad. Our shared experiences after work and during holidays have put a cherry on the PhD cake, be it at bouldering, badminton, or padel, for cycling, padling, or kayaking, on trips, during art or music sessions, at restaurants and cinemas, to cite just a few activities. Thank you Gyoung, Guoxin, Jiwon, Nianlei, Nilay, Priyanka, Stan, Tomo, Victoire, Weichen, Xiangwei, Xuehan, Yikai, Yinan, Yu Ying, Zina, and many more friends! Thank you Camille and Filippo for the bread and house invitations! Thank you to my dear Greek trio with whom we shared countless COVID slow moments, but also original adventures together, and who constantly brought me mental and physical support in this long process. Thank you to the exceptional ENSTA team –Aiky, Claire, Clementine, Damien, David, Felix, Heloise, Hugo, Lea, Marc–, for these amazing summer breaks and winter interruptions, traveling kilometers for a few, fun, days (despite the tarot games...). Finally, a profound thank you to my family, including my aunt and uncle, grandparents and parents, cousins and siblings, who have supported me in many ways since I was born. Their encouragement –even though my research might seem funny at times–, advice, education, culinary support, healthcare aid, artistic influences, and traveling opportunities have all contributed to my PhD. A special acknowledgement to Patricia, Philippe, Joséphine, et Marius, merci!

CURRICULUM VITÆ

Agathe BALAYN

Agathe Balayn was born in Paris, France, on January 8, 1995. She holds a master degree in Computer Science from Delft University of Technology (2016-2018), and a diplôme d'ingénieur in Systems and Control from ENSTA ParisTech Institut Polytechnique de Paris (2014-2018). During her studies, she interned in various research labs, both in industry (Honda Research Institute, Japan; IBM Center for Advanced Studies, the Netherlands) and in academia (Research Institute for Cognition and Robotics, Bielefeld University, Germany).

From April 2019 to April 2023, Agathe Balayn was a PhD candidate in the Web Information Systems group at Delft University of Technology. Her PhD work focused on uncovering, understanding, and mitigating some of the hazards and harms that the deployment of machine learning (ML) systems into society can raise, using a mixed-method approach. She especially investigated the gap between ML research and ML practices, and developed solutions for this gap, going from literature surveys and policies for ML technologies to supportive tools for ML practitioners. Agathe's research has been published in leading conferences and journals (e.g., CHI, WWW, FAccT, VLDBJ, HCOMP, IUI, etc.). Her research has received various recognitions, such as best paper awards at HCOMP'22, CHI'23, and AIES'23, a best demo award at HCOMP'21, and a nomination for best paper award at WWW'22, as well as an honorable mention at the For Women in Science Rising Talent Competition (made possible by L'Oreal, UNESCO and the Royal Holland Society of Sciences and Humanities). Next to presenting her work at international conferences, she has given invited presentations at various local events, workshops, symposiums, and summer schools, and participated to several panel discussions online and in-person. She has also served as a program committee member and reviewer for several conferences and journals, such as CHI, CSCW, WWW, HCOMP, IUI, AAAI, SIGIR, NeurIPs. Next to these academic services, she has worked as a technical and policy consultant for the European Digital Rights (EDRi) organisation, she has also provided technical knowledge to several tech journalists, e.g., at AlgorithmWatch, and she has been a member of the Slow Reading research group gathering artists reflecting on AI and discrimination, organized by the Rotterdam Arts and Sciences Lab and by the AI4FUTURE program cofunded by the Creative Europe programme.

List of Publications

Conference and journal publications

1. Agathe Balayn, Mireia Yurrita, Jie Yang, and Ujwal Gadiraju. ""☑ Fairness Toolkits, A Checkbox Culture?" On the Factors that Fragment Developer Practices in Handling Algorithmic Harms". In: *ACM Conference on AI, Ethics, and Society (AIES)*. 2023 🏆

2. Nirmal Roy, Agathe Balayn, David Maxwell, and Claudia Hauff. "Hear Me Out: A Study on the Use of the Voice Modality for Crowdsourced Relevance Assessments". In: *Special Interest Group on Information Retrieval (SIGIR)*. 2023

3. Luca Nannini, Agathe Balayn, and Adam Leon Smith. "Explainability in AI Policies: A Critical Review of Communications, Reports, Regulations, and Standards in the EU, US, and UK". in: *2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2023

4. Shahin Sharifi, Sihang Qiu, Burcu Sayin, Agathe Balayn, Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. "Perspective, Leveraging Human Understanding for Identifying and Characterizing Image Atypicality". In: *Conference on Intelligent User Interface (IUI)*. 2023

5. Mireia Yurrita, Tim Draws, Agathe Balayn, Dave Murray-Rust, Nava Tintarev, and Alessandro Bozzon. "Disentangling Fairness Perceptions in Algorithmic Decision-Making: the Effects of Explanations, Human Oversight, and Contestability". In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2023, pp. 1–21 🏆

6. Agathe Balayn, Natasa Rikalo, Jie Yang, and Alessandro Bozzon. "Faulty or Ready? Handling Failures in Deep-Learning Computer Vision Models until Deployment: A Study of Practices, Challenges, and Needs". In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2023, pp. 1–20

7. Gaole He, Agathe Balayn, Stefan Buijsman, Jie Yang, and Ujwal Gadiraju. "It Is Like Finding a Polar Bear in the Savannah! Concept-Level AI Explanations with Analogical Inference from Commonsense Knowledge". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. vol. 10. 1. 2022, pp. 89–101 🏆

8. Mireia Yurrita, Dave Murray-Rust, Agathe Balayn, and Alessandro Bozzon. "Towards a multi-stakeholder value-based assessment framework for algorithmic systems". In: *CM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2022, pp. 535–563

9. Agathe Balayn, Natasa Rikalo, Christoph Lofi, Jie Yang, and Alessandro Bozzon. "How can Explainability Methods be Used to Support Bug Identification in Computer Vision Models?" In: *CHI Conference on Human Factors in Computing Systems (CHI)*. 2022, pp. 1–16

10. Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. "Ready Player One! Eliciting Diverse Knowledge Using A Configurable Game". In: *the Web Conference (WWW)*. 2022, pp. 1709–1719

11. Agathe Balayn, Jie Yang, Zoltan Szlavik, and Alessandro Bozzon. "Automatic Identification of Harmful, Aggressive, Abusive, and Offensive Language on the Web: A Survey of Technical Biases Informed by Psychology Literature". In: *ACM Transactions on Social Computing (TSC)* 4.3 (2021), pp. 1–56

12. Agathe Balayn, Christoph Lofi, and Geert-Jan Houben. "Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems". In: *The VLDB Journal (VLDBJ)* 30.5 (2021), pp. 739–768

13. Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, and Alessandro Bozzon. "What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis". In: *Proceedings of the Web Conference 2021 (WWW)*. 2021, pp. 1937–1948

14. *(Under review)* Agathe Balayn, Ujwal Gadiraju, and Jie Yang. ""Accuracy-fairness trade-off, let's cut the burrito in half"? On the Conceptions and Practices of ML Developers towards Algorithmic Fairness and Harms". In: 2023

15. *(Under review)* Andrea Tocchetti, Lorenzo Corti, Agathe Balayn, Mireia Yurrita, Philip Lippmann, Marco Brambilla, and Jie Yang. "AI Robustness: a Human-Centered Perspective on Technological Challenges and Opportunities". In: *arXiv preprint arXiv:2210.08906* (2022)

Reports, workshop papers, demonstration papers

1. Mireia Yurrita, Agathe Balayn, and Ujwal Gadiraju. "Generating Process-Centric Explanations to Enable Contestability in Algorithmic Decision Making: Challenges and Opportunities". In: *HCXAI workshop (CHI'23)*. 2023

2. Agathe Balayn and Seda Gürses. "Beyond Debiasing: Regulating AI and its inequalities". In: *Report for the European Digital Rights organisation (EDRi). https://edri. org/wp-content/uploads/2021/09/EDRi_Beyond-Debiasing-Report_Online. pdf* (2021)

3. Agathe Balayn, Gaole He, Andrea Hu, Jie Yang, and Ujwal Gadiraju. "Finditout: A multiplayer gwap for collecting plural knowledge". In: *Vol. 9 (2021): Proceedings of the Ninth AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 2021 🏆

4. Agathe Balayn, Bogdan Kulynych, and Seda Guerses. "Exploring Data Pipelines through the Process Lens: a Reference Model for Computer Vision". In: *Beyond Fairness workshop (CVPR'21)* (2021)

5. Agathe Balayn and Alessandro Bozzon. "Designing evaluations of machine learning models for subjective inference: the case of sentence toxicity". In: *Rigorous Evaluation of Machine Learning workshop (HCOMP'19)* (2019)

6. Agathe Balayn, Alessandro Bozzon, and Zoltan Szlavik. "Unfairness towards subjective opinions in Machine Learning". In: *Human-Centered Machine Learning workshop (CHI'19)* (2019)

SIKS Dissertation Series

Since 1998, all dissertations written by PhD. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series (following are all the dissertations since 2016).

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization
15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility
21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations
02 Felix Mannhardt (TUE), Multi-perspective Process Mining
03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araújo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses
28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TUE), On local and global structure mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components
08 Ali Mirsoleimani (UL),Structured Parallel Programming for Monte Carlo Tree Search
09 Myriam Traub (UU), Measuring Tool Bias and Improving Data Quality for Digital Humanities Research
10 Alifah Syamsiyah (TUE), In-database Preprocessing for Process Mining
11 Sepideh Mesbah (TUD), Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models
12 Ward van Breda (VU), Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment
13 Marco Virgolin (CWI), Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming
14 Mark Raasveldt (CWI/UL), Integrating Analytics with Relational Databases
15 Konstantinos Georgiadis (OUN), Smart CAT: Machine Learning for Configurable Assessments in Serious Games
16 Ilona Wilmont (RUN), Cognitive Aspects of Conceptual Modelling
17 Daniele Di Mitri (OUN), The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences
18 Georgios Methenitis (TUD), Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems
19 Guido van Capelleveen (UT), Industrial Symbiosis Recommender Systems
20 Albert Hankel (VU), Embedding Green ICT Maturity in Organisations
21 Karine da Silva Miras de Araujo (VU), Where is the robot?: Life as it could be
22 Maryam Masoud Khamis (RUN), Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar
23 Rianne Conijn (UT), The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging
24 Lenin da Nóbrega Medeiros (VUA/RUN), How are you feeling, human? Towards emotionally supportive chatbots
25 Xin Du (TUE), The Uncertainty in Exceptional Model Mining
26 Krzysztof Leszek Sadowski (UU), GAMBIT: Genetic Algorithm for Model-Based mixed-Integer opTimization
27 Ekaterina Muravyeva (TUD), Personal data and informed consent in an educational context
28 Bibeg Limbu (TUD), Multimodal interaction for deliberate practice: Training complex skills with augmented reality
29 Ioan Gabriel Bucur (RUN), Being Bayesian about Causal Inference
30 Bob Zadok Blok (UL), Creatief, Creatiever, Creatiefst
31 Gongjin Lan (VU), Learning better – From Baby to Better
32 Jason Rhuggenaath (TUE), Revenue management in online markets: pricing and online advertising
33 Rick Gilsing (TUE), Supporting service-dominant business model evaluation in the context of business model innovation
34 Anna Bon (MU), Intervention or Collaboration? Redesigning Information and Communication Technologies for Development
35 Siamak Farshidi (UU), Multi-Criteria Decision-Making in Software Production

2021 01 Francisco Xavier Dos Santos Fonseca (TUD),Location-based Games for Social Interaction in Public Space

02 Rijk Mercuur (TUD), Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models
03 Seyyed Hadi Hashemi (UVA), Modeling Users Interacting with Smart Devices
04 Ioana Jivet (OU), The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning
05 Davide Dell'Anna (UU), Data-Driven Supervision of Autonomous Systems
06 Daniel Davison (UT), "Hey robot, what do you think?" How children learn with a social robot
07 Armel Lefebvre (UU), Research data management for open science
08 Nardie Fanchamps (OU), The Influence of Sense-Reason-Act Programming on Computational Thinking
09 Cristina Zaga (UT), The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Children's Collaboration Through Play
10 Quinten Meertens (UvA), Misclassification Bias in Statistical Learning
11 Anne van Rossum (UL), Nonparametric Bayesian Methods in Robotic Vision
12 Lei Pi (UL), External Knowledge Absorption in Chinese SMEs
13 Bob R. Schadenberg (UT), Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning
14 Negin Samaeemofrad (UL), Business Incubators: The Impact of Their Support
15 Onat Ege Adali (TU/e), Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm
16 Esam A. H. Ghaleb (UM), Bimodal emotion recognition from audio-visual cues
17 Dario Dotti (UM), Human Behavior Understanding from motion and bodily cues using deep neural networks
18 Remi Wieten (UU), Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks
19 Roberto Verdecchia (VU), Architectural Technical Debt: Identification and Management
20 Masoud Mansoury (TU/e), Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems
21 Pedro Thiago Timbó Holanda (CWI), Progressive Indexes
22 Sihang Qiu (TUD), Conversational Crowdsourcing
23 Hugo Manuel Proença (LIACS), Robust rules for prediction and description
24 Kaijie Zhu (TUE), On Efficient Temporal Subgraph Query Processing
25 Eoin Martino Grua (VUA), The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications
26 Benno Kruit (CWI & VUA), Reading the Grid: Extending Knowledge Bases from Human-readable Tables
27 Jelte van Waterschoot (UT), Personalized and Personal Conversations: Designing Agents Who Want to Connect With You
28 Christoph Selig (UL), Understanding the Heterogeneity of Corporate Entrepreneurship Programs

2022 01 Judith van Stegeren (UT), Flavor text generation for role-playing video games
02 Paulo da Costa (TU/e), Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey
03 Ali el Hassouni (VUA), A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare
04 Ünal Aksu (UU), A Cross-Organizational Process Mining Framework
05 Shiwei Liu (TU/e), Sparse Neural Network Training with In-Time Over-Parameterization
06 Reza Refaei Afshar (TU/e), Machine Learning for Ad Publishers in Real Time Bidding
07 Sambit Praharaj (OU), Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics
08 Maikel L. van Eck (TU/e), Process Mining for Smart Product Design
09 Oana Andreea Inel (VUA), Understanding Events: A Diversity-driven Human-Machine Approach
10 Felipe Moraes Gomes (TUD), Examining the Effectiveness of Collaborative Search Engines
11 Mirjam de Haas (UT), Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers' engagement with robots and tasks during second-language tutoring
12 Guanyi Chen (UU), Computational Generation of Chinese Noun Phrases
13 Xander Wilcke (VUA), Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented Knowledge
14 Michiel Overeem (UU), Evolution of Low-Code Platforms
15 Jelmer Jan Koorn (UU), Work in Process: Unearthing Meaning using Process Mining
16 Pieter Gijsbers (TU/e), Systems for AutoML Research
17 Laura van der Lubbe (VUA), Empowering vulnerable people with serious games and gamification
18 Paris Mavromoustakos Blom (TiU), Player Affect Modelling and Video Game Personalisation
19 Bilge Yigit Ozkan (UU), Cybersecurity Maturity Assessment and Standardisation
20 Fakhra Jabeen (VUA), Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media
21 Seethu Mariyam Christopher (UM), Intelligent Toys for Physical and Cognitive Assessments
22 Alexandra Sierra Rativa (TiU), Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations
23 Ilir Kola (TUD), Enabling Social Situation Awareness in Support Agents
24 Samaneh Heidari (UU), Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

25 Anna L.D. Latour (LU), Optimal decision-making under constraints and uncertainty
26 Anne Dirkson (LU), Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences
27 Christos Athanasiadis (UM), Emotion-aware cross-modal domain adaptation in video sequences
28 Onuralp Ulusoy (UU), Privacy in Collaborative Systems
29 Jan Kolkmeier (UT), From Head Transform to Mind Transplant: Social Interactions in Mixed Reality
30 Dean De Leo (CWI), Analysis of Dynamic Graphs on Sparse Arrays
31 Konstantinos Traganos (TU/e), Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management
32 Cezara Pastrav (UU), Social simulation for socio-ecological systems
33 Brinn Hekkelman (CWI/TUD), Fair Mechanisms for Smart Grid Congestion Management
34 Nimat Ullah (VUA), Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change
35 Mike E.U. Ligthart (VUA), Shaping the Child-Robot Relationship: Interaction Design Patterns for a Sustainable Interaction

2023 01 Bojan Simoski (VUA), Untangling the Puzzle of Digital Health Interventions
02 Mariana Rachel Dias da Silva (TiU), Grounded or in flight? What our bodies can tell us about the whereabouts of our thoughts
03 Shabnam Najafian (TUD), User Modeling for Privacy-preserving Explanations in Group Recommendations
04 Gineke Wiggers (UL), The Relevance of Impact: bibliometric-enhanced legal information retrieval
05 Anton Bouter (CWI), Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization, Including Real-World Medical Applications
06 António Pereira Barata (UL), Reliable and Fair Machine Learning for Risk Assessment
07 Tianjin Huang (TU/e), The Roles of Adversarial Examples on Trustworthiness of Deep Learning
08 Lu Yin (TU/e), Knowledge Elicitation using Psychometric Learning
09 Xu Wang (VUA), Scientific Dataset Recommendation with Semantic Techniques
10 Dennis J.N.J. Soemers (UM), Learning State-Action Features for General Game Playing
11 Fawad Taj (VUA), Towards Motivating Machines: Computational Modeling of the Mechanism of Actions for Effective Digital Health Behavior Change Applications
12 Tessel Bogaard (VUA), Using Metadata to Understand Search Behavior in Digital Libraries
13 Injy Sarhan (UU), Open Information Extraction for Knowledge Representation
14 Selma Čaušević (TUD), Energy resilience through self-organization
15 Alvaro Henrique Chaim Correia (TU/e), Insights on Learning Tractable Probabilistic Graphical Models
16 Peter Blomsma (TiU), Building Embodied Conversational Agents: Observations on human nonverbal behaviour as a resource for the development of artificial characters
17 Meike Nauta (UT), Explainable AI and Interpretable Computer Vision – From Oversight to Insight
18 Gustavo Penha (TUD), Designing and Diagnosing Models for Conversational Search and Recommendation
19 George Aalbers (TiU), Digital Traces of the Mind: Using Smartphones to Capture Signals of Well-Being in Individuals
20 Arkadiy Dushatskiy (TUD), Expensive Optimization with Model-Based Evolutionary Algorithms applied to Medical Image Segmentation using Deep Learning
21 Gerrit Jan de Bruin (UL), Network Analysis Methods for Smart Inspection in the Transport Domain
22 Alireza Shojaifar (UU), Volitional Cybersecurity
23 Theo Theunissen (UU), Documentation in Continuous Software Development
24 Agathe Balayn (TUD), Practices Towards Hazardous Failure Diagnosis in Machine Learning