RPTU

Accountability of AI-based Algorithmic Decision-Making Systems - from Theory to Software Engineering Practice

Thesis approved by
the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Marc Hauer

Date of Defense: 13.12.2024
Dean: Prof. Dr. Christoph Garth
Reviewer: Prof. Dr. Katharina Zweig
Reviewer: Prof. Dr. Peter Liggesmeyer

DE-386

“*Writing a doctoral thesis is like a voyage of discovery at sea. At first, you accompany your captain as they explore the island they have discovered on their own. Little by little, you set off on your own and discover new areas or even new islands, becoming a captain yourself. You rarely stay alone for long on your voyages because all too often, you meet other explorers with whom you can then share the joys and sorrows of exploration.*”

from memory after Prof. Dr. Katharina Zweig

Abstract

Rapid advancement and widespread adoption of Artificial Intelligence (AI) technologies have revolutionized numerous domains, transforming how we live, work, and interact. However, along with the immense potential of AI-based systems, there are significant challenges and risks that need to be addressed. One critical need is the regulation of such systems, particularly those making decisions about people as well as their property, so-called algorithmic decision-making (ADM) systems. Traditional regulatory approaches are inadequate for AI-based systems due to their vast input and state spaces, reliance on training data, complex inner structures, and the fast-paced nature of technological developments. Existing regulatory frameworks struggle to keep up with these novel challenges, requiring new approaches to ensure the responsible and ethical deployment of such technologies. Moreover, the current understanding and operationalization of key concepts such as fairness, explainability, and accountability in the context of AI-based systems remain vague, hindering the development of effective methods and solutions.

Against this backdrop, the following scientific question arises: How can we address the challenges associated with the development, use, and control of AI-based systems to ensure their responsible and trustworthy deployment?

This thesis employs a multidisciplinary approach to address the challenges associated with accountability in AI-based systems. The research is based on gaining an understanding of what such accountability means. Therefore, a generic software development process is dissected into sections, each examined separately to identify transparency and inspectability mechanisms. Building upon these mechanisms, various auditing procedures are explored, and the concept of certificates is introduced to ensure trustworthy audits. After this, testing of data-driven components and AI-based applications is described, focusing on fairness testing and its application within the audit procedures. As all approaches for promoting accountability require sufficient incentives to implement them, this thesis also reviews various approaches aimed at providing such incentives. It examines the role of the risk-based regulation approach suggested by the upcoming European AI Act and the recent Corporate Digital Responsibility (CDR) approach, highlighting potential benefits and areas for improvement.

The multidisciplinary approach provides a comprehensive toolbox of methods and concepts to establish accountability in AI-based ADM systems. By providing insights into the effectiveness of these approaches, this work contributes to shaping future regulatory frameworks and promoting intrinsic motivations for accountability. Overall, this thesis not only identifies limitations in current approaches but also offers practical solutions and outlines future research directions to further advance the field of AI accountability. By fostering responsible AI development and usage, these results contribute to the long-term benefits of AI in society, ensuring that AI technologies can be deployed ethically, transparently, and in alignment with societal values.

Acknowledgements

I would like to thank the many people who have accompanied, challenged, encouraged, and supported me during my doctorate. However, I want to explicitly thank a few individuals to whom I am particularly grateful:

First and foremost, thanks go to my parents Manfred and Gertrud Hauer, who made it possible for me to only have to worry about my grades during my studies. Without your support, I would not be where I am today. Even though I have some problems showing my gratitude, I am aware of your support and deeply grateful for it.

Thanks go to my supervisor Katharina Zweig, who always showed me where I can still improve myself and what I should do, but also made me permanently reflect on what I do well and what I can be proud of.

Thanks to Tobias Krafft, who not only always encouraged me to reach for the stars, but who is also a great friend at all times.

Additional thanks go to Johannes Kevekordes, Michael Gamer, Stephanie Borgert, Maryam Amir Haeri, Catharina Rudschies, Christopher Koska, Michael Binzen, and Kim Valerie Carl for the many discussions we shared, on- and off-topic.

Finally, I am deeply grateful to my wife Carolin Flach for putting up with me during the stressful phases and for the amazing figures she designed for me.

Contents

List of Abbreviations

Data definitions

$X = (X_1, X_2, \ldots, X_m)$	Input vector of a data instance
$A = (A_1, A_2, \ldots, A_l)$	Part of an input vector containing only sensitive attributes
$X' = (X'_1, X'_2, \ldots, X'_{m-l})$	Part of an input vector containing only insensitive attributes
$X_A = (X'_1, X'_2, \ldots, X'0_{m-l}, A_{m-l+1}, \ldots, A_m)$	Input vector explicitly containing insensitive and sensitive attributes
$Y = (Y_1, Y_2, \ldots, Y_n)$	Output vector of a data instance
$\hat{Y} = (\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_n)$	Output vector of an ADM system

Abbreviations

AC	Assurance Case
ADM	Algorithmic Decision-Making
AI	Artificial Intelligence
AMS	(Austrian) Arbeitsmarktservice
AMS algorithm	Austrian Unemployment Risk Assessment Software developed by the AMS
ANN	Artificial Neural Network
API	Application Programming Interface
ATDD	Acceptance Test-Driven Development
BL	Basic Law
CDR	Corporate Digital Responsibility
DDC	Data-Driven Component
DDM	Data-Driven Model
EU FR-Charta	European Charta of Fundamental Rights
FDR	False Discovery Rate
FN	False Negative
FNR	False Negative Rate
FOR	False Omission Rate
FP	False Positive
FPR	False Positive Rate
GAET	General Act on Equal Treatment
GDPR	General Data Protection Regulation
GSN	Goal Structuring Notation
HLEG-AI	High-Level Expert Group on Artificial Intelligence
ISTQB	International Software Testing Qualifications Board

MCC	Matthews Correlation Coefficient
ML	Machine Learning
NPV	Negative Prediction Value
PLS	Platform Learning Systems
PPV	Positive Predictive Value
SMART	Specific, Measurable, Attainable, Realisable, Time bounded
SVM	Support Vector Machine
TDD	Test-Driven Development
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
QA	Quality Assurance

Chapter 1

Introduction

Artificial Intelligence (AI) has become a ubiquitous technology that has been making a significant impact across various domains. While specific definitions of the term 'AI' are much debated, it generally refers to the development of computer systems that can perform tasks that typically require human intelligence, such as reasoning, problem-solving, learning, perception, and natural language processing (Kok et al., 2009). AI has been utilized in diverse fields, including human resources (Palos-Sánchez et al., 2022), healthcare (Abràmoff et al., 2018), finance (Goodell et al., 2021), predictive policing (Brayne & Christin, 2021), transportation (Bharadiya, 2023), and education (L. Chen et al., 2020), among others[1]. Thus, it has been transforming the way people live, work, and interact with the world around them.

As AI technologies continue to advance, there is increasing interest in exploring their potential and understanding their limitations. This interest has led to a proliferation of research studies, applications, and innovations in AI, resulting in a growing body of knowledge and a broad range of implications for society (Gao & Ding, 2022). The relevance of AI in shaping the future of humanity can barely be overstated, as it is even considered one of the main drivers of the *fourth industrial revolution* (Schwab, 2017).

Today, many risks in the context of AI are well known. There are entire collections of AI-based algorithmic decision-making (ADM) systems whose behavior has led to unexpected or undesirable negative consequences for individuals, groups of people, or society as a whole (e.g., McGregor, 2021[2]).

As decisions made by ADM systems potentially affect the lives of millions of people, the need to regulate them is great (Krafft et al., 2022). Although classical regulatory approaches work well for traditional ADM systems, such as expert systems, trying to regulate systems that are based on at least one AI component leads to new challenges (e.g., Latonero, 2018; Coeckelbergh, 2020).

The input and state spaces of AI-based systems are enormous even for common tasks, rendering exhaustive testing infeasible. Their behavior strongly depends on the data used to train them, and inconsistencies or deliberate manipulations of this data can cause grave consequences. Most currently used AI-based systems have a complex inner structure that

[1] See, for example, the collection of AI-based software products and projects in Germany provided by the '*Platform Learning Systems*' (PLS), https://www.plattform-lernende-systeme.de/ki-landkarte.html, last accessed on August 17, 2023.

[2] https://incidentdatabase.ai/, last accessed on August 11, 2023.

does not lend itself to human interpretation. This makes finding malfunctions or attacks and mitigating them a very hard task. Furthermore, regulators have difficulties understanding the systems of interest. Also, in the analog world, companies have usually moved slowly and new markets have taken decades to develop (Tang et al., 2020). Therefore, legislators, regulators, and people working in standardization have had enough time to work out the regulatory needs of a particular market, implement them, and refine them over time, before major harm can be done. With today's fast-developing technologies, this is no longer the case. Legislators, regulators, and people working in standardization are forced to catch up with insufficient regulatory frameworks and to think preventively about potentially upcoming developments. This is essential, especially as ethically questionable consequences and legal transgressions can often only be detected months after product release – maybe after some damage has already been done – even by the providers of AI-based products (see, e.g., Example 2, p. 21 and Example 4, p. 38).

To counter such risks, in June 2018 the European Commission appointed a group of experts to provide advice on its AI strategy, the so-called *High-Level Expert Group on Artificial Intelligence* (HLEG-AI)[3]. Their findings elaborate the abstract goals that need to be pursued in order to arrive at an engagement with AI technologies in which we can trust, both as individuals and as a society (High-Level Expert Group on AI, 2019a, 2019b, 2020a, 2020b). These findings provide a basis for clarifying 'what' is actually to be achieved. They refer to seven key requirements: (i) human agency and oversight, (ii) technical robustness and safety, (iii) privacy and data governance, (iv) transparency, (v) diversity, non-discrimination and fairness, (vi) environmental and societal well-being, and (vii) accountability. Together, these goals are subsumed under the term *Trustworthy AI* (High-Level Expert Group on AI, 2019a). Unfortunately, these key requirements offer a wide scope for interpretation and are also highly interdependent to a large extent. In particular, the goal of establishing accountability is difficult to distinguish from the other goals, due to its many different and partly very broad meanings. If it is defined as *'a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgment, and the actor may face consequences*' (Bovens, 2007, p.452), all the other goals can be understood as relevant components to achieve accountability (see Chapter 3). Consequently, it can also be understood as a certain interpretation of trustworthiness (see Section 2.2). Beyond that, the HLEG-AI hardly addresses "how" concrete implementations can be realized and does not even make any proposals in this regard.

From the beginning, the aim of this work had been to contribute to this "how", i.e., to develop methods to improve the development, use, and control of AI-based systems that have a direct impact on individuals or society as a whole. However, it became apparent early on that many of the prerequisites for developing such methods have not been met or have been met only inadequately. In most cases, the "what" alone is not

[3] https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai, last accessed on September 09, 2023.

specified sufficiently. Terms such as fairness, explainability, and accountability are used with barely any indication of the complex concepts hidden behind them, which cannot be operationalized in a generally valid way.

This thesis therefore addresses the challenges inherent in many of these concepts. Furthermore, it outlines and tests new approaches to solving them by following a multidisciplinary approach that combines theoretical and empirical research. In addition, new concepts or extensions of existing concepts that aim to improve the development, deployment, and control of AI-based systems are explained.

1.1 Motivation

After obtaining a Master's degree in Applied Computer Science, I started working as a software developer in the 3^{rd} level support of a medium-sized company at the beginning of 2018, and spent 15 months there. We were responsible for support cases that needed to be traced back to the implementation level to find out exactly how an error occurred and how to fix it or, if necessary, to implement new features that would provide relief to 1^{st} and 2^{nd} level support in the long run. We were also responsible for managing automatic tests that were supposed to ensure software quality and stability.

During this time, there was an abundance of media reports on AI-based systems not working as expected or having undesirable side effects. Reading many of these reports, I often thought 'obviously this wouldn't work like this', 'this should have been noticed during development or during quality assurance', and 'of course, this effect was to be expected in the long run'. I justified my considerations by the fact that traditional software development works well as long as the appropriate processes are developed, implemented, controlled, and continuously improved. My initial motivation for this thesis was therefore to transfer the effective processes that already existed in traditional software development to the development of AI-based systems or to develop them wherever necessary.

When I started doing research on this topic, unexpected challenges emerged in the context of AI-based applications. They included the use of ambiguous terms, the complexity behind measuring (non-) discrimination, and the challenge of testing data-driven components in a meaningful way. In addition to that, the standards landscape is still insufficiently developed (Adler et al., 2021, p.43), and the huge number of documents on methods and concepts for improving AI-based systems barely provides any indication of how they could be implemented in practice. Thus, the goal of this thesis has gradually evolved to address these topics instead. In summary, my motivation was to improve the accountability of (primary) AI-based software systems and their development processes.

1.2 Own Contributions

Much of this work is based on interdisciplinary research with many other scientists. To distinguish between contributions that are mine alone and

contributions that are part of a collective effort, I will refer to 'I, me, and my' or 'we, us, and our', respectively. In the latter case, I will list the individuals belonging to the group referred to. Most of this joint work took place within the scope of projects or collaboration with external organizations, which are briefly introduced below.

1.2.1 Contributions in Projects and Organizations

ExamAI – Testing and Auditing of AI: The ExamAI project, which was funded by the *German Federal Ministry of Labour and Social Affairs* (grant number code DKI.00.0002 3.20), was led by the *German Informatics Society (Gesellschaft für Informatik e.V.)* and consisted of an interdisciplinary team of (socio-)computer scientists, software engineers, legal scientists, and political scientists. On the basis of eleven use cases in the application areas 'Human-Machine Cooperation in Industrial Production' and 'AI-based Systems in Human Resources and Talent Management as well as in Recruiting' that were identified at the beginning of the project, the team explored what appropriate control and test procedures for AI-based systems could look like. As part of this project, I co-authored the publications Krafft et al., 2020, K. Zweig et al., 2020, Hauer and Zweig, 2021, Hauer, Adler, and Zweig, 2021, Jöckel et al., 2021, Adler et al., 2021 and Krafft et al., 2023, partly during and partly after the project.

GOAL – Governance of and by Algorithms: The governance of and by algorithmic decision-making systems based on Machine Learning methods was the research subject of the project GOAL, funded by the *German Federal Ministry of Science and Education* (grant number 01IS19020). On the one hand, the project focused on the need for governance instruments, such as value-oriented technology design, self-regulation, and technical standards, on the possibilities they offer, and on the gaps that still need to be closed. On the other hand, it was discussed to what extent algorithms themselves can perform governance functions in order to reduce risks or even avoid them altogether. Based on these discussions, needs for action were identified and addressed. In this project, I co-authored the publications Hallensleben et al., 2020, Hauer, Kevekordes, and Haeri, 2021, Hoffmann et al., 2022a, Kevekordes et al., 2022, Hoffmann et al., 2022b, Hauer, Krafft, Sesing-Wagenpfeil, Zweig, et al., 2023 and Hauer, Krafft, and Zweig, 2023. We also gained many insights that could not be published for a scientific audience but were part of the undisclosed final report, in the hope that they will impact future projects.

Standardization Roadmap of Artificial Intelligence: On behalf of the *German Federal Ministry of Economics and Climate Protection*, DIN and DKE organized the work on the *German Standardization Roadmap for Artificial Intelligence*. With the participation of more than 570 experts from industry, science, the public sector, and civil society, the strategic roadmap for AI standardization was developed. With this roadmap, a part of the German Federal Government's AI strategy was implemented, and thus an essential contribution to "AI – Made in Germany" was made. I contributed to the first (DIN/DKE, 2020) and the second (DIN/DKE, 2023)

version of the roadmap as a member of multiple working groups and an expert on various vertical topics. As the glossary we developed in the glossary working group exceeded the designated space for that chapter by far, only the most important terms were published. However, we[4] received permission to publish the full glossary separately and plan to continuously extend it over the next years.[5]

etami – Ethical and Trustworthy Artificial and Machine Intelligence: *etami* is a non-profit organization that works on bringing ethical AI principles to action. *'By translating European and global principles for ethical AI into actionable and measurable guidelines, tools and methods, etami supports and promotes trustworthy and ethical design, development, and deployment of AI-based systems. The consortium started working in 2019 on quality standards and conformity assessment for AI software. As an initiative of the Machine Learning Research Lab of Volkswagen Group in Munich, it gathered 17 multinationals and universities to jointly develop excellence in AI methods. In 2022, etami joined BDVA[6] and is hosted as a Task Force since'*.[7] The regular interdisciplinary discussions with other scientists, practitioners, and industry representatives in this organization helped me to critically reflect upon my work and the works of others. I contributed to the etami open online guidebook[8], which aims to support researching, developing, and applying AI methodologies, mainly regarding the topics of *Accountability* and *Assurance Cases*.

fAIr by design: *'fAIr by design is a research project involving eight partners from Austria (five companies, two universities and one NGO), focusing on the practical implementation of fairness requirements into AI-based systems. It is a three-year project, running from 2021 until 2024, funded by the National Foundation for Research, Technology and Development Austria'*.[9] I participated as an external expert to help develop an Assurance Case together with the companies *winnovation consulting GmbH*, *Rania Wazir e.U.*, and *rotable* in order to assure that an actual software product developed by *rotable* can be considered fair. In this project, I co-authored the publications Kunze et al., 2023 and Hauer, Müller-Kress, et al., 2023.

1.2.2 Thesis Outline and Contributions Focused on Research Questions

In addition to the objectives of the projects and organizations mentioned above, this thesis also contributes to individual research questions, which are derived from the motivation on the one hand and from the findings of

[4] In this Section, "we" refers to the authors of Runze et al., 2023.
[5] https://www.ai-glossary.org/, last accessed on May 16, 2023.
[6] Big Data Value Assosiation, https://www.bdva.eu/about, last accessed on March 09, 2023.
[7] https://www.etami.org/, last accessed on March 09, 2023.
[8] https://etami.gitlab.io/guidebook/, last accessed on June 19, 2023.
[9] https://www.fairbydesign.eu/, last accessed on April 20, 2023.

the individual projects on the other hand. The following research questions also guide the structure of this thesis.

In Chapter 2, the most relevant terms for this thesis are explained and linked to each other, namely: ADM systems, Machine Learning, Artificial Intelligence, trustworthiness, accountability, bias, discrimination, and fairness. Furthermore, it is defined how these terms are to be understood in the context of the thesis. Consequently, the first contribution of this work is the identification and active resolution of conflicting definitions.

The abstract goal of "fairness" is fundamental within the scope of this thesis, leading to the following question:

RQ 1

What considerations are relevant when selecting fairness measures?

The contribution of this thesis consists of a comprehensive presentation of the topic itself, taking into account the various relevant aspects to consider. Additionally, we[10] provide a new hierarchical perspective in order to consider group fairness and individual fairness in relation to each other and thus solve potential conflicts between these two perspectives.

Chapter 3 addresses the question of how *accountability* can be achieved. Recent scientific work dealing with AI often refers to the accountability explanations provided by Maranke Wieringa, who maps the definition provided by Mark Bovens to the field of AI (Wieringa, 2020). To meet Mark Bovens' definition of accountability, four conditions need to be met:

- It is possible to analyze the system under question.
- It is clear which actors are to be held accountable.
- It is clear which forums hold the actors accountable.
- There are processes that result in consequences for the actors based on the forums' judgments.

The following sections investigate the possibilities for satisfying each of these conditions, starting with the question:

RQ 2

What mechanisms can be implemented to allow system analysis?

To answer this question, we[11] decompose a generic software development process into its subcomponents and investigate for each of them which aspects can be analyzed, which pieces of information need to be disclosed, and which accesses need to be granted to enable analysis. We also elaborate on which actors are to be held accountable towards which forum and what consequences may follow a negative judgment. As the decisions for accountable actors either lie with a company or with regulating bodies, I also deal with a more theoretical approach based on John Austin's Speech Act theory (Austin, 1962) to support these decisions, leading to the question:

[10] Here, 'we' refers to the authors of Haeri Amir et al., 2023.
[11] Here, 'we' refers to the authors of Hauer, Krafft, and Zweig, 2023.

RQ 3

How to determine which actors are to be held accountable based on John Austin's Speech Act theory?

With access to an ADM system, a forum is able to submit self-constructed test data sets to the system and evaluate the results. There are various specific concepts to do so, referred to as *audits*, which are the topic of Chapter 4. This leads to the question:

RQ 4

What kinds of audits are applicable for analyzing AI-based ADM systems?

The chapter discusses the very different audit conceptions of standardization bodies, such as the ISO, which can be used as a basis for accredited certification, and Sandvig et al. (Sandvig et al., 2014), who understand (software) audits as some sort of field experiment. It also elaborates on our practical experiences when we[12] attempted to audit the Facebook News Feed algorithm. In the next section, the chapter introduces the Assurance Case framework, which was developed as a requirements engineering tool for safety and security engineering, and describes how we[13] extended it to fulfill ethical requirements. It concludes with the experiences we made when applying it to assure fairness on an actual software product together with the developing company and other domain experts, thus answering the question:

RQ 5

How suitable is the Assurance Case framework from the field of safety engineering to be applied on extra-functional requirements such as fairness?

The basis of many audit assessments (whether in the context of audits according to ISO or in the context of audits according to Sandvig et al., 2014) is testing. Tests can also provide evidences as required by an Assurance Case. The question of which tests exist and under which conditions they are applicable is the topic of Chapter 5. As the functionality of data-driven components (DDCs) of AI-based systems is automatically derived from data and not implemented by a programmer, traditional test approaches have been transferred and new test approaches have been developed to suit the needs of DDCs. As a result, new test-related terms have also emerged that hamper communication about tests, leading to the question:

RQ 6

How does traditional test terminology differ from test terminology in the context of DDCs?

[12] Here, 'we' refers to the authors of Krafft et al., 2020.
[13] Here, 'we' refers to the authors of Hauer, Müller-Kress, et al., 2023.

To address this challenge, we[14] provide a comparison of the most important test-related terms from traditional software testing and DDC testing.

Another challenge is the extensive literature on software testing, which uses and categorizes many terms inconsistently. This makes it difficult to identify which tests are suitable for testing properties that are especially relevant for AI-based ADM systems. The relevant question is:

RQ 7

Which test-related terms are especially relevant in the context of AI-based ADM systems?

I reviewed some of the most frequently cited books regarding software testing and discussed the contents with various experts in the field to collect and structure the most relevant test-related terms for AI-based ADM systems on different levels of abstraction. The terms, their relationship with each other, and any ambiguities in the terminology that I encountered in the process are also presented in Chapter 5.

In the context of auditing black-box systems as described by Sandvig et al., 2014, only a few test methods are applicable to examine fairness. Therefore, we[15] elaborate on which test methods can be applied in the context of which specific audit under which specific conditions. This leads to the question:

RQ 8

Which test methods for fairness are applicable in the context of which audit procedures according to Sandvig et al. and under which conditions?

At the end of the chapter, the integration of the development of an Assurance Case into an Acceptance Test-Driven Development process is discussed, answering the question:

RQ 9

Can the Assurance Case framework be integrated into a test-driven development process?

Auditing and testing activities imply considerable effort, and thus costs, which often leads to them getting neglected. The first part of Chapter 6 therefore focuses on regulations as an extrinsic motivation approach to promote product quality and benevolence in the form of tests, audits, and certifications. It puts a strong focus on the idea of risk-based regulation approaches and the upcoming *harmonised rules on Artificial Intelligence*[16] (AI Act). It analyzes the following questions:

[14] Here, 'we' refers to the authors of Jöckel et al., 2021.

[15] Here, 'we' refers to the authors of Krafft et al., 2023.

[16] European Commission, Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts COM(2021) 206 final, https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206.

RQ 10

What are the challenges of a risk-based regulation approach for AI-based systems, and which aspects are particularly relevant for assessing their risks?

and

RQ 11

How will the upcoming AI Act affect the AI landscape, and how can this be determined in advance?

The second part of the chapter explores the novel concept of Corporate Digital Responsibility (CDR) as a counter-approach to providing intrinsic motivation to promote product quality and benevolence. It answers the following question:

RQ 12

What role does the increasingly recognized concept of CDR play?

It also introduces our[17] ongoing work on a software solution that supports the assessment of a product's CDR level.

To conclude this thesis, Chapter 7 provides a summary of the results presented and reflects upon the connections between the topics as one big toolbox for algorithmic accountability. Furthermore, it presents ideas for future work in this area of research.

[17] Here, 'our' refers to refers to Kim Valerie Carl, Dr. Thomas Arnold, and me.

Chapter 2

Foundations

Many terms related to AI are often not clearly explained or defined in the literature. Although widely used, there is no agreed understanding of them, and their meaning is basically just assumed more often than not. For this reason, this chapter discusses the most relevant terms and links them to each other. Furthermore, it defines how they are to be understood in the context of this thesis.

2.1 ADM Systems, Machine Learning, and Artificial Intelligence

Algorithmic decision-making systems (ADM systems)[1] are systems that are able to automatically make decisions, including predictions, recommendations,[2] and actions, based on predefined rules and some input, more specifically, on input data (Diakopoulos, 2020). How these rules are generated plays a crucial role in current debates revolving around the regulation of ADM systems.

If, for example, the rules are specified by human experts, the ADM system is called an *expert system*. In this case, the experts' rules can be challenged with regard to ethical or legal aspects and adjusted accordingly if needed.

Another possibility is to automatically derive rules from data by applying some form of supervised *Machine Learning* (ML) procedure. Supervised learning is a strategy *'in which the correctness of acquired knowledge is tested through feedback from an external knowledge source'*.[3] In this case, the ADM system is based on two different source codes. The first one is the ML method that *trains* a statistical model (see Definition 1) based on a set of *training data*. Such a data set is also called a *training data set*.

[1] Some authors use the abbreviation for *automated decision-making systems* (e.g., Larus et al., 2018; Felzmann et al., 2020). The terms are usually understood to be interchangeable.

[2] In case of recommendations, some sources also refer to *decision support systems* (DSS; e.g., Tan et al., 2010) or *computerized decision support systems* (CDSS; e.g., Liberati et al., 2017).

[3] ISO/IEC 2382:2015 Information technology — Vocabulary.

Definition 1 (Machine Learning Method/Procedure/Model)

The terms *Machine Learning Method, Procedure, and Model* are often used interchangeably in the literature, but may refer to different things. To avoid confusion, in this thesis, ML method refers to the 'type' of Machine Learning (e.g., Support Vector Machine, Artificial Neural Network, Decision Tree, etc.). ML procedure refers to the training process (e.g., supervised or unsupervised training, stopping criteria, etc.), whereas ML model refers to the specific result of a Machine Learning procedure (e.g., a specific Artificial Neural Network with a given number of layers and a given number of neurons per layer). A Machine Learning method is sometimes also referred to as an '*untrained model*', which might lead to misunderstandings.

Each point in the training data is a vector of parameters that can be divided into two components: *Input* information $X = (X_1, X_2, \ldots, X_n)$ and *output* information $Y = (Y_1, Y_2, \ldots, Y_m)$. Input information refers to the parameters that are later to be passed to the resulting statistical model in order to make decisions based on them, i.e., to produce outputs. To avoid confusion, the output produced by the model is notated as $\hat{Y} = (\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_m)$. The usage of this model is the second source code of such an ADM system (see Figure 2.1). The output information labels the target output of the decision model (see Definition 2). It is needed in the training data to find correlations between the input information and the target output.

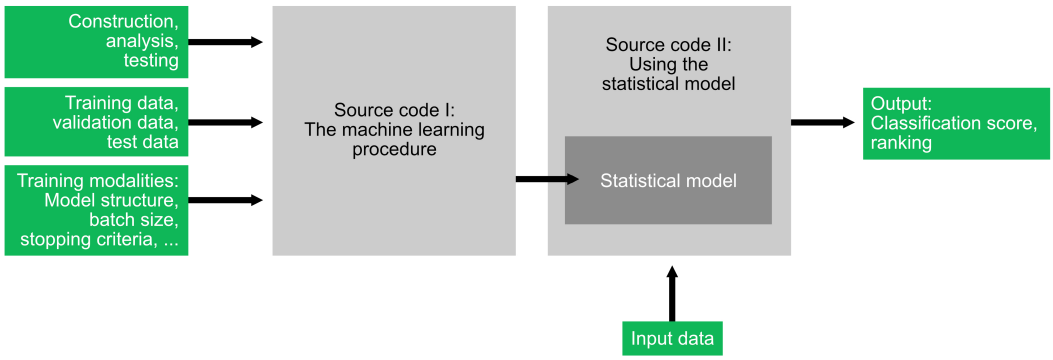


Figure 2.1: ADM systems whose decision structure is derived from data are based on two different source codes: The first code builds a statistical model based on the data and training modalities. This part is referred to as Machine Learning. The second code computes a classification/score/ranking for new data based on the statistical model built by the first code. Figure based on Figure 3 in Hauer, Krafft, and Zweig, 2023.

Definition 2 (Input and Output Information)

A data point may be divided into input information $X = (X_1, X_2, \ldots, X_n)$ and output information $Y = (Y_1, Y_2, \ldots, Y_m)$. To avoid confusion with the output information produced by a model (also called *system output*), the output information in the data is also called 'label' or *target output*. The goal of a supervised learning process is to deduce a decision model based on the correlations between input information and label(s). An input is also the information that is fed into an ADM system to produce an output $\hat{Y} = (\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_m)$ based on the learned model.

How exactly decision rules are derived from data depends on the learning procedure and the specific model to be trained. A data set containing input and output information is called a *ground truth* (see Definition 3).

Definition 3 (Ground Truth)

In the context of ML, the term '*ground truth*' refers to a collection of data that reflects information that is free from errors.[4]) Dividing each data point into three components (input data, output data, and irrelevant data) provides the basis for training ML models. Based on the input and output data, the training process builds a structure that, in the best case, provides the correct output information for each instance of input information. The combination of input information and the matching output information assumed to be correct is called ground truth. The quality of a trained model can be determined based on the ground truth for a given X by comparing the corresponding Y and $\hat{Y}$.

There are also training processes that do not require output information: so-called *unsupervised learning*.[5] Such processes aim for a fundamentally different goal. While supervised learning tries to find correlations between input and output information, unsupervised learning tries to find patterns in the data (see Example 1). In this case, the data is not split into input and output information.

[4] See ISO/IEC DIS 22989 - Information technology - Artificial intelligence - Artificial intelligence concepts and terminology, section 3.2.6.

[5] See ISO/IEC DIS 22989 - Information technology - Artificial intelligence - Artificial intelligence concepts and terminology, section 3.2.23.

Example 1 (Credit Score Assignment)

Assuming a bank has client information about age, gender, income, and whether or not a loan has been approved. To evaluate future loan applications, either a supervised or an unsupervised learning approach can be used. Finding correlations between age, gender, and income (as input information) and whether a loan has been approved or not (as output information) would be a supervised learning task. The resulting model could then be used to automatically evaluate future loan applications. A possible unsupervised learning task would be to identify typical types (so-called clusters) of loan applications by analyzing the input information for similarities. Based on these, loan applications could be identified that, under comparable circumstances, could be repaid in the past. In this case, all information is used as input information.

When an ADM system is built on an ML component, it is considered an *Artificial Intelligence* (AI-based) system. This definition provides a boundary that is usually not disputed. Beyond that, however, many non-ML systems and concepts are sometimes also classified as AI, which results in different possible approaches to defining AI (Jiang et al., 2022). For the first version of the *Harmonised Rules on Artificial Intelligence*[6] (AI Act), the European Commission chose a technology-based definition of AI (see AI Act, Annex I):

(a) *'Machine learning approaches, including supervised, unsupervised, and reinforcement learning, using a wide variety of methods including deep learning;*

(b) *Logic- and knowledge-based approaches, including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning and expert systems;*

(c) *Statistical approaches, Bayesian estimation, search and optimization methods.'*

In other definitions, the underlying technology is less relevant. Instead, AI is defined in terms of performance or purposes. John McCarthy, the father of the term *Artificial Intelligence*, said: *'It [Artificial Intelligence] is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable'* (McCarthy, 2007, p.2).

[6] European Commission, Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts COM(2021) 206 final.

On June 14, 2023, the European Parliament adjusted the definition of AI [systems] in the AI Act to *'a machine-based system that is designed to operate with varying levels of autonomy and that can, for explicit or implicit objectives, generate outputs such as predictions, recommendations, or decisions that influence physical or virtual environments*'.[7] In all these cases, expert systems might be considered AI-based systems as well. The AI Act will be discussed in more detail in Section 6.1.

To avoid confusion, for the remainder of this thesis, the term AI is used for any kind of AI, while the term *data-driven component* (DDC) is used for any AI based on some sort of ML procedure. The resulting model is then referred to as *data-driven model* (DDM) (see Definition 4). As this thesis focuses on AI-based ADM systems, text-, image-, and sound-generating AI-based systems (so-called *generative AI*) are not considered explicitly. They warrant a separate discussion due to their distinct ethical considerations (e.g., media manipulation) and legal implications (e.g., regarding intellectual property).

Note that even though this work addresses the challenges of making ADM systems based on a DDC accountable, most of the approaches suggested for tackling these challenges are also applicable to other kinds of AI-based systems, such as expert systems.

[7] https://artificialintelligenceact.eu/wp-content/uploads/2023/08/AI-Mandates-20-June-2023.pdf, last accessed on September 26, 2023.

Definition 4 (Data-Driven Component (DDC) and Data-Driven Model (DDM))

An AI-based ADM system consists of at least one data-driven component (DDC) and any number of other software components. A DDC may be the only component of an AI-based ADM system, in which case the terms can be used synonymously.

The core of a DDC is a statistical model, also referred to as a data-driven model (DDM). A DDC may also consist of sub-components organized in pipelines that include some pre- and post-processing steps in addition to the trained DDM. A DDC, however, does not necessarily perform any pre- or post-processing steps. In this case, a DDM and a DDC are the same.

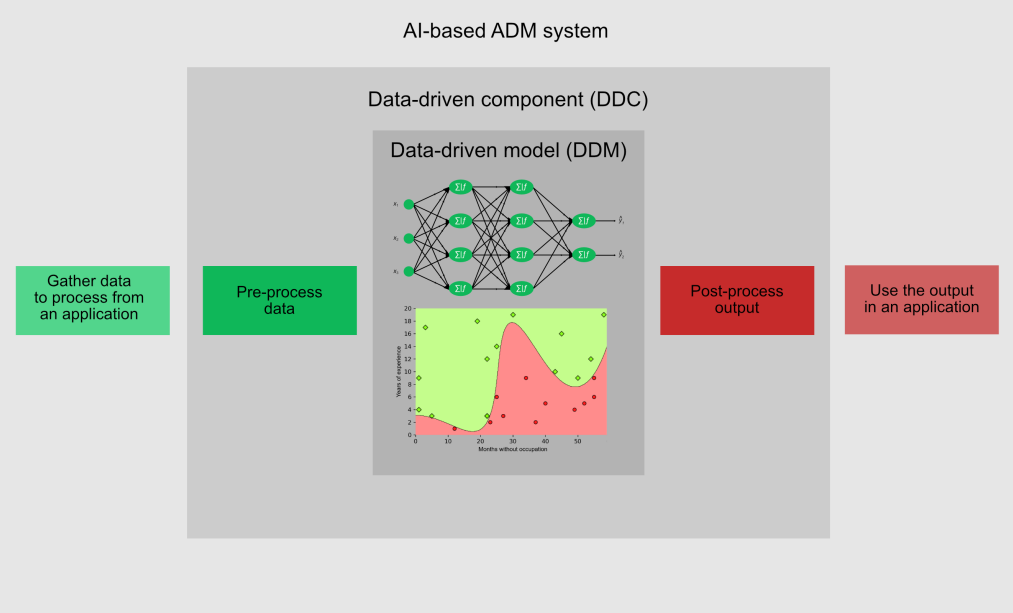


Figure 2.2: Visualization of the relationship between DDM, DDC, and AI-based ADM system.

On multiple occasions, this thesis refers to classifiers and classification systems. As representative examples of ML methods that are classifiers, the basic methods and related terms of *Support Vector Machines* and *Artificial Neural Networks* will be explained in Sections 2.1.1 and 2.1.2.

2.1.1 Support Vector Machines

In its simplest form, a Support Vector Machine (SVM) computes a dividing line that separates training data into two distinct categories, based on their label (Noble, 2006). This line is used to assign a category for future data without a label. Therefore, it is called a *classifier*.

To compute the classifier, each data point in the training data set is represented as a point in a Cartesian coordinate system with $n-1$ dimensions, where $n \in \mathbb{N}$ is the number of parameters each data point has. Each dimension corresponds to one of the $n-1$ features to be considered for categorization (input information). The point is placed on the respective axis according to the numerical value of the corresponding feature (see Definition 5). The only feature that is not used for positioning in the coordinate system is the label (output information). Depending on n, the classifier is not an actual line but a so-called *hyperplane*. A hyperplane is a subspace with one dimension less than the ambient space: For one-dimensional spaces, a hyperplane is a point, for two-dimensional spaces, it is a line, for three-dimensional spaces, it is a plane, and so on (Noble, 2006). The following explanations and visualizations refer to an SVM that considers only two dimensions (i.e., features) for categorization. However, all concepts can be transferred to an SVM with any number of dimensions.

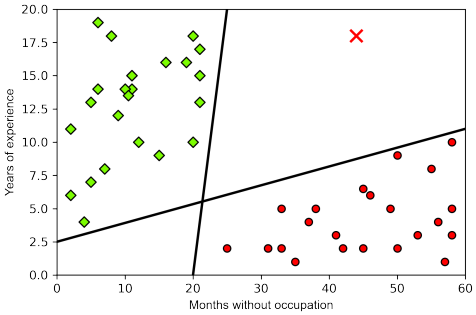
Definition 5 (Attributes, Features, Parameters, and Variables)

In software programming contexts, the terms 'attributes' and 'parameters' are clearly defined. Attributes are variables that denote the state of an object (e.g., an instance of a class). Parameters are variables that a function or method expects as input. As the scientific discussion of data-driven components usually abstracts from specific implementations, input information can be referred to as *attributes, parameters, or variables*. In the context of data-driven components, the terms attributes and features are used synonymously. There is a tendency to use the terms 'attributes' and 'features' when the focus lies on what kind of information the variables represent. The term parameters is used when the focus lies on a more technical level (like the weights of an Artificial Neural Network).[8] The term variables can be found in both situations.

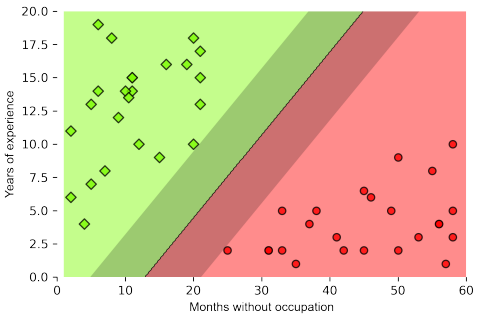
Assume that an ADM system is to be used to decide which candidate should receive an invitation to a job interview. An SVM is trained, based on the data of former job applicants who attended a job interview and were hired or rejected (see Example 2, p. 21). The considered features are *years of experience* in the field of expertise and *months without occupation*. Plotting the data points into a two-dimensional coordinate system shows an infinite number of possible classifiers. For example, both lines shown in Figure 2.3a are equally good, as each poses a perfectly dividing line. A new data point, however, may result in different categorizations based on the selected line. To compute an 'optimal' classifier, there are multiple approaches based on the shortest distance between the classifier and any data point, the so-called *margin* (Noble, 2006). Optimization for a margin

[8] See ISO/IEC DIS 22989 - Information technology - Artificial intelligence - Artificial intelligence concepts and terminology, section 3.1.28.

that is as big as possible therefore results in a *maximal margin classifier*[9] (see Figure 2.3b). The points that are adjacent to the margin and hence decisive for the determination of the classifier are the eponymous support vectors (Bishop & Nasrabadi, 2006, Chapter 7).



(a) Possible classifiers on the example data set that might provide very different future classifications.



(b) Maximal margin classifier based on the example data set.

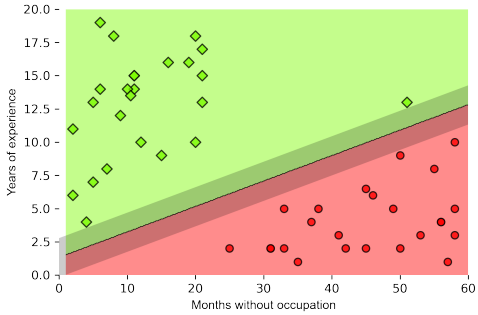
Figure 2.3: Example data set containing information about years of experience, months of occupation, and whether job applicants got hired or not.

For less clustered data, this option might not be a good choice, as one outlier can have a significant effect on the result (see Figure 2.4a). One possible solution to this problem is the use of the so-called *kernel trick* (Noble, 2006). With this method, the ambient space is extended by one hypothetical dimension to transform the originally linear classifier into a non-linear classifier (see Figure 2.4b). How this works can be visualized for two-dimensional examples. Imagine that, instead of cutting a line through a plane, the red area is pulled up with two fingers like a blanket (extension by one dimension). Then, a straight plane is cut through this blanket, which then falls down again. The red area is the cut-out part and the green area is the remaining part. What was a straight plane as long as the red part was pulled up is seen as a curve from a two-dimensional point of view.

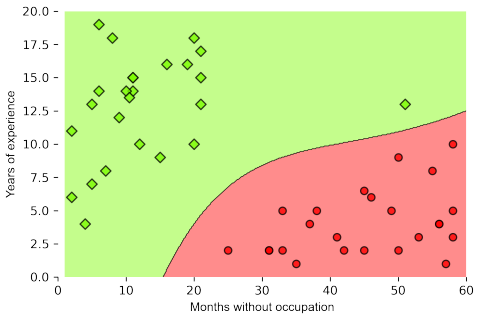
In data from real applications, there may be data points that seem to belong to one group based on their input information, but actually belong to the other group according to the ground truth. This may, for example, happen due to wrong assumptions in the ground truth or due to some unusual occurrences. The reasons do not matter, as an SVM might need to be able to deal with such a situation as well. The kernel trick might provide a solution, but it may also happen that the classifier is adapted to the training data to such an extent that the result is hardly meaningful for future data points (see Figure 2.5a). This is referred to as *overfitting* (Dietterich, 1995).

Another approach is to allow for some misclassifications (see Figure 2.5b). In this case, the margin is called a *soft margin*, which contains at least all misclassified data points, and the classifier is a *soft margin classifier*. All data points that fall within the range of the soft margin are support vectors (Bishop & Nasrabadi, 2006, p. 332).

[9] Sometimes also referred to as *maximum margin classifier*.

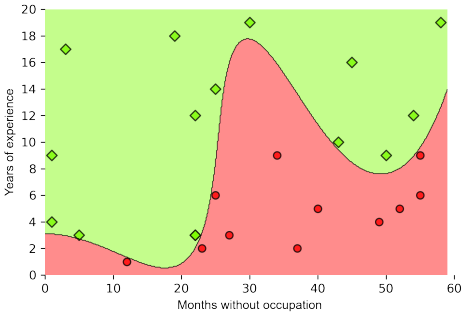


(a) Maximal margin classifier based on the example data set and one additional data point.

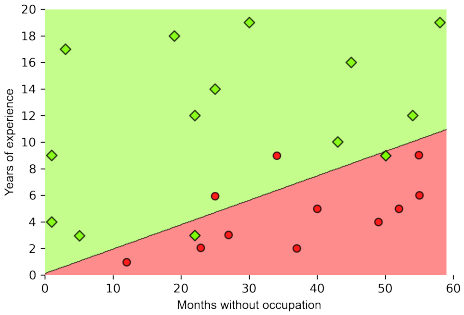


(b) Maximal margin classifier based on the extended data set using the kernel trick.

Figure 2.4: SVM results for a linear maximal margin classifier and a polynomial maximal margin classifier using the kernel trick on the example data set with an added outlier.



(a) Polynomial maximal margin classifier on a less clustered data set that results in overfitting.



(b) Soft margin classifier on a less clustered data set provided by Hoffmann et al., 2022a.

Figure 2.5: SVM results for a maximal margin classifier using the kernel trick to provide a polynomial classification and a soft margin classifier without using the kernel trick on a less clustered data set.

The choice of a kernel and a margin type are important parameters that need to be set prior to the training process. To distinguish them from the parameters introduced in Definition 5, p. 17, they are also referred to as *hyperparameters* (see Definition 6).

Definition 6 (Hyperparameter)

In the context of ML, a *hyperparameter* is any parameter that is independent of the training data and not a result of the training process (the weights in an Artificial Neural Network, for example, are considered (model) parameters but not hyperparameters) (L. Yang & Shami, 2020). In the training process of an SVM, some hyperparameters are the kernel and the margin. For an Artificial Neural Network (see Section 2.1.2), some hyperparameters are the number of layers, the number of neurons per layer, the activation function(s), the learning rate, the number of training iterations before weights are updated (the so-called *batch size*), and the conditions for concluding the training (*stopping criteria*).

There are many more details and sophisticated concepts to improve the basic idea of SVMs, but for the remainder of this thesis, this level of detail is sufficient. Bishop and Nasrabadi, 2006 provide a thorough overview and deeper explanation of the established concepts in the field.

Example 2 (Amazon Job Performance Prediction)

In 2018, *Reuters* reported that Amazon's AI recruiting tool shows a strong bias against women (Dastin, 2018). The tool expected anonymized CVs of job applicants and an open (technical) job position as input and suggested the most promising CVs for that position as output. However, even though the system had no information about the applicants' gender, it showed a strong bias against CVs from women. Subsequent analysis showed that the AI was able to deduce the gender based on information implying it, like being a member of a women's chess club. But even stripping the CVs of gender-implying information did not resolve the problem, as CVs from persons with different genders did indeed look differently due to the tendency to use gender-specific phrases and because the system had mainly been trained with CVs from men.

The development and application of AI recruiting tools face several other challenges. For example, the training process cannot take into account why a person was not hired. It is neglected whether the person was rejected by the company because they did not fit into the team, or whether the person withdrew their application because they had already accepted another offer or did not agree with the salary. Furthermore, there is a risk that such a system will only recruit people of the 'same type' over and over again. This potentially leads to an operational monoculture that is highly specialized for a specific task but does not provide a broad spectrum of other potentially valuable skills for current or future needs.

There are also a variety of AI recruiting tools based on information other than CVs. Especially critical are those that include facial expressions and gestures. Inferring characteristic traits or abilities from phrenologic or physiognomic information is generally considered to be pseudoscience (Stark & Hutson, 2021).

Usually, AI recruiting tools (including Amazon's) are only used as recommendation systems. The final decision lies with a human being who can react to these challenges. Nevertheless, it can be assumed that such a recommendation system has at least a slight influence on the final decision. After all, a human person must take responsibility for a decision that contradicts the recommendation of the machine. If the recommendation is followed, machine failure can still be blamed for the cause of undesired consequences.

2.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are an approach to simulate an abstract idea of the structure and function of the brain. They are modeled as directed graphs consisting of nodes and edges. The nodes represent the neurons of the network, the edges represent the connections between them. The edges have weights to amplify or dampen the information transported from one neuron to the next. They represent information on whether the receiving neuron is stimulated – in the context of ANNs, this is referred to *activation* – and how strong this stimulation is (Rosenblatt, 1958).

ANNs are organized into three types of layers: an input layer, any static number of hidden layers, and an output layer. Each layer again consists of any static number of neurons. In the simplest form of an ANN, each neuron of a layer is connected with each neuron of the next layer (see Figure 2.6). Such networks are called *feed-forward networks* (Sazli, 2006). If there is no hidden layer, i.e., if the network only consists of an input layer and an output layer, the network is called a *single-layer perceptron*. If there is at least one hidden layer, it is called a *multi-layer perceptron*.

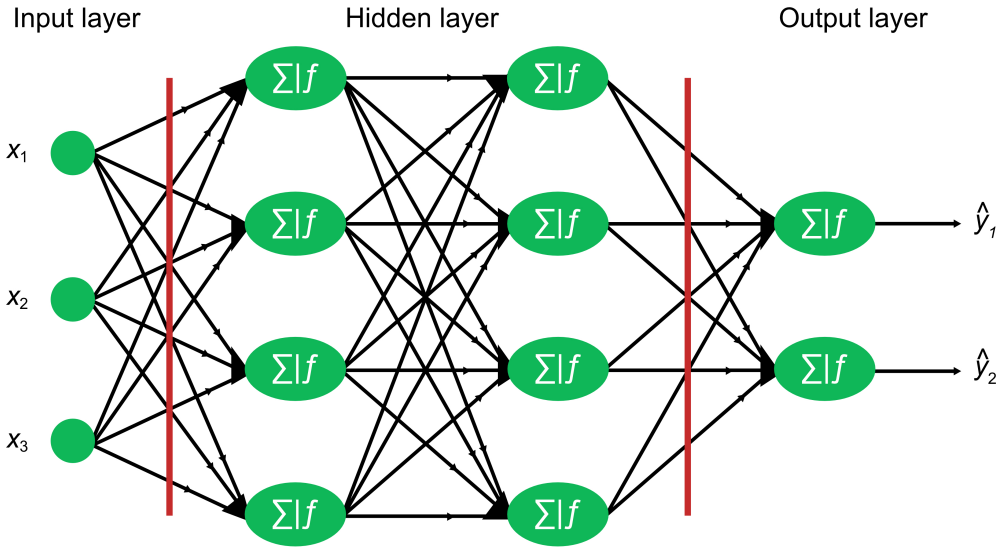


Figure 2.6: Multi-layer perceptron with two hidden layers each consisting of four neurons. It receives inputs with three parameters and outputs two values.

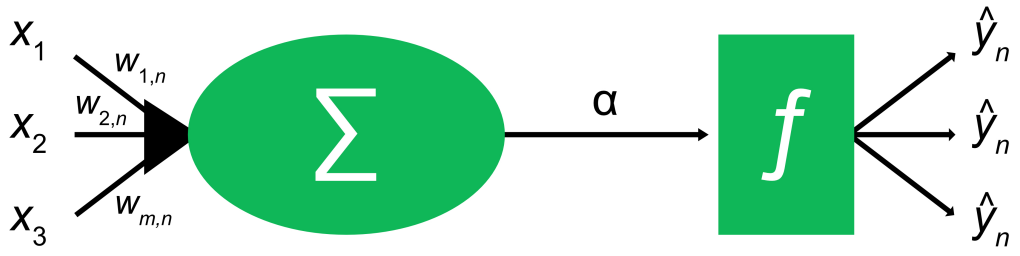
In feed-forward networks, incoming information x is multiplied by the weighting factor w of the edge transporting the information. In this way, each neuron receives one value for each neuron in the previous layer. The sum α of these values is normalized by a so-called *activation function* (Abraham, 2005). Simple activation functions output 1 if $\alpha \geq 0$ and 0 (or -1) if $\alpha < 0$. More sophisticated activation functions output a range between 0 (or -1) and 1, for example, sigmoid[10] functions (Sharma et al., 2017). The value resulting from the activation function represents how much a neuron is activated and also represents the output of a neuron

[10] S-shaped functions, such as the hyperbolic tangent function: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

$\hat{y}$. This output is propagated to the next layer of neurons to be received as input x. The neurons on the input layer receive the parameter values of an input vector and feed them into the network. There needs to be one neuron on the input layer for each parameter to be processed. The output produced by neurons on the output layer is also the actual output of the ANN. Depending on the last activation function, the outputs $\hat{y}$ can be values $\hat{y} \in \mathbb{N}$ – in this case, the model would be called a *Classifier* (see Section 2.1.1) – or values $\hat{y} \in \mathbb{R}$ – in this case, the model would be called a *Scorer* (see Definition 7).

Definition 7 (Classifier and Scorer)

If the output of an ML model is a value $\hat{y} \in \mathbb{N}$ that corresponds to a category, it is called a classifier. If the output is a value $\hat{y} \in \mathbb{R}$ that, for example, corresponds to a level of certainty, it is called a scorer. However, thresholds can be defined to transform the score that results from an ML model into categories, which allows using a scorer as a classifier. Also, there can be one output neuron per category. Then the scores can be interpreted as the probability of the input information belonging to any of the predefined categories. Such an approach allows for multi-label categorization.



$$x_1 \cdot w_{1,n} + x_2 \cdot w_{2,n} + \ldots x_n \cdot w_{m,n} = \sum_{j=1}^{m} w_{j,n} \cdot x = \alpha_n$$

$$f(\alpha_n) = \hat{y}_n$$

Figure 2.7: Schematic overview of the n^{th} neuron in a layer that receives m inputs from the previous layer in a multi-layer perceptron. Visualization based on Jain et al., 1996, p.34.

Initially, all weights w in the network are chosen randomly. During training, the weights are adjusted iteratively to produce better results. The most popular training method for ANNs is called *gradient descent*, for which multiple variations exist (Ruder, 2016). The variation explained in the following is called *stochastic gradient descent*. For a given set of input vectors X with respective output vectors Y that are considered to

be correct, the following steps are repeated for multiple training iterations (e.g., once per data point):

1. Compute the output vector based on one input vector (feed forward).
2. Compute the error e as the difference between the target output y and the system output $\hat{y}$.
3. Compute local gradients, starting at the output layer of the network.
4. Adjust the weights of each edge based on the output value of the respective source and the local gradient of the respective target.

As this process iteratively propagates the errors from the last layer back to the first layer, it is also called *backpropagation* (Abraham, 2005). In the following, these steps will be explained in detail based on Figure 2.8.

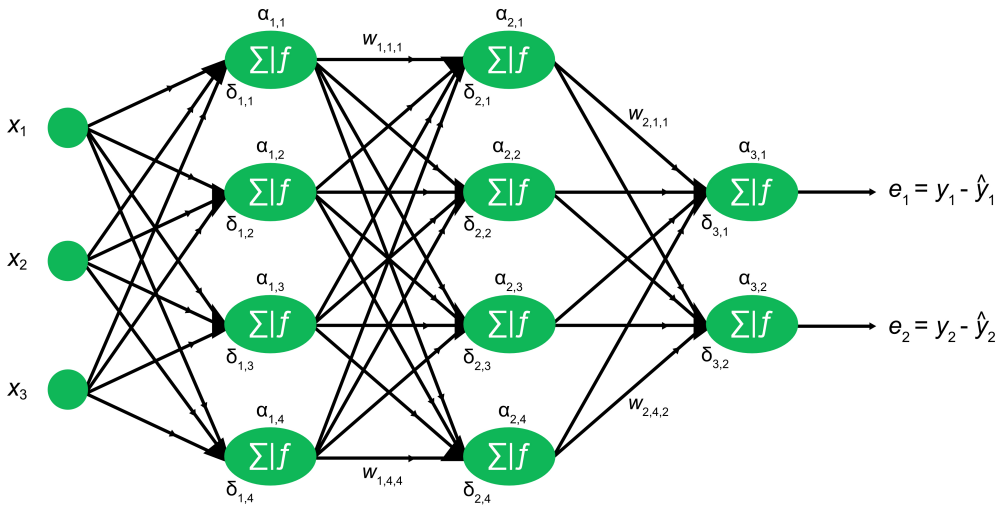


Figure 2.8: Previously introduced multi-layer perceptron, including relevant variables for the backpropagation algorithm.

The first step is the already explained computation of an output vector, based on the input vector of the data point. Next, the errors are computed as the difference between the target outputs and the system outputs $e = y - \hat{y}$ (see Equations 2.1 and 2.2).

$$e_1 = y_1 - \hat{y}_1 \tag{2.1}$$

$$e_2 = y_2 - \hat{y}_2 \tag{2.2}$$

These errors are used to compute the local gradients δ of the output layer neurons by inserting the respective sum result during the feed forward process α into the respective derived activation function and multiplying the result by the error e (see Equations 2.3 and 2.4). This computation is repeated iteratively for each layer by using the sum of all gradients δ of the neurons in the next layer multiplied by the respective weights of the connecting edges w as the error for the current computation (see Equations 2.5 and 2.6).

$$\delta_{3,1} = e_1 \cdot f'(\alpha_{3,1}) \tag{2.3}$$
$$\delta_{3,2} = e_2 \cdot f'(\alpha_{3,2}) \tag{2.4}$$
$$\delta_{2,1} = \Big(\sum_{n=1}^{2} w_{2,1,n} \cdot \delta_{3,n}\Big) \cdot f'(\alpha_{2,1}) \tag{2.5}$$
$$\vdots$$
$$\delta_{1,4} = \Big(\sum_{n=1}^{4} w_{1,4,n} \cdot \delta_{2,n}\Big) \cdot f'(\alpha_{1,4}) \tag{2.6}$$

where n refers to the target neuron of an edge.

Finally, each weight is adjusted by adding the product of the output value of the respective source node during feed forward and the local gradient of the respective target (see Equations 2.7 and 2.8).

$$\Delta w_{l,m,n} = \delta_{l,n} \cdot \hat{y}_{l-1,m} \tag{2.7}$$
$$w_{l,m,n,\text{new}} = w_{l,m,n,\text{old}} + \Delta w_{l,m,n} \tag{2.8}$$

where l refers to the layer of the neuron or the layer of the source neuron of an edge and m refers to the source neuron of an edge.

To prevent the stochastic gradient descent approach from resulting in overcorrection, $\Delta w_{l,m,n}$ is usually additionally multiplied with a learning rate η, which is either reduced by the number of training iterations performed or scales with the current error values.

There are various possible conditions for concluding the training, so-called *stopping criteria*, such as a set number of training iterations, using all training data a set number of times, or getting an error lower than a set value for a set number of feed forward iterations (Prechelt, 1998, p. 763). Summed up, the necessary hyperparameters to set for an ANN are the number of layers, the number of neurons per layer, the activation function(s), the learning rate, the number of training iterations before weights are updated (the so-called *batch size*, Abraham, 2005, and the *stopping criteria*.

2.2 Trustworthiness and Accountability

It seems obvious that the abstract concepts of trustworthiness and accountability are somehow related to each other, but it is challenging to grasp that relationship. To get there, the term *trust* needs to be defined first of all. While there are various definitions and theories revolving around that term, the following elaborations are based on the review about trust provided by Kaminski, 2017 (Section 2.2.1). Section 2.2.2 explains the term *accountability* and bridges the gap to the term trustworthiness.

2.2.1 Trustworthiness

There are many models of trust that imply very different understandings of the concept. Kaminski, 2017 deals extensively with different philosophies of trust and categorizes them into multiple, mutually exclusive types of understanding of trust. It is not clear what understanding of trust underlies the European goal of trustworthy AI. However, the focus of the AI Act on conformity assessments suggests an understanding of trust that Kaminsky refers to as 'trust as a decision' (Kaminski, 2017). Within this understanding of trust, a distinction can be made between (at least) two perspectives:

The first perspective is an evidential view of trust: 'I trust in a system because there is evidence that it has the properties that are important for me'; for example, it performs sufficiently well in terms of test results regarding its reliability. This perspective requires trusting the provided evidences, which again requires evidences whose source is trustworthy. This recursive conflict cannot be solved by the evidential view alone (Lahno, 2002, p. 177).

The second perspective is an assurance view of trust: 'I trust a system because a person is sufficiently confident that it is "good" enough for them to agree to be accountable for it'. This person does not want to face any personal or legal consequences; therefore, they ensure that the system lives up to its expectations. So this s rather trust in an actor and the process behind it than trust in the actual system (Moran, 2006, Section *Assertion as Assurance*).

2.2.2 Accountability

The demand for more accountability in ADM systems is encountered frequently. On the one hand, there are several mentions in current regulatory proposals and laws. For example, Art. 17 para. 1 lit. m AI Act (proposal) demands: '*Providers of high-risk AI systems shall put a quality management system in place that ensures compliance with this Regulation. That system shall be documented in a systematic and orderly manner in the form of written policies, procedures and instructions, and shall include at least the following aspects: [...] (m) an accountability framework setting out the responsibilities of the management and other staff with regard to all aspects listed in this paragraph*' (European Commission, 2021). The ethics guidelines for trustworthy AI provided by the HLEG-AI list accountability as one of seven key requirements for trustworthy AI (High-Level Expert Group on AI, 2019a). They state that accountability requires auditability, minimization and reporting of negative impact, trade-offs, and redress. Besides, the HLEG-AI states that '*[the requirement of accountability] necessitates that mechanisms be put in place to ensure responsibility and accountability for AI systems and their outcomes, both before and after their development, deployment and use*'. Art. 5 para. 2 GDPR[11]

[11] European General Data Protection Regulation.

states that *'The controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1 ('accountability')'*[12].

At the same time, a lot of technical standards refer to accountability. Various ISO standards, such as ISO/IEC 2382:2015[13], ISO/IEC TR 24028:2020[14], and ISO 25010[15] define accountability as a *'property that ensures that the actions of an entity may be traced uniquely to that entity'*. ISO 26000[16], however, defines accountability as a *'state of being answerable for decisions and activities to the organization's governing bodies, legal authorities and, more broadly, its stakeholders'*.

The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems (A/IS) lists accountability as a goal, saying *'A/IS should be adopted in a legal system only if all those engaged in their design, development, procurement, deployment, operation, and validation of effectiveness maintain clear and transparent lines of responsibility for their outcomes and are open to inquiries as may be appropriate'*[17].

According to IEEE Std. 7000[18] Annex F, the main difference between responsibility and accountability is that someone who is responsible has to face legal sanctions that may follow.

Most of these demands for accountability explain only very vaguely what is actually to be achieved. Sometimes, there is no distinction between accountability and responsibility. This can easily be attributed to the number of definitions in the field of accountability theory, which differ from each other in detail (see, e.g., Kacianka et al., 2017, p. 9; Kohli et al., 2018, or Ibrahim et al., 2021). Recent scientific work on AI often refers to the accountability explanations provided by Maranke Wieringa, who maps the definition provided by Mark Bovens specifically to the field of AI (Wieringa, 2020). Bovens' definition says: Accountability is *'a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgment, and the actor may face consequences'*[19] (Bovens, 2007, p. 452). Mulgan, 2000 elaborate on the development of the meaning of the term 'accountability' over the course of time. The 'core sense' of accountability they explain matches the definition of Bovens. Wieringa points out

[12] Art. 5 para. 1 GDPR demands lawfulness, fairness, transparency, purpose limitation, data minimisation, accuracy, storage limitation, integrity and confidentiality of personal data.

[13] Information technology - Vocabulary.

[14] Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence.

[15] Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.

[16] Guidance on social responsibility.

[17] https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead1e_law.pdf, last accessed on May 03, 2023.

[18] IEEE Standard Model Process for Addressing Ethical Concerns during System Design.

[19] Note that this definition is consistent with IEEE Std. 7000, as it does not necessarily speak of legal consequences. At the same time, it does, for example, contradict the definition provided by Ibrahim et al., 2021, p. 2. They define accountability *'as a property of a system that helps to identify the causes of (unwanted) events related to a quality attribute. The process of enabling accountability entails developing system's (forensic) capabilities in identifying miss-behaving parties responsible for specific violations'*. Neither Bovens nor Wieringa limit accountability to 'unwanted events'. There are more contradictions in the details of their respective elaborations.

that in the context of AI development, several actors as well as several forums with different responsibilities and interests might be relevant in different phases of the software development process.

At this point, the connection to the assurance view of trust as a decision becomes clear. A system is considered the more trustworthy the more the individual forums can rely on the accountable actors facing consequences should they arrive at a negative judgment. The process of posing questions and passing judgment, in turn, is reminiscent of the evidential view of trust. Depending on their respective authority and competence, the forums can examine a system to see whether it has the properties that are relevant to them. The results of these examinations constitute the evidences. The evidences can also be provided vicariously by other forums, for example, those that have more authority or competence than oneself (e.g., NGOs, technical experts, legal bodies, certification bodies, etc.). Appraising the trustworthiness of these other forums is particularly important, so evidential view and assurance view come into play at the same time.

For the remainder of this thesis, accountability is understood as explained by Bovens and Wieringa (see Definition 8).

Definition 8 (Accountability)

According to Bovens, 2007, accountability is a relationship between an actor and a forum, in which the actor has an obligation to explain and justify their conduct. The forum can pose questions and pass judgment, and the actor may face consequences.

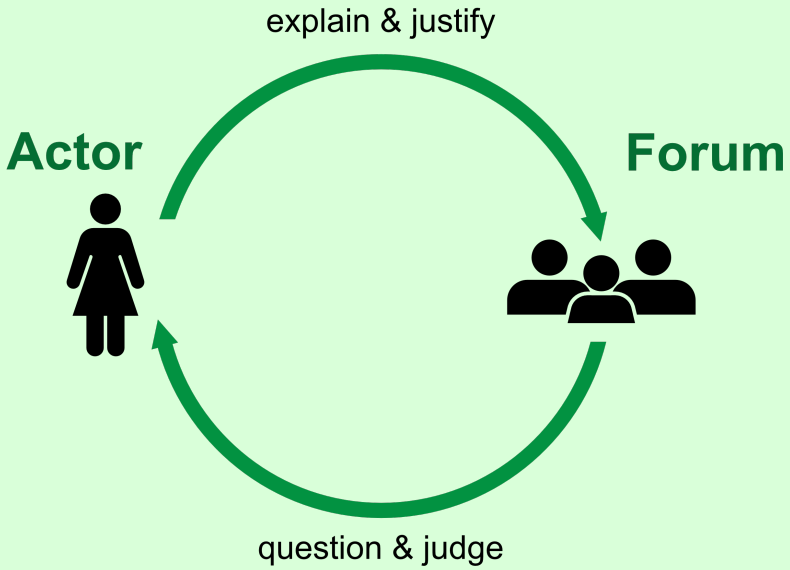


Figure 2.9: Visualization of the accountability process according to Bovens, 2007, p. 450.

2.3 Bias, Discrimination, and Fairness

Outside of expert groups, ADM systems are often associated with more objectivity and are therefore supposed to be less *discriminatory* than human decision makers. The idea behind this seems plausible. Human decisions are prone to various biases (Hilbert, 2012) and can be influenced by external factors (e.g., Eren and Mocan, 2018[20]). Awareness of such influences may help to make less subjective decisions, but cannot prevent them altogether. ADM systems, on the other hand, base their decisions solely on the given (input) information and their decision structure. At this point, however, it is often not taken into account that the decision structure was also chosen by humans and can thus incorporate *bias* in a roundabout way. In ADM systems that build their decision structure based on training data (ADM systems based on an ML component), there is human subjectivity in the generation and selection of the data on which the system is trained.

Therefore, it can be inferred that ADM systems are also unavoidably biased. Still, there is evidence that at least sometimes, they can also be less biased than human decision makers, despite the fact that their decision-making structure is shaped by human decisions (e.g., Gates et al., 2002; Kleinberg et al., 2018).

In addition, a system trained on historical data is able to make the bias in the data visible to everyone, which can be used as a basis for taking action against it. Nevertheless, different people also make judgments with different biases that might cancel each other out, at least to some extent. An AI-based system can be used for a large number of people and apply the same bias over all of them.

In order to implement measures against these risks, the basic terms *bias*, *discrimination*, and *fairness* need to be clearly understood. Unfortunately, the underlying concepts are understood differently across disciplines and sometimes even within disciplines. To avoid misunderstandings in the remainder of this thesis, these terms are introduced below.

2.3.1 Bias

There are many ways to define bias; it has no universal definition within the scientific community (e.g., Delgado-Rodriguez and Llorca, 2004; Ntoutsi et al., 2020; Steineck and Ahlbom, 1992). In general, the term is used to denote a *deviation from a standard* (Danks & London, 2017). In the context of ADM systems, it is usually a statistical standard that is being discussed, so the term bias is used in this thesis to refer to statistical bias. By this definition, a bias can also be an actual but skewed distribution. In debates about discrimination risks and fairness, bias is often understood

[20] Eren and Mocan, 2018 show that judges in Louisiana who got their undergraduate degrees at Louisiana State University (LSU) made stricter judgments in the days after the LSU football team suffered a loss.

more specifically as a deviation that does not represent the actual or desired distribution and is therefore seen as unfair to a particular person or group (Angwin et al., 2016; Datta et al., 2015; O'Neil, 2016).

In a supervised learning approach, the goal is to find a statistical correlation between input information and a label, which can also be considered as intentionally preserving and replicating the bias in the data towards that label (Danks & London, 2017).

If an ADM system is to be used to decide, for example, which candidate should receive an invitation to a job interview, feedback about the attendees' background (e.g., school grades, work experience, employer references, etc.) is used to automatically estimate their future performance (see Example 2, p. 21). Decisions based on such information are biased toward some type of competence attestation. This bias is thus intentional, socially desired, and accepted.

2.3.2 Discrimination

There is a range of attributes according to which a society explicitly does not want biased decisions. These attributes are called *protected*, or *sensitive*, attributes. *Discrimination* refers to an unjustified unequal treatment of individuals on account of protected attributes (Romei & Ruggieri, 2014). Which attributes are considered protected in a society depends on the respective culture or legislation. Whether a judgment is considered unjustified depends on the respective context. While treating job applicants differently based on their gender would be considered illegal, at least in Western societies, it may be fine to restrict entry to a privately owned shop, for example, to women only.[21] There are even cases in which differentiation by gender is useful or even necessary due to physiological differences, for example, in the context of medication.

Table 2.1 lists some examples of German laws and European directives that deal, among other things, with discrimination, and shows which attributes are affected in each case.

In this context, the difference between supervised and unsupervised learning becomes important. If a protected parameter is explicitly taken into account in the decision-making structure of an ADM system (direct discrimination), this can constitute legal evidence of discrimination. If,

[21] See, for example, Jasper von Altenbockum, *NRW verliert seinen letzten Frauenbuchladen*, December 17, 2011, Börsenblatt article, https://archive.ph/20120804175647/http://www.boersenblatt.net/466556/#selection-743.0-743.3, last accessed on June 20, 2023.

[22] Basic Law ('Grundgesetz' in German). The general principle of equality (Art. 3 No. 1 BL) requires the legislator to treat substantially equal things equally and substantially unequal things unequally in accordance with their nature (Order of June 28, 2022 - 2 BvL 9/14, Reason C.II.1.a) (68)). This principle applies beyond the explicitly mentioned attributes.

[23] General Act on Equal Treatment ('Allgemeines Gleichbehandlungsgesetz' in German).

[24] General Data Protection Regulation.

[25] European Charta of Fundamental Rights.

[26] From a biological point of view, there are no human 'races' (Templeton, 1998). In 2020, the German Parliament held a debate about replacing this term in German legislative texts (Deutscher Bundestag, 2020). However, due to disagreement over the most appropriate alternative wording, a decision was postponed indefinitely.

[27] According to Art. 9, §3 BL.

Table 2.1: Protected attributes based on various German laws and European directives, according to K. A. Zweig et al., 2021, p. 240 who extend Orwat, 2019, p. 25.

Feature	Art. 3 BL[22]	§ 1 GAET[23]	Art. 9 GDPR[24]	Art. 21 EU FR Charta[25]
Ethnicity / race[26]	✓	✓	✓	✓
Country of origin	✓	✗	✗	✗
Gender	✓	✓	✗	✓
Language	✓	✗	✗	✗
Political attitude	✓	✗	✓	✓
Religion	✓	✓	✓	✓
Disability	✓	✓	✗	✓
Age	✗	✓	✗	✓
Union membership	✓[27]	✗	✓	✓
Genetic traits and health condition	✓	✗	✓	✓
Physical characteristics	✗	✗	✓	✓
Sexual orientation	✗	✓	✓	✓

however, the decision-making structure has been trained in an unsupervised way, it might not be possible to see or understand which specific parameters are taken into account. This means that such systems can potentially escape legal action, despite discriminatory behavior (Haag, 2022, p. 133).

To differentiate between protected and unprotected parameters of an input vector X_A, $X = (X_1, \ldots X_n)$ are considered to denote the unprotected attributes and $A = (A_1, \ldots A_n)$ the protected attributes, which means $X_A = (X, A) = (X_1, \ldots X_n, A_1, \ldots A_n)$ (see Definition 9).

Definition 9 (Input and Output Information – Extension of Definition 2, p. 13)

A data point may be divided into input information $X_A = (X_1, X_2, \ldots, X_n, A_1, A_2, \ldots, A_n)$ and output information $Y = (Y_1, Y_2, \ldots, Y_m)$. To avoid confusion with the output information produced by a model, the output information in the data is also called a label.[28] The goal of a supervised learning process is to deduce a decision model based on the correlations between input information and label(s).

The input information of a data point may be further divided into protected information $A = (A_1, A_2, \ldots, A_n)$ and unprotected information $X = (X_1, X_2, \ldots, X_n)$. If the input vector does not contain any protected information, X_A equals X.

An input is also the information that is fed into an ADM system to produce an output $\hat{Y} = (\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_m)$ based on the learned model.

If no protected attributes are available, it seems reasonable to assume that an algorithm cannot take them into account. However, discrimination can hide behind unprotected, seemingly acceptable, parameters, so-called *proxy variables* (indirect discrimination). Such proxy variables (see Definition 10) have sometimes led to social problems or even scandals in the past (see Example 3), so a static (and especially the current) list of protected attributes may no longer be sufficient. Also, in order to test for discriminatory bias, it is necessary for information about these protected attributes to be collected; otherwise, it is impossible to check their influence on a decision (Hoffmann et al., 2022a; Žliobaitė & Custers, 2016).

Definition 10 (Proxy Variable)

A *proxy variable* is an explicitly specified variable that implicitly provides information about another, not explicitly specified variable because of some sort of underlying correlation (Wickens, 1972). For example, shoe size is a proxy variable for gender. Even if the gender is not explicitly specified, the shoe size can be used to infer the gender with some degree of certainty.

Example 3 (Admission Post-Bac)

The French ADM system for assigning university places, APB (Admission Post-Baccalaureat), took the current residence of prospective students into account via their zip code. For overloaded courses, the algorithm first prioritized those applicants who had graduated from the same school district; personal preferences were prioritized second.[29] The primary selection by place of residence meant that students from Paris had higher chances of being able to study at one of the prestigious universities in and around Paris. In contrast, students from rural regions had to put up with considerable disadvantages (Frouillou, 2016). Since housing in the area around Paris is generally much more expensive than in more rural regions, the place of residence is a proxy for financial status in this case.

For the remainder of this thesis, discrimination refers to a bias toward any legally protected attribute in a given context, even if it is only by proxy.

2.3.3 Fairness

Notably, anti-discrimination laws do not define precise measures that should govern ADM systems. This is because jurisprudence is (at least in most cases) based on an assessment of individual cases. It is a process in the course of which a human being can take all relevant factors into account, weigh them, and decide whether some sort of unequal treatment is to be considered *unjustified*.

However, due to the increasing use of ADM systems, decisions are made about countless people within a short period of time. In this process, discriminatory behavior is often subtle and hardly recognizable. This makes it necessary to support the detection of discriminatory behavior or even prove it by calculating values that somehow operationalize the extent of discrimination (Wachter et al., 2021, p. 5). To evaluate the extent to which an ADM system is (un)biased or (non-)discriminatory toward certain groups of people, so-called *fairness measures* can be computed.

The meaning of the term fairness in the context of ADM systems is highly controversial. Fairness as an ethical principle is often abstract and expresses an idea of justice that is not legally regulated (Velasquez et al., 1990). It might address a broad field of subtopics, such as (non-)discrimination, accessibility, coherence, inclusion, integration, diversity, revisability, and many more. Potentially, everyone has their own idea of

[29] According to an article in Le Monde: *Admission post-bac, l'algorithme révélateur des failles de l'université*, by Séverin Graveleau (June 01, 2016), https://www.lemonde.fr/campus/article/2016/06/01/admission-post-bac-l-algorithme-revelateur-des-failles-de-l-universite_4929949_4401467.html, last accessed on March 15, 2023.

what fairness means, especially in the context of a specific application (see Section 4.2). In the field of computer science, fairness usually refers to the effort to measure the extent of discrimination by an algorithmic system inversely (i.e., the 'non-discrimination').[30] This is the foundation of the term *fairness measure*. The definition of fairness has developed in computer science parallel to an ethical-philosophical understanding. Thus, this understanding of fairness does not cover all aspects of fairness as an ethical principle. Fairness measures can be divided into *group fairness measures*, *individual fairness measures*, and *counterfactual fairness*, which can be argued to belong to any of the two groups.

2.3.3.1 Group Fairness Measures

Group fairness measures (also called *distributive fairness measures*) require two sensitive groups to be treated equally or similarly. What 'equal or similar treatment' means, however, can be expressed in different ways. One option is to define a quality measure and require similar values of that measure for both sensitive groups (see Definition 11). Quality measures evaluate the prediction quality of a decision-making system based on the confusion matrix (see Definition 12).

Definition 11 (Quality and Fairness Measures)

Quality measures evaluate the number and severity of wrong classifications or wrong scoring/ranking results. *Fairness measures* assess whether certain groups of people are more often affected by errors than others (which, for example, can be computed based on quality measures) or whether the system's output distribution with regard to subgroups compared to each other is imbalanced (Verma & Rubin, 2018).

[30] The HLEG-AI refers to this understanding as *substantial fairness*, as opposed to procedural fairness, which '*entails the ability to contest and seek effective redress against decisions made by AI-based systems and by the humans operating them*, (High-Level Expert Group on AI, 2019a, pp. 12-13).'

Definition 12 (Confusion Matrix)

The confusion matrix is a table used to evaluate the quality of a decision-making system. It contains four entries: the numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) (see Figure 2.10). These entries can be retrieved from a ground truth (see Definition 3, p. 13). The values in the matrix can be used to compute various quality measures (see Table 2.2).

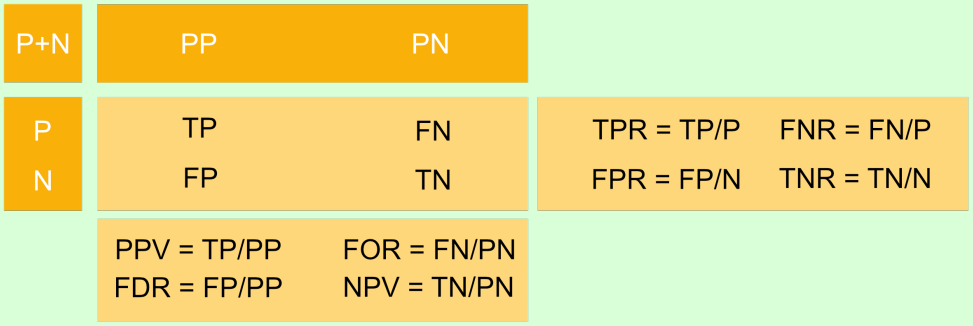


Figure 2.10: P is the overall number of positive instances ($TP + FN$). N is the overall number of negative instances ($FP + TN$). PP is the overall number of instances classified as positive ($TP + FP$). PN is the overall number of instances classified as negative($FN + TN$).

Various group fairness measures are based on such quality measures, as shown in Table 2.3. This list shows only some of the most common fairness measures based on quality measures (Tharwat, 2020, pp. 170-173). It is relatively easy to construct new fairness measures based on the comparison of quality measures between two subgroups. It is not even necessary for the quality measurement to be based on a quality measure resulting from the confusion matrix. One could also compare how often people from different subgroups complain about their assessment or how long it takes to process information from people from different subgroups. Furthermore, both quality and group fairness measures can take special circumstances into account. For example, it is not always useful to compare two complete subgroups (e.g., men and women applying to a company). Rather, it may be more useful to limit the comparisons by conditions (e.g., only applications for entry-level jobs and only applications for manager positions, each separately). In this way, more subtle questions can also be examined in the context of fairness. For example, it could be examined whether single mothers with an academic degree who are younger than 30 years are at a disadvantage compared to married mothers with an academic degree who are younger than 30 years, at least if the necessary information is available. In any case, it is important that the

Table 2.2: Common quality measures based on the confusion matrix listed by Tharwat, 2020, pp. 170-173.

Quality measure	Formula
Accuracy (ACC)	$\frac{TP+TN}{TP+TN+FP+FN}$
True positive rate (TPR), Recall or sensitivity	$\frac{TP}{TP+FN}$
True negative rate (TNR) or specifity	$\frac{TN}{TN+FP}$
False positive rate (FPR) or fall-out	$\frac{FP}{FP+TN}$
False negative rate (FNR) or miss rate	$\frac{FN}{FN+TP}$
Positive predictive value (PPV) or precision	$\frac{TP}{TP+FP}$
False discovery rate (FDR)	$\frac{FP}{FP+TP}$
Negative predictive value (NPV)	$\frac{TN}{TN+FN}$
False omission rate (FOR)	$\frac{FN}{FN+TN}$
Matthews correlation coefficient (MCC)	$\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

chosen quality measure be meaningful in the context of the fairness assessment (see Example 4, p. 38). In addition, the chosen quality measure may need to deal with unbalanced data. Unbalanced data are data sets in which one subset is significantly larger than the other. Many measures are unsuitable for use on unbalanced data sets because they implicitly weigh the consideration of subgroups by their size (e.g., accuracy). There exist measures that are explicitly designed for use on unbalanced data sets (e.g., MCC, Chicco and Jurman, 2020).

There is also at least one group fairness measure that is not based on equal quality measure values but only on the output distribution of a system; this is *statistical parity* (Dwork et al., 2012). It requires that the probability of assigning a data point to the positive class and the probability of assigning a data point to the negative class be the same for each sensitive group, independent of any ground truth. The measure is also referred to as *independence*, as it requires a decision independent of sensitive values. However, the term can easily be confused with the statistical independence of values.

Table 2.3: Common fairness measures based on quality measures. Some measures have been introduced in multiple scientific disciplines, so they have multiple names and different formalizations that are, in part, difficult to map to quality measures. The mapping of each fairness measure is explained by Haeri Amir et al., 2023.

Fairness measure	Requires equality of	References
Overall accuracy equality	ACC	Berk et al., 2021
Separation	TPR and FPR	Hardt et al., 2016
Conditional procedure accuracy		Barocas et al., 2017
Equalized odds		Berk et al., 2021
Equal opportunity	TPR	Barocas et al., 2017
Error rate balance	FPR and FNR	Chouldechova, 2017
Sufficiency	PPV and FOR	Barocas et al., 2017
Conditional use accuracy	PPV and NPV	Berk et al., 2021

Example 4 (COMPAS)

The Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) system, developed by Northpointe Inc. (now Equivant), rates criminals on a scale of 1-10 (called percentiles), with a high score representing a high risk of re-offending. This information is presented to judges to help them determine a sentence and whether it should be suspended (EPIC, 2020). The development of such systems was requested by the American Civil Liberties Union (ACLU), among others, in 2011 to reduce the consequences of human bias in the justice system (Chettiar & Gupta, 2011, p. 9), as the imprisonment rate of black Americans is around five times that of white Americans (Nellis, 2021). In 2019, however, the same organization (and many others) spoke out against the use of such systems, as some analyses have shown that such systems perpetuate and even reinforce existing patterns of discrimination (Leadership Conference on Civil and Human Rights, 2018).

To prove that the system does not discriminate against black people, Northpointe tested whether the ratio of recidivists to all criminals within a percentile is approximately the same for different ethnicities. This is a special form of statistical parity, which they call *predictive parity* (Dieterich et al., 2016). In this test, they could not find any discrimination.

Angwin et al., 2016, on the other hand, computed the fairness of the COMPAS system based on the quality measures FDR and FOR between white and black people, with the result that the assessment of black people had a significantly higher FDR and the assessment of white people a significantly higher FOR. For COMPAS, a high FDR indicates that among the people who are classified in the high-risk category, a proportionately large number are wrongly given this rating. A high FOR, on the other hand, means that among the people who are classified in the low-risk category, a proportionately large number are wrongly given this rating (see Table 2.2). So there are errors in both directions, but the errors are disadvantageous for black people and advantageous for white people. This result supports the assumption that COMPAS is a discriminatory system. At least as of 2020, COMPAS was still used in five U.S. states (EPIC, 2020).

2.3.3.2 Individual Fairness Measures

Individual fairness measures require similar individuals to be treated similarly (Dwork et al., 2012). The similarity needs to be computed based on some sort of distance measurement D. It might be necessary to formulate two distance measurements, one for measuring the similarity of

system outputs $D_{\hat{Y}}$ and one for measuring the similarity of system inputs D_X. Formalized, *individual fairness* means $D_{\hat{Y}}(\hat{Y}(X_i), \hat{Y}(X_j)) \leq D_X(X_i, X_j)$, where $\hat{Y}(X_i)$ is the system output $\hat{Y}$ for the system input X of individual i. The more attributes are considered to compute a decision, the more difficult it is to define a proper distance measurement. Consider multiple data instances that are similar in one half of their attributes but distinct in the other. Whether the system output for all of these instances is supposed to be equally distant or not cannot be answered in general terms. The decision for a specific distance measure might be very subjective. As the computation is not based on any ground truth, *individual fairness* does not require any information about the target output Y or the confusion matrix.

2.3.3.3 Mediating Between Group Fairness and Individual Fairness

The scientific literature refers to more than 30 fairness measures and a great number of variations of them, which can be combined in different ways. Fulfilling all fairness measures equally is only possible in constructed examples because a large number of fairness measures contradict each other in their aims (Berk et al., 2021, p.17). The more complex the fairness measures being compared, the more difficult it is to identify conflicts (see, for example, Barocas et al., 2017, Chapter 3, for some relatively simple conflicts between *group fairness measures* and Dwork et al., 2012, for the general conflict between *individual fairness* and *group fairness*).

In Haeri Amir et al., 2023, we propose a novel approach to address the conflict between *group* and *individual fairness*. As already stated, conditions can be formulated to take into account only very specific subgroups. By dividing groups into smaller ones based on carefully selected conditions, and comparing the equality of the quality measure, it is possible to go from *group fairness* to *individual fairness* step-by-step, introducing a hierarchical perspective on fairness (see Figure 2.11).

Finding an appropriate definition of similarity for *individual fairness* often poses a problem, as such decisions are nearly always subjective. An alternative approach is to define conditions that split the data into fine-grained subgroups in which very high fairness measure results should be expected (Haeri Amir et al., 2023). It may not always be possible to find such specific (and reasonable) conditions which result in only separate individuals remaining in the dataset. But this is not necessary to satisfy the statement of individual fairness measures: 'How similarly are similar individuals treated'.

Suppose that for a job-hiring system, the appropriate definition of fairness is the equality of *accuracy* for different groups, and gender is the only protected attribute to worry about. This means that the prediction *accuracy* for all men and all women is compared, with the definition of fairness at the top of the hierarchy. The results of the groups are compared independent of the individual traits (see Figure 2.11a).

If the same definition of group equality is applied to smaller sub-groups of men and women – for example, for those with at least five years of

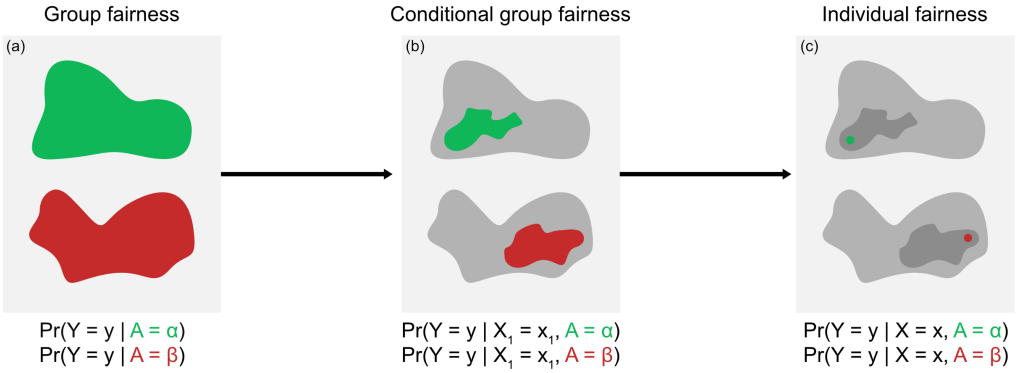


Figure 2.11: At the highest level, group fairness measures require equality of quality with respect to only two groups defined by the protected attribute. The middle level considers fairness measures computed based on subsets that are similar to each other and their equivalents with a different protected attribute, respectively. With increasing granularity, the subsets become smaller, up to the point where only a few individuals are compared with each other. This represents the lowest level in the hierarchy.

practical experience and evidence of specific expertise – it is considered a conditional group fairness measure (see Figure 2.11b).

The more conditions are applied that further reduce the number of people per group, the more the perspective shifts from *group fairness* to *individual fairness* (see Figure 2.11c).

Providing fairness results at different levels of the proposed hierarchy may provide a more complete picture of the fairness of a system. However, the idea only addresses the trade-off between *group* and *individual fairness*. Other conflicts remain and should be addressed based on the particular notions of fairness implemented through a measure.

A selection of conditions might also be tailored to the data set to intentionally hide bias in the system. Therefore, there should always be proper justification for the selection of conditions provided, together with the resulting fairness values. Such justification could be provided by so-called *Assurance Cases*, which will be thoroughly discussed in Section 4.2. Optimally, access to the system is provided to allow experimentation with other conditions in case of doubt, as described in Chapter 3.

The suggested approach for a hierarchical perspective of fairness may be susceptible to *Simpson's Paradox* Simpson, 1951 (see Definition 13).

Definition 13 (Simpson's Paradox)

Simpson's Paradox refers to the phenomenon that subgroups show a trend in results that disappears or even reverses when all subgroups are considered together or vice versa (Pearl et al., 2000). It can occur when a parameter can have both positive and negative effects on a result or when it is a confounding parameter. The paradox can be resolved by removing confounding and causal parameters, but the lack of knowledge about what these parameters are is one of the reasons for using ML approaches in the first place.

Following this general framework, all group fairness measures described in Table 2.3 can be used to assess the fairness of a system at different levels of granularity.

2.3.3.4 Counterfactual Fairness

A system satisfies *counterfactual fairness* if the system output $\hat{Y}$ is the same for any system input and its respective *counterfactual*. The counterfactual of an input is an identical twin that differs in only one attribute (Pearl et al., 2009, p. 120); in the context of fairness evaluations, a sensitive attribute:

$$\Pr(\hat{Y} = \hat{y} \mid X = x, A = \alpha) = \Pr(\hat{Y} = \hat{y} \mid X = x, A = \beta) \tag{2.9}$$

This means that *counterfactual fairness* is based on the individual comparison of pairs of data points and can thus be considered a special form of *individual fairness*.

A different notion of *counterfactual fairness* tries to respect the influence of a changing sensitive attribute on correlating insensitive attributes (Kusner et al., 2017). In this case, *counterfactual fairness* cannot be understood as a special form of *individual fairness* anymore. In Hauer, Kevekordes, and Haeri, 2021, we provide an easy-to-understand explanation of this kind of *counterfactual fairness* and further literature references.

2.3.3.5 Further Challenges Regarding Fairness Measures

Independent of the chosen fairness measure, there is always the question of whether the measure is calculated using real data, artificially generated data, or real but carefully selected data. Here, too, a meaningful decision must be made. If a fairness measure is calculated using real data, a statement is made about how fair the system is in real use, and whether it can compensate for any unfairness that may already exist in the input. In the case of artificial or selected data, which, for example, correspond to

a normal distribution with regard to selected parameters, the fairness of the decision-making structure itself is assessed. However, in the case of a fair decision-making structure and unfair input data, unfairness may still be preserved and reflected.

RQ 1

What considerations are relevant when selecting fairness measures?

Answer: The choice of an appropriate fairness measure mainly depends on the intended objective of such a measure.

Group fairness measures compare groups of people who are similar to each other in at least one attribute. For **group fairness measures based on a quality measure**, it is particularly important to choose a quality measure that is relevant to the targeted statement. This can be a quality measure based on the confusion matrix or some other form of quality assessment. If system inputs are expected from groups of unequal size (with regard to the attribute for which fairness is to be determined), a quality measure must be selected that is suitable for calculation on the basis of unequal group sizes.

Group fairness measures that are not based on a quality measure compare the output distribution between two groups independent of the inputs. This makes it possible to measure the extent to which the entire socio-technical system, i.e., not only the decision-making system itself but also the social background of the input data, meets an equal treatment requirement. A low value of such a measure does not necessarily mean that the system makes discriminatory decisions, but may indicate an inability of the system to compensate for discrimination in the input data.

Individual fairness measures provide information on the extent to which similar persons are treated similarly. If such a measure indicates a low level of fairness, targeted pairwise comparisons can be used to systematically identify the specific characteristics that lead the system to make unfair decisions.

In both cases, real data as well as artificially generated data can be used for the calculation. With artificially generated data, there is a risk that the distributions of certain properties will differ from real distributions and, in addition, that a change in the real distributions over a longer period of time cannot be reflected. In exchange, hypothetical ideal states can be postulated in the data that are not distorted by unequal treatment in preceding systems. This is particularly useful if the fairness of the decision-making system itself is to be assessed independent of unequal treatment already reflected in the input data.

Counterfactual fairness can be used to examine whether a person would be treated differently if only a certain (protected) attribute

were different. The significance of *counterfactual fairness* is controversial. Since counterfactual pairs can hardly be found in real data (see, e.g., Bertrand and Duflo, 2017, p. 318), at most half of such measures can be calculated on the basis of real data. The counterfactual duplicates do not necessarily represent realistic data points, since a change in a single attribute in reality may result in further changes in other attributes, whether through direct or indirect dependencies. A variation of *counterfactual fairness* that attempts to take precisely this circumstance into account has not yet been sufficiently studied to be able to make a statement about it.

If multiple fairness statements are pursued, they are likely to contradict each other to some extent. The same applies to fairness measures that reflect these statements. Conflicts between group and individual fairness statements can be resolved at least partially by calculating group fairness measures on the basis of homogeneous subgroups, thereby approximating the statements of individual fairness measures. An investigation of the extent to which this consideration is suitable in practice to counter conflicts between group and individual fairness measures is still pending.

Despite the flexibility with which fairness measures can be defined, they (currently) cannot be sufficient to establish or disprove discrimination in our process-oriented jurisprudence. However, they can support the decision-making process. In Hauer, Kevekordes, and Haeri, 2021, we present several fairness measures, their implications, and their relevance to the assessment of discrimination in human resources in the context of legal disputes.

Communicating fairness measures to non-experts can be difficult, as a good basic understanding of mathematics and statistics is necessary. It is therefore not sufficient to state the choice of fairness measures and the values achieved in each case, but they must also be explained in a way that is appropriate for the affected party. For many basic measures, there are sufficient scientific foundations to be able to provide explanations appropriate for the target group. Such explanations may serve as a good basis for argumentation in the event of an audit (e.g., certification or legal dispute; see Chapter 4 for a thorough discussion) and for being able to justify oneself to a customer or affected party (e.g., as a contribution to CDR; see Section 6.2 for more details).

Although this thesis is about accountability as a whole, fairness, as an operationalization of non-discrimination, is often understood as an integral part of accountability (as in Art 5 para. 1 and Art. 5 para. 2 GDPR) and is therefore explicitly considered at several points in this work.

Chapter 3

Transparency and Inspectability Mechanisms for Achieving Accountability

Beyond providing only a narrow definition of accountability, legislation and standardization literature hardly goes into detail about what implementation could look like and who exactly should be accountable for what. To tackle this problem, K. A. Zweig et al., 2018 introduced the long chain of responsibility to model modular parts of an ADM system development process and their relationships to each other (see Figure 3.2, p. 49). At least at every connection of these modules, the responsibilities might be unclear and should be clarified. The model thus aims to support the frequent demand for accountability in ADM systems. However, it does not address the question of how accountability can be achieved. To meet Bovens' definition of accountability (see Definition 8, p. 28), four conditions need to be met (Bovens, 2007):

Condition 1: It is possible to analyze the system under question.

Condition 2: It is clear which actors are to be held accountable.

Condition 3: It is clear which forums hold the actors accountable.

Condition 4: There are processes that result in consequences for the actors based on the forums' judgments.

The following sections review the possibilities of satisfying these conditions. Section 3.1 introduces the possible mechanisms in each phase of the long chain of responsibilities (condition 1). Section 3.2 elaborates on the relevant actors to be held accountable (condition 2). As to whom an actor is accountable to is very much situation specific, might be prescribed by law, and depends on the rights of actors to keep certain decisions and systems private, possible forums are discussed in Section 3.3 (condition 3). Possible consequences also depend on the specific forums and their judgments, which is why these are discussed in Section 3.4 (condition 4).

3.1 Possibilities to Analyze the System

We[1] extend the long chain of responsibilities to represent the entire software development process, from requirements engineering to evaluation

[1] In this Chapter, 'we' and 'our' refers to the authors of Hauer, Krafft, and Zweig, 2023.

in the field. Furthermore, we identify specific mechanisms[2] per module that promote the establishment of accountability based on the definition of Bovens (Hauer, Krafft, & Zweig, 2023).

Possibilities to analyze the system can be split into two kinds of mechanisms, *transparency mechanisms* and *inspectability mechanisms* (Busuioc, 2021, p. 827).[3] Transparency mechanisms denote the disclosure of static information like data, processes, and results. They allow the actor(s) to explain and justify parts of their system, and the forum to judge the appropriateness of these parts based on the information provided. Rosalie, 2022, p. 7, even argue that *'transparency implies having the power to know or understand what happens to one's data and on what bases decisions are made'*; this results in individual empowerment.

However, the benefits of transparency mechanisms alone have their limits (Kroll et al., 2017, p. 638; Ananny and Crawford, 2018). The information disclosed could have been manipulated or outright made up. Without further mechanisms, the forum has no way to verify them. It has to be assumed that many projects are too complicated to be assessed on the basis of the source code. In 2014, the code of the French tax computation software was published. It was written in a self-developed programming language called 'm' and contained over 17,000 variables and approximately 1,000 functions.[4] With data-driven components (DDC), the problem becomes even more severe, as the resulting model might be inherently incomprehensible. To enable at least limited assessment in such a case, the data basis on which learning took place and the learning objective (e.g., in the form of quality measure values to be reached) are of interest instead. However, the added value is low, since only very limited conclusions can be drawn from the data and the learning objective to the model; for example, because the (random) order in which the training data is supplied has an influence on the statistical model (Lopes et al., 2017, p. 621). Moreover, the code alone cannot be used to infer its impact on a social system. Finally, disclosing certain information might permit exploiting the system. This happened, for example, in 1999, when Page et al., 1999, published the PageRank algorithm. PageRank is the method developed and initially used by the founders of Google for rating Web pages objectively and mechanically. The details of the algorithm allowed website operators to exploit how it works by, among other things, launching link farms that link to websites that should be artificially presented as more relevant than they are (Grimmelmann, 2008, p. 946). Since then, Google has no longer disclosed details on how their algorithm works.

Any mechanism that allows the forums to explore the system themselves and thus does not require relying on information provided by someone else is considered an inspectability mechanism in our work. Such

[2] The term 'mechanism' was chosen as the term 'measure' is ambiguous. It can be understood in terms of 'taking action' but also in terms of 'measuring something'. To avoid confusion, this work refers to the term 'measure' only in the latter sense.

[3] Busuioc, 2021, p. 827 states that '*While necessary, in and of itself transparency is not a sufficient condition for accountability. Accountability is closely linked to 'answerability', and a key element thereof is that of explanation or justification.*'. We chose to refer to mechanisms that provide such 'answerability' as 'inspectability mechanisms', as this term does not yet seem to be occupied by other definitions.

[4] https://github.com/etalab/calculette-impots-m-source-code, last accessed on August 12, 2023.

mechanisms can be implemented, for example, in the form of application programming interface (API for short) accesses, to enable a forum to query their own inputs to the system and evaluate the resulting outputs (Diakopoulos, 2014). Depending on the implementation, it would even be conceivable to allow a forum to incorporate intermediate outputs, for example, to examine decision branches within the system. In this way, the forum could 'pose questions' directly to the system without depending on an actor. Although inspectability mechanisms have great potential to complement transparency mechanisms, they raise problems and challenges of their own. First of all, they might allow finding ways to exploit the system even better. Being able to submit a large number of inputs and to receive the corresponding outputs could also enable a forum to reverse engineer a similar system, which could pose a risk to a company's success. Finally, maintaining security while granting access may be more difficult.

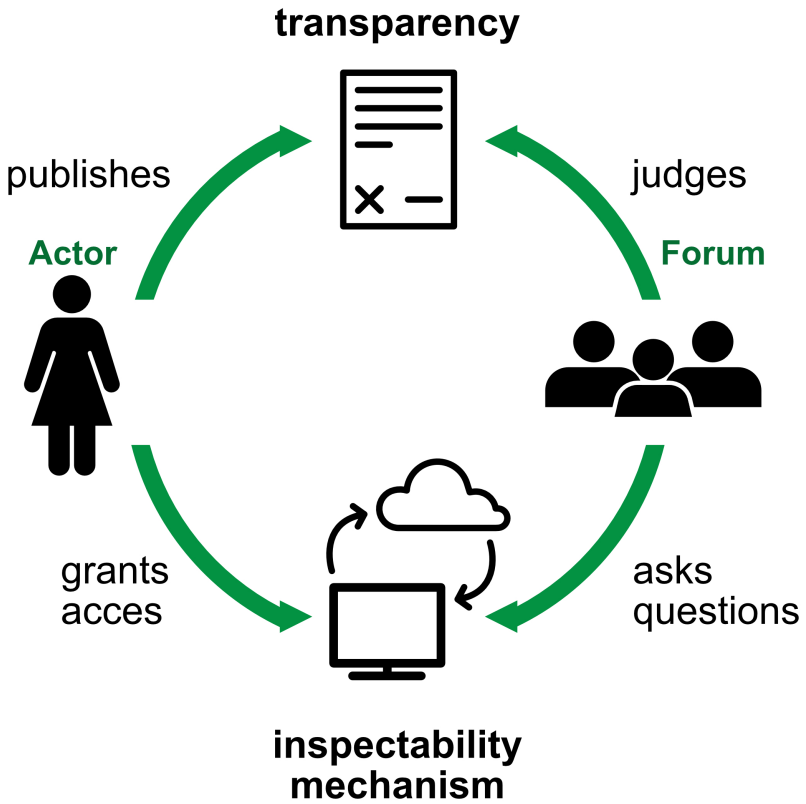


Figure 3.1: Transparency about past decisions and actions, system properties, and test results, plus access to the data and the system help to establish an asynchronous accountability process between different actors and forums that matches the synchronous accountability process described by Bovens (see Figure 2.9, p. 28, for comparison).

Transparency and inspectability mechanisms together provide the basis for almost any conceivable form of system analysis by a forum in an asynchronous and scalable fashion (see Fig. 3.1). The possibilities of implementing such mechanisms differ massively with respect to the concrete use case and the necessary effort (Kroll et al., 2016; Lepri et al.,

2018). Therefore, appropriate transparency and inspectability mechanisms cannot be determined generically for all AI-based ADM systems. They need to be determined depending on the system and its concrete application in the specific individual case. Also, different actors may be accountable to different forums in different phases of the development process, and in each phase, different mechanisms can be implemented. In the following, accountability mechanisms for each phase will be explained based on an extension of the long chain of responsibilities as provided by Hauer, Krafft, and Zweig, 2023[5] (see Figure 3.2):

- Phase A addresses **requirements engineering**. This involves information about what exactly is to be achieved with the ADM system, what the (informal) target criteria are, a benefit and risk assessment, and further information about the requirements.
- Phase B addresses the **data collection** procedure, which can be the cause of errors and biases if not carried out properly.
- Phase C addresses the **creation of the training data set**, for which preparation and cleaning steps are applied.
- Phase D addresses the **method selection**, i.e., the choice of an ML method and parameters, including the tool(s) used. The choice of an ML method and the creation of the training data set are interdependent. The developers can either choose a procedure for which the data is suitable or which requires little pre-processing, or they can design the details of the pre-processing with regard to the chosen procedure.
- Phase E addresses the details of the **learning procedure**, which may contain very different kinds of information depending on the selected ML method.
- Phase F addresses the **quality assessment**. This includes the data used for testing, the conditions under which tests are performed, and the test results.
- Phase G addresses **the system usage in an application scenario**. Here, the focus lies on application-specific requirements that were not yet available when the system was developed.
- Phase H addresses the **evaluation of the ADM system in an application scenario** for which there is a specified environment with corresponding quality requirements that may not have been known in Phase F.

The individual phases and the mechanisms that can be implemented in each of them are described and explained in detail below. This means for each phase:

1. Information is described that can be published to help forums identify responsible actors and potential problems (transparency mechanisms) and

[5] In this publication, we chose our wording very carefully. The explanations of the phases in the remainder of this section may therefore be partly very similar.

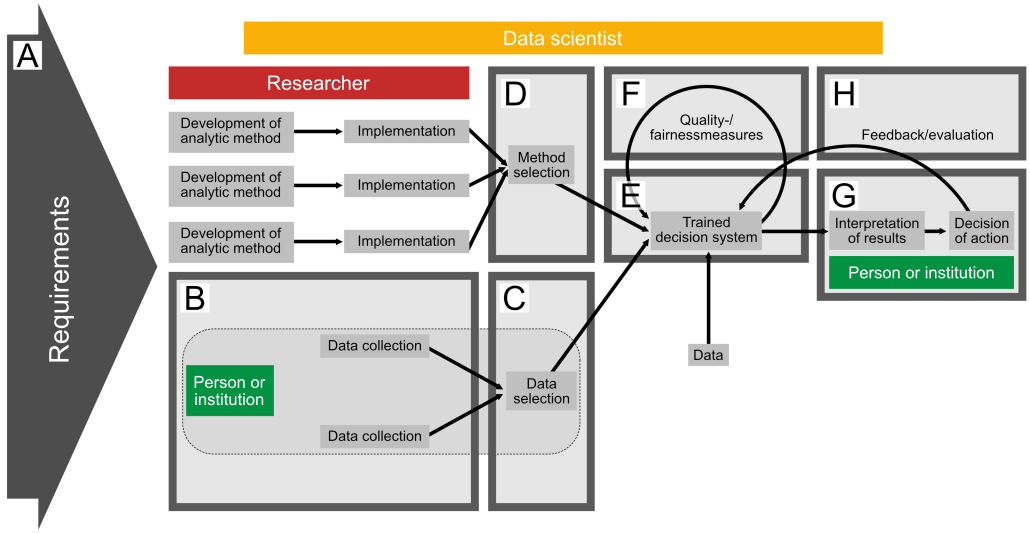


Figure 3.2: The long chain of responsibilities according to K. A. Zweig et al., 2018. Extension to cover the complete software development and deployment process provided by Hauer, Krafft, and Zweig, 2023.

2. access mechanisms are described that allow a forum to explore relevant internal information or to experimentally interact with the ADM system (inspectability mechanisms). Note that for some phases, there are no inspectability mechanisms we are aware of.

Each subsection begins with a brief description of the relevant phase. Then the respective mechanisms are discussed, including a designation for the mechanism, the relevant actors, and what kind of insight the mechanism might bring to a forum. A company's internal distribution of responsibilities is potentially more complicated than a generalized explanation could do justice to. Therefore, from an external point of view, '*actor*' initially refers only to those directly responsible, but they can forward requests to the other people involved in a particular phase. Therefore, '*people involved*' is used for the sake of simplification.

With this approach, we are making an attempt to cover as broad a spectrum of applications as possible. However, this also means that it most likely will never be possible to transfer all the concepts for creating accountability to one concrete application. This is why many formulations are abstract and list only a selection of easily understandable examples.

In the following, transparency mechanisms are labeled with Arabic numerals (A1, A2, A3, B1, etc.) and inspectability mechanisms with Roman numerals (I, II, III).

3.1.1 Phase A: Requirements Engineering

Before building a software system, it is important to understand what functions it should provide to its users (functional requirements) and what other requirements must be met, such as reliability, security, etc. (non-functional requirements). The most relevant standard regarding software

quality, ISO 25010[6], provides an exhaustive list of functional and non-functional characteristics that are relevant for determining product quality. Other requirements not listed in this (or other) standard(s) can also be divided into functional and non-functional requirements by definition, but are sometimes referred to as extra-functional requirements (Panunzio & Vardanega, 2014; Sentilles et al., 2009).

Definition 14 (Functional, Non-Functional, and Extra-Functional Requirements)

The term *'functional requirement'* refers to any requirement on the range of functions (functional completeness, correctness, and appropriateness) of a system. The term 'non-functional requirement' refers to any requirement beyond plain functionality, such as reliability, compatibility, security, or maintainability. However, it is sometimes used ambiguously. Some definitions provide a conclusive list of requirements, like ISO 25010, without mentioning some non-functional aspects, such as fairness or diversity. Other definitions understand any requirement that is per definition not a functional requirement as a non-functional requirement. To refer to the latter understanding, the term 'extra-functional requirement' is sometimes used, e.g., in Sentilles et al., 2009 and Panunzio and Vardanega, 2014.

Requirements engineering is a systematic approach to identifying, specifying, and managing these requirements. It aims to understand the needs of customers and future users and communicate them to the developers in order to deliver a satisfactory product (Glinz, 2014). It may also include a risk assessment and information on how the identified risks are to be mitigated or handled. In some cases, this might even be legally required.[7] The results of the requirements engineering process are recorded in so-called requirements documents.

In addition to the specified requirements, requirements can also contain use cases that enable both clients and developers to get a similar idea of the end product. It is particularly important for developers to know what the later operational environment will look like, since specific quality requirements may depend on it, regardless of the technology to be implemented. A recommendation system, for example, tends to have rather low requirements if it is only to be used to recommend household items. However, the exact same system, trained on other data, can also be used to recommend applicants for an advertised job (Krafft & Zweig, 2019, p.33). In this case, there is a legal requirement to provide recommendations regardless of gender, religion, ethnicity, etc. If the developers only have the task of developing a generic recommendation system, such requirements

[6] Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[7] See, e.g., the American Algorithmic Accountability Act of 2022.

can easily be overlooked.

If the application context demands explainable decisions (see Definition 15) of the ADM system, this requirement should also be made explicit. As some ML models suffer from a lack of explainability (e.g., Artificial Neural Networks), there are attempts to achieve explainability post hoc (e.g., with techniques such as LIME, see, Ribeiro et al., 2016, or Anchors, see Ribeiro et al., 2018). However, the research field of AI explainability is still evolving, and currently popular methods have some limitations (Rudin, 2019a). Moreover, there is no agreement yet regarding the circumstances under which what type of explanation is appropriate and sufficient. If explainability is required, it seems reasonable not to rely on such methods (Rudin, 2019a).

Definition 15 (Explainability)

Explainability is a vague concept, and what it means seems to depend on who requires an explanation for what reasons (Lu et al., 2019). Software developers want to understand the data-driven model (DDM) in their software product to be able to detect potential sources of errors as well as to improve the system. For example, they might be interested in what a hypothetical optimal input looks like for a given output or what training data example fits that output most, so-called *'Prototypes'* (Barbalau et al., 2020; Li et al., 2018). They might also be interested in what parts of an input or sections of a model are especially relevant for a specific output. Such explanation concepts are called 'Attribution Methods' (e.g., *'Activation Maximization'*, see Nguyen et al., 2016, *'Gradient-Weighted Class Activation Mapping'* (CAM), see Zhou et al., 2016, and *'Layer-Wise Relevance Propagation'* (LRP), see Montavon et al., 2017). However, people affected by a system potentially have great interest in understanding whether changing certain input information will affect their system output. For example, they want to know whether a different job or even a small salary increase would significantly affect their credit limit. Such an explainability concept is called 'Sensitivity Analysis' (Embrechts et al., 2003; Cortez and Embrechts, 2011). There are a large number of methods that produce different forms of explainability.[8] To what extent these actually represent explanations in the philosophical sense is an open research question (Doshi-Velez and Kim, 2017; Guidotti et al., 2018, p. 36; Broniatowski et al., 2021).

It is often assumed that transparent, traceable models, such as decision trees, are inherently explainable (Freitas, 2014). Even if a decision tree is small enough to remain comprehensible to a person,[9] recent research shows that discriminatory information in the training data can be hidden by decision trees (Wilhelm & Zweig, n.d.). Thus, decision trees do not necessarily provide a form of explanation that is helpful to a forum, and explainability approaches that emulate complex DDCs by simpler models – such as decision trees (so-called 'surrogate models', see Craven and Shavlik, 1995 and P. Hall, 2018, p. 2) – are potentially rendered useless for affected persons interested in assessing whether they have been treated unfairly. However, decision trees might still be useful to get an idea of the decision structure of the complex model.

Another important information to include in the requirements documents for a system with a DDC is whether the final product should be able to continue learning while in use. If it should be, but the developers were unaware of this and did not implement it with this sort of intended use in mind, problems are likely to arise (Kroll et al., 2017, p. 660).

There are at least three transparency mechanisms relevant to the requirements engineering phase:

A.1 Disclosure of the Application Scenarios

Disclosure of the application scenarios enables third parties to assess whether a particular ADM system is used in ways and contexts in which it is supposed to be used or not to be used. For this transparency mechanism, a distinction can be made between customized and general-purpose software. In customized software, the customers define the application scenario and help the software provider determine all subsequent requirements. If the software is not used as communicated by the customers, the software provider should not be held accountable for problems resulting from improper use. In the case of general-purpose software, the software provider is accountable for considering all possible uses and setting possibly strict requirements for the system accordingly. At best, the provider officially limits the general-purpose software to a list of application scenarios for which the requirements have been tailored, shifting accountability regarding usage for unlisted application scenarios to the user. The cost of publishing information on the application scenarios for a particular ADM system that have been considered can be estimated to be low, as a semi-structured document should be sufficient in most cases.

A.2 Disclosure of Requirements Documents

Instead of disclosing only the application scenarios, the complete requirements documents could be disclosed. Using these, a forum can investigate the system regarding multiple questions, such as (i) whether the specified requirements are adequate, (ii) how the fulfillment of requirements is to be evaluated, (iii) whether important requirements are missing, (iv) whether an appropriate risk assessment and mitigation process is in place, or (v) whether the system is to be designed in such a way that spontaneous manual intervention is possible in case of errors or problems. Publication of requirements documents also includes publication of the people (roles) involved in requirements engineering, as there are, for example, scenarios in which the requirements documents are not developed by the system's developers.

[8] See Guidotti et al., 2018, Arrieta et al., 2020, Sokol and Flach, 2020, Kraus et al., 2022 and Speith, 2022, who provide great overviews but use different taxonomies and terminologies. Miller, 2019 elaborates on different perceptions of the term 'explanation' by various social sciences.

[9] Ribeiro et al., 2016, Section 3.2 discuss this aspect briefly, but there appears to be no experimental evaluation of the conditions under which a decision tree is still considered comprehensible by a person.

Depending on how sophisticated a company's requirements engineering process is, especially in terms of producing requirements documents and their readability for different forums, publishing them hardly presents any additional effort. However, there are potential indirect costs that go beyond the effort of publishing. Requirements documents can expose trade secrets and provide insights into the inner workings of a system and a company. These, in turn, potentially reduce security, allowing hackers to discover and exploit vulnerabilities. Considering proportionality (see, e.g., European Commission, 2021, section 2.3), regulation could only require disclosure of certain parts of requirements documents. This would incur further costs for selecting and preparing the relevant content. In addition, access to requirements documents can mislead a forum into seeing abstract or unrealistic problems where none actually exist (e.g., due to a lack of background knowledge). The effort and damage that can result from unjustified accusations is a good reason for corporate actors not to disclose requirements documents.

A.3 Disclosure of the Goal of Using an ADM System

Detailed information on the goal of using an ADM system can play a major role for the forum and be published independent of the rest of the requirements information.

Suppose the goal of the *Austrian Unemployment Risk Assessment Software* (AMS algorithm, see Example 5) is to help companies find the best possible future employees based on the premise that people classified as long-term unemployed are less productive. Apart from the highly questionable assumption, it is very difficult to test such a system, since the performance of people who are not hired cannot be measured. Accordingly, there are restrictions on the ability to answer questions asked by the forum from the beginning. However, according to the software company that developed the system, the goal is to minimize the cost of training and other interventions and maximize the number of people on the labor market (Holl et al., 2018). How reliable the system's predictions and the benefits of the interventions are can be determined by having the system perform test evaluations without having an effect at first.

Example 5 (Austrian Unemployment Risk Assessment Software)

In Austria, a system for predicting the chances of integration into the labor market for currently unemployed persons, called Austrian Unemployment Risk Assessment Software (AMS algorithm), was tested. Unemployed persons were categorized into three groups: low, medium, and high chance of integration. The goal was to minimize the cost of training and other interventions and maximize the number of people on the labor market (Holl et al., 2018). To do this, the interventions were to focus on those with a medium chance of integration, under the assumption that interventions often do not help those with a low chance of integration and are unnecessary for those with a high chance of integration. The announcement of the AMS algorithm, as well as various transparency reports, triggered great resistance. Among the main criticisms were the lack of up-to-date training data on which the decision-making structure was based (the data dates back to the pre-COVID-19 period, when the labor market situation was completely different from today, see Schmidt et al., 2022), the stigmatization of an entire population group as 'hopeless cases' (Schmidt et al., 2022), and discrimination against various protected characteristics (Allhutter et al., 2020, Section 3.1). The developers justified the latter point by saying that the real (discriminatory) conditions on the labor market should be reflected, among other things, in order to be able to justify targeted measures against these conditions. They also proposed social compliance standards for the use of the AMS algorithm, compliance with which was intended to counteract the points of the criticism raised (Holl et al., 2019, p. 4).

As a result of these accusations, the test run was delayed and the nationwide roll-out, originally planned for January 1, 2021, was prohibited by the national data protection authority.[10] The case is still under investigation by the Administrative Court.[11]

Disclosure of the goal of using an ADM system enables a third party to judge whether it is consistent with societal goals and where potential risks are to be expected. Together with the evaluation process and success criteria by which the operator of an ADM system judges whether the goal of using the ADM system has been reached, it can also be judged whether the evaluation process is adequate for verifying the achievement of the goal. Together with inspectability mechanisms for later phases, a forum may even perform manual tests to see whether the goal has actually been

[10] https://www.derstandard.de/story/2000119486931/datenschutzbehoerde-kippt-umstrittenen-ams-algorithmus, last accessed on October 15, 2023.

[11] https://www.derstandard.de/story/2000135277980/neuerliche-kritik-am-ams-algorithmus-zum-in-die-tonne-treten, last accessed on October 15, 2023.

achieved.

While publishing the goal poses no considerable effort, the reaction of a forum (e.g., the general public) can be costly. The operators of AMS had multiple public discussions after activist groups[12] and scientists (Allhutter et al., 2020, pp. 7-8) deemed the software to be inherently discriminatory. One of the main points of criticism was that a separate group is classified as 'hopeless' and that the chances of these people finding their way back into employment are thereby made even more difficult.

There is no inspectability mechanism associated with this phase that we are aware of.

3.1.2 Phase B: Data Collection

ADM systems with a DDC are only able to derive good decision rules from data if a sufficiently large amount of high-quality data is available for the training process (Cortes et al., 1995; Roh et al., 2019). The collection of this data is often performed by different actors than the processing and further use, so the actors of the different phases potentially have different quality expectations. Actors in later phases often cannot trace the data collection process and thus cannot assess whether procedures may have been used that lead to bias in the data.

Data collection is subject to several legal restrictions, such as the General Data Protection Regulation (GDPR) in Europe. The requirements for collected data heavily depend on the application scenario. For example, facial recognition software that provides access to restricted areas in a company needs to learn from multiple images of all employees with the right to access. The data set does not need to be balanced to fairly represent all ethnicities, ages, religions, etc., but multiple images of each employee are potentially necessary to achieve high reliability later in the process. However, a facial recognition system that is to be used to find criminals in public spaces requires a diverse and balanced training data set to allow comparable prediction quality for all relevant groups (see Example 6). In addition, different application scenarios might accept different types and magnitudes of errors in the data.

Therefore, accountability of the data collection process is necessary for all subsequent steps in the development and usage of an ADM system. The Alan Turing Institute, 2021, even suggests that several scientific challenges related to the COVID-19 pandemic could have been mitigated through transparency mechanisms related to the data collection process.

The first transparency mechanism focuses on how and under what circumstances the data was collected.

[12] For example, independent trade unionists in the public service and in outsourced companies (orig.: *Unabhängige GewerkschafterInnen im Öffentlichen Dienst und in ausgegliederten Betrieben*): Neunteufel-Zechner, B., Pacak, M., "Diskriminierender AMS-Algorithmus", UGöD, https://www.ugoed.at/diskriminierender-ams-algorithmus/, last accessed on June 23, 2023.

Example 6 (Facial Recognition in Policing)

Facial recognition algorithms have persistently been found to have much higher error rates for minorities (e.g., Buolamwini and Gebru, 2018[13]). Especially when facial recognition technologies are used for policing, the higher error rates lead to additional adverse treatment of already marginalized groups. The most frequently discussed cause of this is that the training data for facial recognition technologies is often disproportionately populated with light-skinned (male) individuals (Schwartz et al., 2022, Chapter 3.1.1). At the same time, the technology is used in policing contexts to identify persons based on mugshots, which in Western cultures (especially the USA) are disproportionately populated with dark-skinned individuals (Garvie et al., 2016). In addition, it is sometimes argued that camera technologies are optimized to recognize the contours of light-skinned people against a light background. Recognizing dark-skinned people against a dark background is a secondary goal, if any (Buolamwini & Gebru, 2018). Together, these effects result in especially high false positive rates for dark-skinned (female) individuals (Klare et al., 2012).

B.1 Disclosure of How, When, and What Data was Collected

How and when data was collected can have a massive impact on how representative it is. If the method of data collection has a direct influence on the data, the influence is called a *conceptual bias* (Baer, 2019, pp. 74-75). If the data is too old or collected during a time when there are temporary societal changes (as during the COVID-19 crisis), it is possible that the data does not fit the current circumstances in an application. The effect of this phenomenon is called *concept drift* (Webb et al., 2016). Besides, the collected data must fit the intended task of the ADM system. Information that is irrelevant for an application and only collected and used because it is available may lead to so-called *availability bias* (Baer, 2019, pp. 19-23). Such information can still be filtered out before model training starts. However, if important information is filtered and thus missing, the training process might not be able to find sufficiently meaningful correlations between features and labels. Transparency about how, when, and what data has been collected not only helps the developers of an ADM system to make good decisions in the next steps, but also helps a forum to assess

[13] Note that Buolamwini and Gebru, 2018 confuse *True Positive Rate* (TPR) and *Precision* (PPV) in their Table 4. Also, they define the *Error Rate* as 1−*PPV*, which is usually referred to as *Miss Rate* or *False Negative Rate* (FNR) (Powers, 2020, p. 39). The *Error Rate* is usually defined as 1−Accuracy (*ACC*) (Powers, 2020, p. 39). Prof. Dr. Katharina Zweig noticed that something was off, and together we worked out what exactly was wrong. We talked with Dr. Joy Buolamwini and Dr. Timnit Gebru; they are going to republish a corrected version of their paper that is not available yet.

whether the ADM system can make meaningful decisions at all based on this information.

Many operators do not have the ability to collect the data they need for their own applications. Therefore, publicly available data is often used, or data is purchased from data collectors. Often, information about the circumstances under which such data sets were created is poor or missing. If such information is not provided explicitly, finding it is often difficult or impossible. Data sets that provide this information from the start may be more difficult to obtain or more expensive. However, if data is specifically collected for the task at hand, documenting the collection process does not involve any significant additional costs.

B.2 Disclosure of Operationalizations

Some information is not easy to quantify. An employee's satisfaction with their job, for example, cannot simply be measured or read out. If satisfaction is to be taken into account in a system, there is no choice but to infer it from other information, be it already available or still needing to be gathered, for example, by conducting a survey. However, how exactly satisfaction is to be quantified is a design decision. The process of making properties and concepts measurable (quantifiable) is called *'operationalization'* (Roskam, 1989, p. 241). In general, every complex term allows for multiple operationalizations with different grades of general agreeability. Furthermore, in most cases, there is not a single operationalization to which all agree. Thus, if such operationalization decisions were made before data collection, transparency about the way in which complex terms were measured and why this specific operationalization was chosen can allow a forum to judge whether it captures the most important aspects of the term to be quantified.

The cost of disclosing operationalizations can be estimated to be low. The person(s) who conducted the operationalizations (or chose already existing ones) and the person(s) who decided that the operationalizations were reasonable should have thought about this thoroughly and documented it anyway. If not, the cost for this transparency mechanism might increase, but applying it is also most likely to help uncover problems with operationalizations.

3.1.3 Phase C: Training Data Set Construction

The raw collected data needs to be transformed into an appropriate training data set before it can be used for training a model. What can be considered appropriate strongly depends on the available data and its intended use. The transformation steps that are relevant for a forum include:

1. *Identifying the output variable* in the data set. The output variable is the label to be predicted by the ADM system (see Definition 2, p. 13, and 3, p. 13). If the output variable is not part of the original data

collection, it needs to be added to each input for the training data set.

2. *Selection of input variables*. The collected data may contain more variables than are relevant for a given task. Therefore, those variables in the data set that are most likely related to the output variable are selected (see Definition 2, p. 13, and 3, p. 13). Also, it might be necessary to compute more complex variables than in the original data set, for example, by normalizing the values of a variable or by combining variables into a new variable (García et al., 2016).

3. Further '*data pre-processing*', i.e., dealing with input data that contains missing data (missing values imputation) or wrong data (noise treatment) (García et al., 2016).

Just as with the data collection process, actors who subsequently use the training data set do not necessarily have insight into the construction process and the design decisions made there (Gebru et al., 2021). Bad decisions or decisions that do not fit a later application can not only lead to discriminating decisions. It might happen, for example, that the label (and/or description) of a parameter does not correspond exactly to what is hidden behind its values. Using it for model training or validation might therefore adversely affect the ADM system, or create unwarranted confidence in it.

There are multiple transparency mechanisms that relate to the construction of the training data set.

C.1 Disclosure of the Labeling Process

The output variable to be predicted by an ADM system is sometimes part of the collected data (or can be computed ad-hoc). In this case, how and when it was observed is already part of B.1. If this is not the case, the output variable needs to be labeled by hand. There are various labeling processes, ranging from manual labeling (A. Taylor et al., 2003) to crowd-sourced techniques (Chang et al., 2017; Snow et al., 2008) to fully automated approaches (Agichtein et al., 2006). For example, one data set uses the tags that users gave to photos on *flickr*[14] (Panteras et al., 2016); *recaptcha* asks users to classify photos for their customers, for example: '*Pick all images with a car on it*' (Von Ahn et al., 2008). Others base the data classification on user behavior; for example, the relevance of a web page for a user is measured by the time spent on that page (Agichtein et al., 2006). Each of these approaches comes with different weaknesses and qualities. If the labeling is conducted as a separate step, the details of this process could be disclosed to better understand the quality of the resulting training data set (DIN/DKE, 2020, p. 99).

The information made transparent could include (i) who labeled the data, (ii) how these persons were trained, (iii) whether they were representative of the population, (iv) whether there was a process to identify

[14] *flickr* is an image and short video hosting platform that allows users to add notes and comments. https://www.flickr.com/, last accessed on March 21, 2023.

false labels (e.g., multiple persons labeled the same data and discrepancies were resolved, like we did in Hauer, Krafft, Sesing-Wagenpfeil, Zweig, et al., 2023), (v) how multiple fitting labels were handled, and much more. Based on this information, a forum can judge to what extent the persons organizing the labeling process made appropriate decisions and, therefore, whether the labels are truthful and match their description. Which information is of particular interest can hardly be stated in general and depends on the specific data and labels.

The costs are rather low, but publication of the details might result in social pushback. For example, the labels assigned to unemployed people in the AMS algorithm (see Example 5, p. 55) have been widely debated and partly condemned (Allhutter et al., 2020).

C.2 Disclosure of Why Which Variables Were Included in the Training Data Set

The selection of variables for the training data set consists of variables already included in the collected data. The data might be transformed. For example, values could be normalized between 0 and 1 and a classification label might be assigned a number. The selection might also involve the more complex construction of new variables, which is called *feature engineering*. This is done, for example, by multiplying variables with each other or by clustering them (Heaton, 2016, p. 1).

If this selection of variables is made transparent, a forum is able to judge whether all variables are likely to be causally related to the output variable. Additionally, the persons constructing the training data set can justify their selection of relevant variables and explain feature engineering steps, which enables the forum to evaluate the appropriateness of the selection. They can also justify why some information has been omitted.

As a short explanation for each variable should be sufficient, we estimate the cost of this mechanism to be very low. Optimally, these explanations will have been documented anyway during the development process and would only need to be translated into a format that the respective forums can understand.

C.3 Disclosure of Pre-Processing Techniques

Any data collection process will most likely result in data that contains inaccurate, false, or missing values and has other imperfections that make the data not perfectly fit for further processing purposes (Frénay & Verleysen, 2014). Thus, many approaches have been developed to tackle various data quality problems (Luengo et al., 2012), all with their pros and cons. When data points have missing values, for example, one option is to artificially create values based on statistical evaluation of other values. This may, however, reduce the expressiveness of the data. Another option is to completely remove the data points (Little & Rubin, 2019), which could mean eliminating certain edge cases where data acquisition is more difficult. Another important approach is '*data reduction*', which aims to increase the information density inherent to the training data. One specific

technique is targeted feature selection to identify and remove redundant information (M. A. Hall, 1999), which would only introduce a source of error, for example, by inducing spurious correlations. The curators of the data set need to decide which methods to apply under which conditions. The process of improving data quality and adjusting it to a given task is called '*data pre-processing*' (see Definition 16).

Transparency about the exact methods applied allows a forum to judge their general suitability and their suitability in the context of (a) given application scenario(s).

The costs are rather low. In many cases, all pre-processing steps are contained in one rather small script which is easy to understand and follow. Thus, it might actually be the simplest way to publish that code.

Definition 16 (Data Pre-Processing)

Data pre-processing addresses techniques for dealing with imperfect data. García et al., 2016, show a wide field of specific approaches and suggest some classifications, though there is no absolute and final list. Some of the most prominent data pre-processing tasks are:

- Data cleaning (removing data points with missing values or filling in these values),
- Data reduction (removing similar data that does not provide additional insight),
- Data transformation (normalization, aggregation, discretization, etc.).

C.4 Disclosure of Training Data Properties

After the pre-processing steps, the polished data set is ready to be used as a training data set. It can be evaluated for its properties based on various factors and distributions, for example, the number of data points, the number of data points for various groups of interest (e.g., based on gender, ethnicity, age, etc.), and statistics of the distribution of the values for each variable (e.g., mean, standard deviation, etc.).

Besides aggregated statistics, the rates of missing and possibly wrong data for each variable before pre-processing are of interest, and, if applicable, the error range of the corresponding measurement process (e.g., sensor error rates). Last but not least, the same statistics about the deleted input data could be published to understand whether the deleted data deviates from the kept data.

All this information can be provided in the form of a spreadsheet or a database (Diakopoulos, 2014). Gebru et al., 2021 provide an extensive list

and structure for such aggregated information about collected data that can help to identify relevant aspects in the form of a so-called *datasheet for data sets*. This allows disclosing information without publishing any potentially protected data at relatively low cost.

With such information, a forum can estimate the quality of the resulting data set (DIN/DKE, 2020, p. 85) and whether the pre-processing steps have removed or distorted important aspects of the original data set. Depending on the findings, the persons involved in data collection or pre-processing are the actors to be held accountable.

I Full Access to the Data

The pure information about the data might not be enough to make the actors accountable. In some cases, it might be necessary to give a forum access to the data (either fully or in parts), including all information necessary to understand it, as, for example, demanded by Algorithm Watch[15] or the German standardization roadmap on Artificial Intelligence (DIN/DKE, 2023, multiple occasions, e.g., p. 2). This access allows a forum to answer questions about the data quality and data set properties themselves. In particular, all kinds of questions regarding the suitability of the data with respect to certain vulnerable groups based on, for example, gender, religion, ethnicity, or age can be answered with this access. Additionally, the mechanism may provide the basis for making data available under appropriate conditions, for example, for medical research purposes, as also discussed in the *Data science and AI in the age of COVID-19* report[16] by the Alan Turing Institute.

The costs for providing such access are considerable. Data is often a valuable asset for a company, the provision of which could promote economic competition. Furthermore, the use of a lot of data is restricted by law. Identifying which data can be published or made accessible in which way is likely to be costly, as is the implementation of an access system. If direct publication of the data is not advisable, a system can be built that allows asking for aggregate statistics of groups of input data, for example, the mean salary of women vs. men in a financial data set (Michener & Bersch, 2013, p. 239), or that provides access to more general information, like the town instead of the specific address. However, in these cases, it might be necessary to ensure that the publication of information cannot be used to deanonymize any specific person in the data set (Ohm, 2009, p. 1751). Checking for this possibility and producing a system that prohibits deanonymization is likely to increase the costs.

With regard to accountability, this mechanism allows forums to asynchronously ask questions regarding all information made transparent by mechanism B.2 (Disclosure of Operationalizaions) and in phase C (Training

[15] Draft AI Act: EU needs to live up to its own ambitions in terms of governance and enforcement, p. 8: https://algorithmwatch.org/en/wp-content/uploads/2021/08/EU-AI-Act-Consultation-Submission-by-AlgorithmWatch-August-2021.pdf (last accessed on September 21, 2023).

[16] Data science and AI in the age of COVID-19: https://www.turing.ac.uk/sites/default/files/2021-06/data-science-and-ai-in-the-age-of-covid_full-report_2.pdf, last accessed on August 31, 2023.

Data Set Construction). It therefore addresses all actors relevant in these phases, namely everyone who was involved in data operationalization, data collection, and data pre-processing.

3.1.4 Phase D: Choice of Machine Learning Method and Hyperparameters

In the development process of an ADM system, multiple ML methods might be tried out until the actors settle for the one with the best quality assessment results (see Section 3.1.6). Most of these methods come with a handful of *hyperparameters* (see Definition 6, p. 20), for which several values are usually explored before settling on the purportedly best combination. Information about a method and the hyperparameters allows assessing their suitability under the given circumstances (such as the amount of available training data).

Another important piece of information is how the ML method was implemented – in most cases, one of the many freely available software packages will have been used. Common tools are, for example, Keras[17], KNIME[18], and RapidMiner[19]. Very rarely will software development teams implement an ML method from scratch.

D.1 Disclosure of the Method, its Implementation, and the Hyperparameters

The ML method, its implementation, and the settings for all hyperparameters[20] are relevant pieces of information if the suitability of choices needs to be assessed (DIN/DKE, 2020, p. 86).

Determining which method and which hyperparameters yield sufficiently accurate predictions often requires multiple iterations of trial and error by the responsible data science team developing the model. There is no way of saying for sure which choices are potentially the best, but for some situations, there are plainly wrong choices. For example, a very simple regression model might make assumptions about the form of the data that are not met, and a very complex ML procedure might require more data than available in the training data. Another example is given by unbalanced data, i.e., data that contains more information about one subgroup than about other subgroups. Here, certain methods perform clearly better than others, as thoroughly investigated by Haixiang et al., 2017, p. 225.

Based on the information retrieved in phase A (Requirements Engineering, see Section 3.1.1), it can be assessed whether the selected method complies with the specified explainability requirements. Combined with the information from phase B (Data Collection, see Section 3.1.2) or even phase C (Training Data Set Construction, see Section 3.1.3), it can be evaluated whether the properties of the available data may be sufficient for a selected method or not (Cheng et al., 2018).

[17] https://keras.io/, last accessed on June 23, 2023.
[18] https://www.knime.com/, last accessed on June 23, 2023.
[19] https://rapidminer.com/, last accessed on June 23, 2023.
[20] Note that some hyperparameters are more relevant in the context of the training procedure, such as the stopping criteria.

The costs for disclosing these pieces of information are very low, although providing a comprehensible description might be challenging. Mitchell et al. introduced a framework called *model cards* for structuring and disclosing such information (Mitchell et al., 2019).

3.1.5 Phase E: Learning Procedure

When the training data set has been constructed and the ML method has been decided upon, the data science team has to decide on a learning procedure. Whether this will be a supervised or an unsupervised learning approach is most likely determined by the task at hand and the data available (see Example 1, p. 14). Depending on the ML method, there are plenty of details to specify in either case. For an ANN, for example, the number of training iterations before weights are updated (the so-called *batch* size) and the conditions for concluding the training (the so-called *stopping criteria*) have to be set (see Section 2.1.2). Based on the training data, the system is, in almost all cases, only trained on a part of that data and later tested on another part (see Chapter 5 and Definition 19, p. 109). How the data set is divided into these parts is an important decision. Additionally, some procedures are order-dependent, i.e., the resulting statistical model might be influenced by the order in which the training data is fed into the learning system (Lopes et al., 2017, p. 621). Furthermore, there are approaches for continuing learning during testing or after deployment of a system, for example, by retraining on the basis of misclassifications (so-called *reinforcement learning*).

E.1 Disclosure of Training Details

Transparency of the training details may raise awareness of where to look for potential sources of error, especially in the case of continuous training. Together with the information from C.I (access to the data; see Section 3.1.3) and D.1 (Machine Learning method, implementation, and hyperparameters; see Section 3.1.4), it allows building an ML model under the same conditions. If a deterministic method and a deterministic learning procedure have been selected (including the order of the data used for training), even the exact same model can be retrained. Building a new model that is as comparable as possible to the model inspected (a so-called *surrogate model*, see Definition 15, p. 52) may be a promising approach to examine a model that cannot be accessed directly to identify potential misbehavior.

While the cost of publication appears to be low, it may be difficult to document the exact learning procedure in a comprehensible way. Since the ML method and the learning procedure are potentially interdependent and usually selected or implemented by the same data science team, it makes sense to collect and disclose the information about both phases in one document. However, it may be that there are special trade secrets in only one of the two phases. In this case, separate documentation may be more advantageous. The handling of trade secrets and how they affect costs will be thoroughly discussed in Section 3.1.9.

3.1.6 Phase F: Quality Assessment

Quality assessment can be divided into two separate phases. The first is the assessment of the system itself, i.e., the question of how many and which errors it makes. The second phase addresses the evaluation of the system in an application scenario and is discussed in phase H (Evaluation in an Application Scenario; see Section 3.1.8). Optimally, how to assess system quality has already been defined during requirements engineering (for more about the so-called *test-first approach*, see Section 5.7): This phase discusses the disclosure of assessment details and results. The quality of a trained ML model is usually assessed by computing quality and fairness measures (see Section 2.3.3) based on a data set the system has not yet seen before, i.e., that has not been used for training (but went through the same pre-processing steps). However, there may be further evaluation metrics (see Chapter 5 for a broad elaboration on testing).

F.1 Disclosure of the Results of all Evaluation Metrics

The results of all evaluation metrics mentioned in the requirements documents and any further evaluation metrics deemed necessary later in the process as well as all information relevant to assessing the meaning of the metrics, like details about the test data set and test conditions, can be disclosed in two different accountability processes: If accountability is to be established toward a customer (who might also be an operator), the software development team has to justify the quality of the system they developed and tested. If any third parties (e.g., NGOs, auditors, or judges) have to evaluate whether a system is good enough to be used in a certain scenario, the operators of the system are the responsible actors, since they actually decide whether to use the system or not. In any case, explicitly mentioning what metrics have been computed with what justification and what implications these metrics entail is important to be able to assess whether or not these metrics are appropriate for a particular application in a particular context (Busuioc, 2021, Section '*From Implicit to Explicit Value Trade-Offs*').

The costs of disclosing the results of evaluation metrics and how they have been computed are very low. However, (honestly) justifying the selection of specific metrics can be challenging – either because a lot of thought went into the selection process, which is difficult to explain, or because standard metrics were simply chosen without any reflection, which may present a target for justified criticism.

In this crucial phase, the ADM system is handed over to the customer or operator and the system starts to be used in an application scenario. While information about its quality is helpful, forums might in fact want to question that information and inspect it for themselves (Citron & Pasquale, 2014). This could be achieved by giving access to the ADM system such that it can be tested with any test data set deemed appropriate.

II – Full Access to the Output of the ADM System for any Specific Input Data under Lab Conditions

With unrestricted access to the system in terms of feeding it with input data and accessing the resulting output, forums can run any test they deem fit to assess the system's quality and determine whether the specified requirements are satisfied. With carefully selected or crafted input data, new questions can also be answered; for example, counterfactual questions like: If this input data came from a 23-year-old person instead of a 45-year-old, would the decision be different? If the operator did not provide any fairness measure results, they can also be produced by this access for any desired subgroups of input data. Asking questions in this way is a special form of black-box audits, which will be thoroughly discussed in Chapter 4. Combined with the mechanisms I (Full Access to the Data; see Section 3.1.3), D.1 (Disclosure of the Method, its Implementation, and the Hyperparameters; see Section 3.1.4), and E.1 (Disclosure of Training Details; see Section 3.1.5), a forum is theoretically able to train a model under the same conditions and run any explainability approach they deem relevant (see Definition 15, p. 52). The same is possible with less effort if the code of the trained model is handed over to the forum. This also makes the analyses more meaningful, as it is not necessary to first recreate a model that does not necessarily correspond one-to-one to the actual model to be investigated.

The costs for this mechanism can vary strongly, but can be estimated to be rather high. In cases where such access is justifiably required by legal authorities, the ADM system is likely to be used in a sensitive application scenario. Thus, unrestricted access might not be advisable. Setting up protected access for specific forums, for example, by means of a password-restricted API, may drive up costs. Furthermore, extensive inspection of the ADM system might reveal its inner workings so much that someone with access could rebuild a similar system, which could infringe on trade secrets. Others could learn how to exploit the system, i.e., how to adjust input data to achieve a specific decision by the system. All in all, implementation and maintenance of this mechanism are likely to be very costly and maybe even dangerous for the operator.

Without question, this kind of access for (selected) forums is the most valuable in assessing the suitability of an ADM system; however, it is also the one whose requirements need to be carefully justified. With regard to accountability, this mechanism allows a forum to asynchronously ask questions regarding all information made transparent in phases D, E, and F.

3.1.7 Phase G: System Usage in Application Scenario

In addition to the requirements for the ADM system itself, the operator could define some further procedural requirements for its use in an application scenario. For example, ADM systems operated by the state will generally be used by civil servants. In such cases, the operator might set some internal rules regarding usage. The developers of the AMS algorithm

(see Example 5, p. 55) required a set of standards addressing social compliance (orig. *'Sozialverträglichkeit'*, see Holl et al., 2019); for example, that (i) any algorithmic output is only meant to support the decision of a human, (ii) the person judged by the system can object to its decision, or (iii) the person can also see their input data and correct it, if necessary. This means that if the stakeholders do not comply with the social compliance standards, they take full responsibility for any effects resulting from using the software, independent of whether it works as intended or not.

G.1 Disclosure of the Procedural Requirements in the Usage of an ADM System

With all rules or requirements regarding the use of an ADM system, such as those concerning how the data used for actual decisions is collected, whether there are humans in the loop, how they can overrule a decision by the system, how a person can object to a decision, etc., forums can judge whether they are suitable for the application. They can also use the disclosed information to check whether the rules and requirements are actually followed in practice.

The direct costs associated with this mechanism are rather low for the same reasons as in phase A. Since it also discloses important protection mechanisms for users and stakeholders, it is likely to be of great value to the forum in most cases.

In some cases, there is another phase in the usage of an ADM system in which its results are evaluated and complaints are collected and used for feedback to improve the system. The following section lists transparency and inspectability mechanisms for this phase.

3.1.8 Phase H: Evaluation in an Application Scenario

In most cases, the operator of an ADM system will pursue an overarching goal with its use. It can be assumed that an evaluation process will be implemented to check whether the ADM system is in fact suited to reach this goal. Since the decisions of an AI-based ADM system may be challenged after it has been updated or continued training, an evaluation of the current software version says nothing about its quality at the time it made the challenged decision(s). Therefore, it may be necessary to make use of a version history for the system in order to be able to inspect the version that was active when the decision being challenged was made.

In some circumstances – for example, if the operator is the state or when the usage of the ADM is in conflict with rights of other parties (e.g., employees) – it might be necessary to disclose the evaluation process(es) (H.1) (Krafft et al., 2022, p. 11) of a system in operation and/or their results (H.2) (Crawford, 2016).

H.1 Disclosure of the Evaluation Process of the ADM System in Operation

There are many ways an ADM system can be evaluated in operation. Automatic tests, such as the computation of quality and fairness measures,

can be run on a regular basis (e.g., after each new classification for the last *n* classifications; see *Field Testing* in Section 5.5) or a manual review process can be established (e.g., by letting test persons try the system and evaluating their feedback). The details of the evaluation process of an ADM system in operation and the consequences of unsatisfactory results allow forums to judge its suitability.

The costs of publication may be low if the process is already well documented by the operator.

H.2 Disclosure of the Evaluation Results of the ADM System in Operation

The results of the evaluation process can be disclosed independently. However, if an operator publishes any metrics without explanation, they might be difficult for a forum to assess. Together with knowledge about the evaluation process of the ADM system in operation (H.1), a forum may be able to estimate to what extent the ADM system is able to satisfy the overarching goals. However, this strongly depends on the suitability of the evaluation process.

As the results of any evaluation process should be documented anyway, the costs can be estimated to be low.

III – Full Access to the Output of the ADM System for any Specific Input in Operation

Even if a system has been extensively tested before operation, it may behave unexpectedly in actual use; for example, because circumstances are relevant that were not taken into account in the tests or because the real inputs differ from the test data.

Full access to the output of the ADM system for any specific input in operation provides the most reliable information about how the system actually behaves. Using it enables forums to monitor any system behavior and experiment with it under usage conditions. They can also control whether the requirements were really satisfied, for example, whether the quality of the system is at least as good as the predefined quality threshold. As requirements can change over time (e.g., due to concept drift), it can also be used to monitor the system on a regular basis.

Sandvig et al., 2014, describe different possibilities of getting this access as *black-box audits*, which will be discussed in detail later (see Section 4.1.2). The option of creating an API has already been described for phase II (*scraping audit*). Another option is to survey people who have been evaluated by the ADM system in order to find out whether they had a chance to see their input and output data (*non-invasive user audit*). Informed test candidates can be assigned to probe the process or provide actual results of the system for specified inputs in its application context (*crowdsourced audit*).

Since this mechanism allows experimentation with the system in operation, the costs are high. On the one hand, the effort to protect the people affected by the system must be considered. Their data must not

Table 3.1: Summary of all transparency and inspectability mechanisms and their estimated costs according to Hauer, Krafft, and Zweig, 2023. The costs of A.2. (Disclosure of Requirements Documents) and A.3. (Disclosure of the Goal of Using an ADM System) might be considerably higher, depending on the circumstances (see explanations of phase A in Section 3.1.1).

Characteristic	Transparency	Cost of disclosing information	Inspectability	Cost of granting access
A. Requirements engineering	1. Disclosure of application scenarios	low		
	2. Disclosure of requirements documents	low*		
	3. Disclosure of the goal of using an ADM system	low*		
B. Data collection	1. Disclosure of how, when, and what data was collected	low		
	2. Disclosure of operationalizations	low		
C. Training data set construction	1. Disclosure of the labeling process	low	I. Full access to the data	medium
	2. Disclosure of why which variables were included in the training data set	low		
	3. Disclosure of pre-processing techniques	low		
	4. Disclosure of training data properties	low		
D. Method selection	1. Disclosure of the method, its implementation, and the parameter settings used	low	II. Full access to the output of the ADM system for any specific input data	high
E. Training	1. Disclosure of training details	low		
F. Quality assessment	1. Disclosure of the results of all evaluation metrics	low		
G. Application	1. Disclosure of the procedural requirements in the usage of an ADM system	low	III. Full access to the output of the ADM system in operation	high
H. Evaluation in applications	1. Disclosure of the evaluation process of the ADM system in operation	low		
	2. Disclosure of the evaluation results of the ADM system in operation	low		

be made public through the use of this mechanism, and it must be ensured that the experiments do not affect how the system treats them. On the other hand, the operator runs the risk of forums gaining insights into the system behavior that jeopardize a trade secret or allow the system to be exploited.

3.1.9 Discussion

As noted several times, providing possibilities to inspect the system also comes with downsides. Transparency, as defined here, is identified with the disclosure of information that already exists or can be prepared relatively easily. Thus, the costs of applying these kinds of transparency mechanisms are generally rather low (see Table 3.1). However, there is no central infrastructure or any other defined way for this kind of information to be published and made accessible yet. For some specific transparency mechanisms, there are suggestions on how to best communicate them, such as the '*datasheets*' suggested by Gebru et al., 2021, with regard to disclosing information about the data, or the '*model cards*' suggested by Mitchell et al., 2019, for disclosing information about an ADM system. The transparency mechanisms allow forums to review disclosed information and explore it for aspects that are relevant to them. However, if a

question cannot be answered with the information, only a limited judgment is possible. Requiring transparency mechanisms to be implemented also poses hidden challenges that are very difficult to estimate upfront. Paul B. de Laat, for example, lists four major concerns: (i) privacy challenges, (ii) making a system gameable (i.e., exploitable), (iii) publishing trade secrets, and (iv) limited use of transparency mechanisms (De Laat, 2018, pp. 534-535). To mitigate risks, research is geared toward finding new solutions, such as introducing methods that prevent some of them, for example, by disclosing only aggregated information (Ohm, 2009, p. 1751) and thus protecting the private information of individuals, or by letting qualified organizations pre-process the information that is made transparent as outlined by Pasquale, 2015, p. 142. Others state that some risks, such as the possibility of exploiting systems, are exaggerated (Cofone & Strandburg, 2019). The protection of trade secrets allows companies to limit the rights to access information. For example, both the GDPR (Wachter et al., 2017) and the US Freedom of Information Act (FOIA) (Diakopoulos, 2014, p. 12) allow companies to deny software transparency requests. This limits the ability of forums to exercise meaningful oversight over the operation and functioning of algorithms on a legal basis. At the same time, this gives actors leverage to avoid having to comply with transparency and justification obligations (Busuioc, 2021, p. 829). The limited use of transparency mechanisms is addressed by suggesting complementary inspectability mechanisms.

Inspectability mechanisms allow forums to directly validate information about an ADM system. In terms of costs, such mechanisms take more time and effort to implement (see Table 3.1). Additionally, they may require confidentiality toward the forum, as reverse engineering based on the actually used training data (mechanism I) as well as unrestricted access to the output of the ADM system for certain data (mechanisms II and III) is possible. Using the inspectability mechanisms, a forum is able to pose very general questions and systematically refine them to enable them to pass judgment. Imagine, for example, that a forum is interested in the question of whether the system is biased against an ethnic minority. With full access to the data (mechanism I), the forum can compute any statistical value in which it is interested. It may also just examine the data for anything unusual or unexpected and thus dynamically develop a more specific question, such as *'Is the minority subgroup sufficiently represented in the data?'* or *'Do certain values in the data of the subgroup differ significantly from those of other groups?'*. With access to the output of the ADM system (mechanisms II and III), a forum is able to submit self-constructed test data sets to a system and evaluate the results. Various specific concepts for doing this are referred to as audits (for more about auditing, see Chapter 4). They also enable forums to perform various explainability approaches, such as constructing a white-box surrogate model based on a large number of pairs of inputs and outputs. In any case, it is practical to pose any questions in the form of clearly defined tests, as these provide a deterministic and comprehensible process, the result of which can be used directly as a judgment (for more about testing, see Chapter 5).

The primary focus of legal institutions is to assess which mechanisms

must be accessible by which forums to hold which actors accountable (Nemitz, 2018, p. 8). In the context of AI-based ADM systems, risk-based regulation approaches, in particular, are being discussed as a basis for choosing application-specific transparency and inspectability requirements (more on this in Section 6.1.1).

RQ 2

What mechanisms can be implemented to allow system analysis?

Answer: In the context of algorithmic accountability, it makes sense to differentiate between transparency mechanisms and inspectability mechanisms.

Transparency mechanisms refer to any means that disclose relevant information. In the context of an AI-based ADM system, this includes at least:

- The application scenarios
- The requirements documents
- The goal of using an ADM system
- How, when, and what data was collected
- The operationalization of information in the form of data
- The data labeling process
- Why which variables were included in the training data set
- The pre-processing techniques applied
- The training data properties
- The method, its implementation, and the parameter settings used
- The training details
- The details and results of all quality evaluation processes and metrics
- The procedural requirements on the usage of an ADM system in operation
- The details and results of all quality evaluation processes and metrics in operation

Disclosure of any of this information allows forums to inspect, analyze, and judge information. However, they will not be able to assess whether the disclosed information is correct or made up.

Inspectability mechanisms refer to any means used to provide forums with access to a system and its components, allowing them to analyze them on their own. In the context of an AI-based ADM system, this includes at least:

- Access to the training data
- Access to the output of the system for any specific input

- Access to the output of the system in operation

Such forms of access allow forums to assess, to a large extent, whether the disclosed information is correct. They also allow forums to apply any testing method, measurement computation, or explainability concept they deem suitable and helpful.

3.2 Actors to be Held Accountable

The relevant actors are only identified in a very abstract way in the explanations given so far. This is because legally responsible actors must be determined by the legislature or, more generally, by regulatory bodies. Computer scientists can only provide a scientific basis to support the decision-making process. In the context of regulating AI-based systems, the EU Commission recognizes the need to differentiate between different actors. This is also reflected, for example, in the AI Act.[21]

John Austin's Speech Act theory could also provide a strong rationale for deciding which actors are to be held accountable for certain aspects of AI-based ADM systems.

John Austin's Speech Act Theory

From 1939, John Austin developed a theory according to which most statements are implicitly activities beyond the mere utterance of words. He believed that in saying something, one is performing an act he refers to as *speech act*. For example, when construction workers shout *'Attention!'*, they are not just saying the word, they are also performing the act of warning of an imminent danger. Over the years, Austin further refined his theory and discussed it with students in his lectures from 1952 to 1955. In 1962, the latest version of his lecture notes was edited minimally and published posthumously (Austin, 1962).

Unlike a simple statement, a speech act cannot be judged as being true or false. Instead, according to Austin, six conditions can be used to determine to what extent a speech act will succeed or may fail (Austin, 1962, pp. 14-18). He divides these conditions into three categories (see Example 7, p. 75). If conditions of type A or B are not fulfilled, the act is void. If conditions of type Γ are not fulfilled, the act is professed, but hollow. This difference is pointed out by choosing Γ instead of C:

- A: *'Misinvocations'*.
  - A.1: *'There must exist an accepted conventional procedure having a certain conventional effect, that procedure to include the uttering of certain words by certain persons in certain circumstances, and further, ...'*
  - A.2: *... the particular persons and circumstances in a given case must be appropriate for the invocation of the particular procedure invoked*'.

[21] For example, Art. 16 - Art. 23 refer to the provider, Art. 24 to product manufacturers, Art. 25 to authorized representatives, Art. 26 to importers, Art. 27 to distributors, Art. 28 to distributors, importers, users, or any other third-party, and Art. 29 to users.

- B: *'Misexecutions'*.
  - B.1: *'The procedure must be executed by all participants both correctly and...'*
  - B.2: *'... completely'*.
- Γ: *'The act is professed, but void'*.
  - Γ.1: *'Where, as often, the procedure is designed for use by persons having certain thoughts or feelings, or for the inauguration of certain consequential conduct on the part of any participant, then a person participating in and so invoking the procedure must in fact have those thoughts or feelings, and the participants must intend so to conduct themselves, and further...'*
  - Γ.2: *'... must actually so conduct themselves subsequently'*.

If one of these conditions is not fulfilled, the speech act fails because of an error of the respective type. Austin points out, however, that a speech act does not necessarily succeed just because all these conditions are fulfilled. They are necessary but not sufficient conditions.

Example 7 (Selling a Car)

Anita wants to sell her car to Bob. She agrees with Bob by shaking hands that she will transfer the keys and all relevant documents to him in exchange for a certain amount of money. In order for this speech act to be successful, the following six conditions must be met:

A.1 The conventional procedure in this case is the making of a verbal contract for selling the car and signing it by shaking hands. The conventional effect is the creation of an obligation on the part of the seller (Anita) to perform the promised action (transfer the car title) once she receives the money.

A.2 Anita and Bob are appropriate persons to invoke the procedure of buying and selling a car; for example, they need to be legally allowed to perform that transaction and Anita needs to own the car.

B.1 Anita must use the correct words to make the verbal contract and Bob must understand the contract that Anita formulates.

B.2 Anita must formalize the complete contract to transfer the car title in exchange for money and Bob must accept these conditions by shaking hands with Anita.

Γ.1 Anita must intend to transfer the keys and all relevant documents to Bob and Bob must intend to pay Anita for the car.

Γ.2 Anita must actually transfer the keys and all relevant documents to Bob and Bob must actually pay Anita for the car.

According to Austin, a speech act can be studied as a locutionary, illocutionary, and perlocutionary act, which are all aspects of the whole speech act. The locutionary act is the actual act of saying something. The illocutionary act refers to the intended effect of the act. The perlocutionary act implies a certain consequential effect of the act. These explanations are deliberately brief. Austin's elaborations on the differences and boundaries take up a great part of his book and are thus too exhaustive for this document. However, the examples chosen by Austin are suitable for giving an idea of these three types of acts (Austin, 1962, pp. 101-102):

Example 1: *'Shoot her!'*

Locutionary act: He said *'Shoot her!'*.

Illocutionary act: He urged (or advised, ordered, ...) the officer to shoot her.

Perlocutionary act: He persuaded the officer to shoot her. He got the officer to (or made the officer, ...) shoot her.

Example 2: *'You can't do that!'*

Locutionary act: He said *'You can't do that'*.

Illocutionary act: He protested against my doing it.

Perlocutionary act: He stopped me, he brought me to my senses, ...

Example 3: *'Saying something'*

Locutionary act: He said that...

Illocutionary act: He argued that...

Perlocutionary act: He convinced me that...

According to Janich, 2015, p. 313-314, only a human being can perform a purposeful action, but a machine can perform a substitution of equal performance on behalf of a human being.[22] K. Zweig, 2023, p. 169, explains this statement with a sign warning of an electric fence. The owner of the fence 'speaks' the warning, but the sign substitutes this act.

In our 'Master Reading Course', Prof. Dr. Katharina Zweig and I discussed the Speech Act theory with Master students and interpreted the recommendations made by an AI-based ADM system as a substitution of equal performance performed by a machine. My primary question in the context of this lecture series was: *Who are the actors whose (partial) acts are substituted by the machine, so that it can be determined who should be accountable for individual aspects of the use of an AI-based ADM system according to Boven's and Wireinga's understanding of accountability*?

To reflect the theoretical considerations directly using a practical example, I discussed how this can be used to identify accountable persons for an online shop for which software calculates automatic product recommendations that are displayed on the website. In my opinion,[23] the speech act of a recommendation made by an AI-based ADM system consists of multiple partial acts (e.g., the phases of the long chain of responsibility could each be considered partial acts), which together form a substitution of the act of recommendation when given an input. This substitution may fail according to A.1-B.2 faults (e.g., due to incorrect data or method selection) or be void as a result of Γ faults. To investigate the question of which actors should be accountable for which partial act, I examined the speech act as a locutionary, illocutionary, and perlocutionary act. In the following, I will look at possible faults that can occur according to the conditions proposed by Austin and explore who would be responsible for the fault in each case, something Austin himself did not address in his lectures. However, regarding the identification of actors, I find this method helpful for investigating who is accountable for a speech act that is substituted by a machine.

[22] Original (German): *'Unterschied zwischen der (menschlichen) Handlung des Rechnens und der leistungsgleichen Substitution des Rechnens durch Maschinen'*.

[23] The following considerations were discussed with students of Socioinformatics, Dr. Katharina Zweig, professor of Computer Science, and Dr. Jan Georg Schneider, professor of German Linguistics. Nevertheless, as someone who is not a philosopher of language, I cannot interpret a text on the philosophy of language as clearly and plausibly as a domain expert could.

The mere creation of the string (and the possible inclusion of images) is the substitution of the locutionary act. This act is only about calculating a system output, regardless of its content, correctness, or added value. It results from a set of mathematical operations that are performed on a system input without any judgment. With this definition, I kind of contradict Austin's explanations of locutionary acts. He points out, for example, that a) an animal can make sounds, which have no linguistic meaning. They do not perform, what he calls, a 'phatic act' Austin, 1962, p. 92. Furthermore, a locutionary act presupposes that b) the words have a semantic meaning, i.e. they refer to something. This is what Austin calls a 'rhetic act' Austin, 1962, p. 92. For the application of the Speech Act theory on language-generating AI-based systems (commonly known as *Large Language Models*, such as ChatGPT[24] or Bard[25]), this restriction is of utmost importance. In the context of an automatic product recommendation system, however, the machine only places words (e.g., product names) and images given by humans at given positions. Therefore, I consider this interpretation of the substitution of the locutionary act by the machine as reasonable.

A fault of type A cannot occur. The rules of mathematics provide an accepted conventional procedure (A.1), and any person who is able to understand and apply these rules is appropriately equipped to calculate them (A.2). Developers are responsible for ensuring that a machine performs computational operations correctly and that no fault of type B occurs. Since this locutionary act is independent of thoughts or feelings, the fault type Γ is not applicable here. Therefore, I see the developer as the only actor whose locutionary speech act is substituted by a machine.

The display on a shop's website substitutes a recommendation, which is an illocutionary act. An illocutionary act presupposes that it is heard and understood. The recommendation by a machine can therefore only be a successful speech act if the recommendation is also 'heard' and understood. Otherwise, there is an A.1 type of fault. It is difficult to say in general terms how this fault can occur in practice and who is then responsible for it, but it should be easy to identify in a concrete application with a concrete process. In the case of an online shop, for example, the recommendation could be made in a language that the reader does not understand. This consideration presupposes that the presentation of the website explicitly refers to a recommendation. The Amazon online shop, for example, (currently[26]) uses the formulation '*Frequently bought together*'. In technical jargon, this is still called a recommendation system, but it does not perform the act of a recommendation in the sense of the Speech Act theory. Rather, it communicates some information without explicit intentionality.

A recommendation does not have to be followed or even taken into account for the speech act to be successful. The realization of a recommendation is an act in itself. However, the effect of a recommendation, for

[24] https://openai.com/blog/chatgpt, last accessed on July 05, 2023
[25] https://bard.google.com/?hl=en, last accessed on July 05, 2023.
[26] https://www.amazon.com/, last accessed on June 12, 2023.

example, a user being persuaded to buy a product, can also be understood as part of a perlocutionary interpretation of the speech act. A fault of type A.2 occurs when the recommender system is used in a context for which it was not intended. In this case, the person or institution that decided to use the application in the unintended context is responsible for the fault. Users are responsible for a fault of type A.2 if they use the wrong or inappropriate data as input. For faults of type B, the developers are responsible. If functional faults are made in programming, a fault of type B.1 is present. Since an extensive testing process is part of a clean development process, inappropriate or insufficient quality control can be interpreted as an incomplete substitution of the recommendation act and thus corresponds to a fault of type B.2. Since a thorough quality control process should also reveal problems with the choice of data, model, or learning procedure, I also classify these as faults of type B.2. However, it should be noted that developers can potentially only identify formal problems with the data and fix them if necessary (see Data Set Testing in Section 5.3). The decision that the data is suitable and usable for the speech act of a specific recommendation is supported by a thorough testing process and thus by the developers, but ultimately the decision maker is to be held accountable. If a 'wrong' recommendation is deliberately made or a 'correct' recommendation is deliberately not made, there is a fault of type Γ (in the context of recommendations, it might not make sense to distinguish between faults of type Γ.1 and Γ.2). Such faults are not possible in the substitution by a machine since intention is part of them – something that a machine cannot have. However, false expectations can be raised when providing a recommendation system, which implicitly leads to a fault of type Γ. This is the case, for example, if the distributing company (deliberately) miscommunicates the performance of the system or decides to deploy the system even though the quality assurance process shows clear deficits.

The question of who is accountable for the effect of a recommendation leads to a basic moral and philosophical problem: *Is a developer or distributor accountable for the impact of a product, or is it the user?* Legally, it may be possible to find an answer to this question in a specific case. However, the general question of who is accountable for unintended or unforeseeable side effects remains open. These can occur despite a completely successful speech act.

Furthermore, the question arises whether, for example, a programmer in a company is really accountable for faults of type B, or if they are only responsible (in terms of 'being in charge of') for them and the company behind the programmer is accountable. This consideration leads to the question of whether accountability should not generally be clarified contractually as part of the development and distribution process. In this way, complete accountability could initially lie with the developing company. In each phase of the long chain of responsibility, the respective responsible persons could (contractually) agree to be accountable for all explicit faults and consequences that lie within their area of responsibility, insofar as they consider this to be appropriate in the specific case. For all faults and consequences for which no one is accountable in this way, the company would remain accountable. Once the product is ready to

be deployed, the company could (contractually) specify for which faults and consequences it remains accountable and for which the buyers of the product must take over accountability before a purchase can be made. Also, if the buyers are not the users of the product, they can similarly pass on or share accountability with the users. In this way, the question of accountability could be at least partially regulated by the market. However, this consideration should be treated with caution, as undesirable side effects can also occur here. For example, users could be made to become accountable for aspects for which they do not want to be accountable due to a lack of technical understanding.

In any case, the same actor positions may not necessarily have the same name in different companies. In this respect, the question arises to what extent relevant actor positions have to be determined by their function instead of what the actors are called.

RQ 3

How to determine which actors are to be held accountable based on John Austin's Speech Act theory?

Answer: Consider a decision of an AI-based ADM system as a combination of multiple partial acts that together substitute the speech act of deciding upon something or someone. Each of these partial acts may fail or be void due to any of the faults A.1-Γ.2. Following this thought experiment, the developing company could be initially accountable for everything concerning the system it develops and deploys. For each of the faults that can occur in substituting any specific partial act in the software development process, the developing company could link accountability to job positions or project leads (before they are occupied). If no one agrees to be accountable for a specific fault, the company would remain accountable. It could transfer accountability to buyers if they agree to assume it as part of the product they buy. In this way, the market could regulate which actor is to be held accountable. Accountability would become an issue of balancing trust, control, and costs. How this thought experiment would play out in implementation and what undesirable side effects would arise from it is yet to be investigated.

3.3 Possible Forums and their Goals

Different forums have different goals. A governmental organization may want to know whether the current regulation is fit to protect affected persons and society from negative consequences; a legal body may want to know whether legal requirements are met; an NGO may want to know whether societal requirements are met; and an affected person may have very individual goals, like understanding how the system arrives at a decision.

There is no conclusive list of possibly relevant forums. However, ISO 9000:2015[27] refers in Section 2.2.4 to the term *interested parties*, being *'those that provide significant risk to organizational sustainability if their needs and expectations are not met'*. In Section 3.2.3, it lists *'Customers, owners, people in an organization, providers, bankers, regulators, unions, partners or society that can include competitors or opposing pressure groups'* as examples. Busuioc, 2021, p. 827, additionally suggests *'courts, parliamentary committees, ombudsmen, etc. but also purpose-built forums such as AI ethics, standardization, and audit bodies'*.

For governmental organizations, legal bodies, and NGOs, the implementation of mechanisms could be bound to certain conditions, such as a non-disclosure agreement. This would allow such forums to judge a system and communicate the results of the assessment without sensitive information or trade secrets being shared.

A forum without sufficient technical expertise might not be able to handle mechanisms or, worse, may come to hasty and wrong conclusions. This aspect is particularly important for civil society representatives and journalists, as their judgment can lead to unjustified reputational damage for the operator or the company behind the product (see Example 8).

Example 8 (Apple Card)

On November 07, 2019, David Heinemeier Hansson tweeted: *'The @AppleCard is such a [...] sexist program. My wife and I filed joint tax returns, live in a community-property state, and have been married for a long time. Yet Apple's black box algorithm thinks I deserve 20x the credit limit she does. [...]*[28]*'*. The tweet received a lot of attention (over 25k likes) and even led to an investigation by the New York State Department of Financial Services (Vigdor, 2019). In March 2021, the investigation report was published, concluding that Apple Card *'did not consider prohibited characteristics of applicants and would not produce disparate impacts'* (Campbell, 2021). The accusations were therefore not justified, but have led to reputational and thus financial damage for Apple and Goldman Sachs.

Which mechanisms need to be provided for which forums and under which conditions is a decision that also rests with legislators and other regulating bodies. As the field of AI-based ADM systems is relatively new, there is a lack of experience regarding appropriate legal and societal requirements. In many cases, rigid rules would potentially be overly restrictive and industry representatives frequently proclaim the inhibition of innovation. Conversely, these rigid rules would potentially also fail to adequately regulate highly critical applications. To address this problem,

[27] Quality management systems - Fundamentals and vocabulary.

[28] https://twitter.com/dhh/status/1192540900393705474?s=20, last accessed on June 26, 2023.

risk-based regulation is an option much debated (e.g., by the AI Act; see Section 6.1). It means that the risk posed by the use of an ADM system is assessed somehow and that legal requirements are based on the outcome. Several approaches to performing risk assessment are discussed in 6.1.1. Other considerations that regulating bodies can take into account are: *What are the objectives of a forum? Does the legislature consider these objectives worthy of support? Are the current means of achieving the objectives sufficient? Is it appropriate to prescribe further mechanisms for the particular forum? And what are the circumstances that must apply for prescribed measures to be proportionate?*

3.4 Consequences for the Actor

Possible consequences depend at least indirectly on the forum. If the forum is a legal representative asking legally relevant questions, the answers constitute pertinent evidence in the context of a lawsuit, resulting in common possible consequences of legal proceedings. If it is any other forum that comes to a legally relevant conclusion, it can initiate legal proceedings and set common possible consequences of legal proceedings in motion. These consequences include taking remedial (or corrective) action to address the shortcomings, making amends to those affected, and, in the most severe case, imprisonment.

However, if a forum reaches a negative, non-legally relevant conclusion, the possibilities of initiating consequences are severely limited, as in most cases, there is a strong power imbalance between a forum and an actor. In this case, the primary consequence is reputational damage or loss of trust for the actor or the company behind it, and thus economic damage. The more attention and credibility a forum has, the greater the potential reputational damage, which is why it can make sense as an individual to turn, for example, to NGOs or ethics bodies. However, social platforms also enable individuals to cause significant reputational damage or induce legal investigations (see Example 8, p. 80). Taking preventive responsibility and accommodating different forums is increasingly seen as a competitive advantage, as it creates reputational benefits. Thus, it can make sense for a company to offer transparency and inspectability mechanisms beyond what is required by law, despite the associated costs and risks. Such considerations are discussed under the term Corporate Digital Responsibility (for more about CDR, see Section 6.2).

In addition, customers can collectively agree to purchase a product or service only if certain minimum requirements are met, such as compliance with certain standards or having certain certifications (for more on certifications, see 4.1.1). In the context of medical devices, for example, compliance with standards is required by law.[29] In other cases, compliance with standards is theoretically optional but required by the market.

[29] For example, Regulation (EU) 2017/745 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002, and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC, Article 6.

For example, there are insurers who require conformity with certain standards beyond legal obligations.[30]

Psychological research on the topic of accountability shows that accountable actors strongly adapt their behavior to forums they are familiar with (Lerner & Tetlock, 1999, p. 256). What is at least partly discussed in psychology as a criticism of accountability could represent an interesting mechanism for generating more initiative for actors to develop new and creative solutions to meet the expectations of various forums, even if it does not entail any immediate economic benefit or is legally required. This consideration can be put into practice, for example, by using so-called Assurance Cases, which will be described in Section 4.2. At the same time, actors who are accountable to an unknown forum display much more recognition of value trade-offs when evaluating controversial issues. Lerner and Tetlock, 1999, p. 257, also show in their literature review that actors align themselves more strongly with the subjective needs of explicit forums and communicate their information accordingly in a targeted (if necessary also manipulative) manner if they are aware of them, thus placing the needs of unknown forums in the background. Furthermore, accountability mechanisms can lead to actors focusing on justifying their past decisions instead of reflecting on them and remedying shortcomings or making better decisions in the future. Considering such effects, accountability is not a social panacea. Therefore, it is important for concrete implementations of the mechanisms proposed here to be accompanied by scientific research to ensure that the desired effects do occur and that undesirable side effects at least remain within acceptable limits.

[30] For example, the regulations of the DGUV (German Social Accident Insurance, "Deutsche Gesetzliche Unfallversicherung" in German) are issued by the respective accident insurance institution responsible and taken into account in legal cases (see https://www.dguv.de/de/praevention/vorschriften_regeln/vorschriften/index.jsp, last accessed on June 26, 2023).

Chapter 4

Auditing

According to Bovens, accountability requires '*to pose questions and pass judgment*' (Bovens, 2007, see Definition 8, p. 28). However, many questions about an ADM system cannot be answered directly by an actor. To solve this problem, static information can be disclosed (transparency) and access to the system and its data can be granted (inspectability) (Hauer, Krafft, & Zweig, 2023). This allows a forum to 'pose' their questions directly to the disclosed information and the accessible system. Such a process can be understood as an *audit* (see Section 4.1). However, there is more to auditing than just the process itself. It must be clear what exactly is being audited, what the objectives are, and what methods are appropriate for demonstrating that the objectives have been achieved. Consider the objective of ensuring that an ADM system is fair. This leads to a number of challenges as described in Section 2.3.3. In order to meet these challenges, the Assurance Case framework will be introduced in Section 4.2.

4.1 Audit Definitions

In the context of assessing software systems, a distinction is often made between white-box and black-box systems (Loyola-Gonzalez, 2019). White-box systems are systems where insights into inner mechanics are possible. This may refer to an open code base, an interpretable model, a well-defined process, and so on.

Black-box systems are systems where such insights are not possible (see Figure 4.1). With regard to auditing possibilities, black-box systems can be differentiated into four groups:

- Systems for which only the output can be observed (Diakopoulos, 2014; see Figure 4.2 a)).
- Systems for which only some inputs can be observed but some inputs are unknown (Krafft et al., 2023; see Figure 4.2 b)).
- Systems for which all the inputs and resulting outputs can be observed (Diakopoulos, 2014; see Figure 4.2 c)).
- Systems that are white-box systems in principle, but which are too complex to be comprehensible to humans and therefore need to be treated as black-box systems (Ananny and Crawford, 2018, pp. 977-978, Rudin, 2019b, Appendix A, see Figure 4.2 d)).

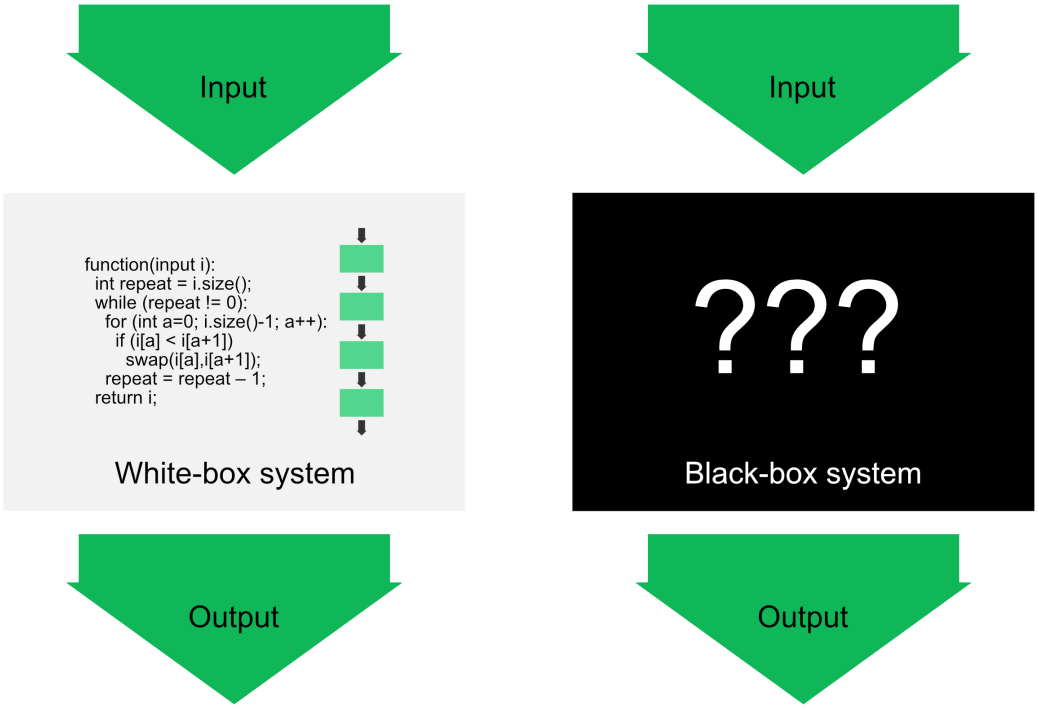


Figure 4.1: The inner mechanics of a white-box system (left) are visible and can be examined. The inner mechanics of a black-box system (right) are unknown. Conclusions about them can only be drawn by comparing outputs with their respective inputs.

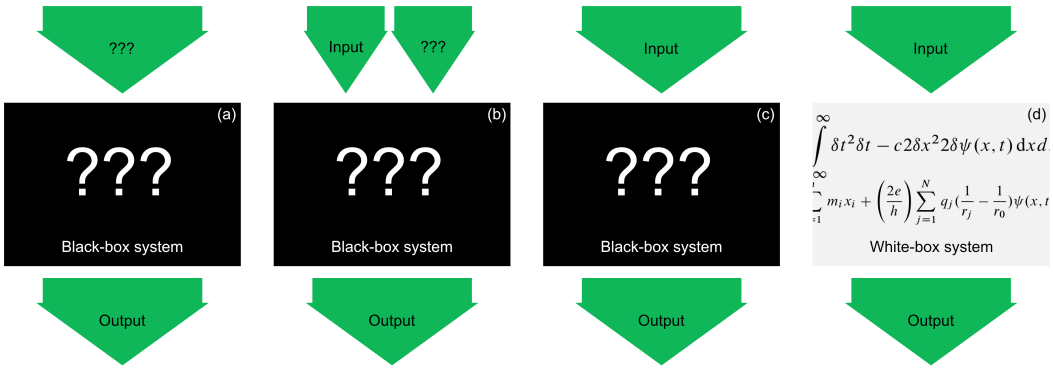


Figure 4.2: The four distinct types of black-box systems. From left to right, they are decreasingly (but still) challenging to audit.

Furthermore, from an auditor's point of view, systems can be differentiated according to which accesses and insights they have to the system, regardless of whether it is a white-box or a black-box system (see Figure 4.3). Depending on this, they may require additional accesses or insights from the provider or may need to use auditing methods explicitly designed for limited access and insight.

It is important to be aware of the different understandings of the term 'audit', as a divergent understanding can quickly lead to people agreeing on actions without realizing they are talking about completely different

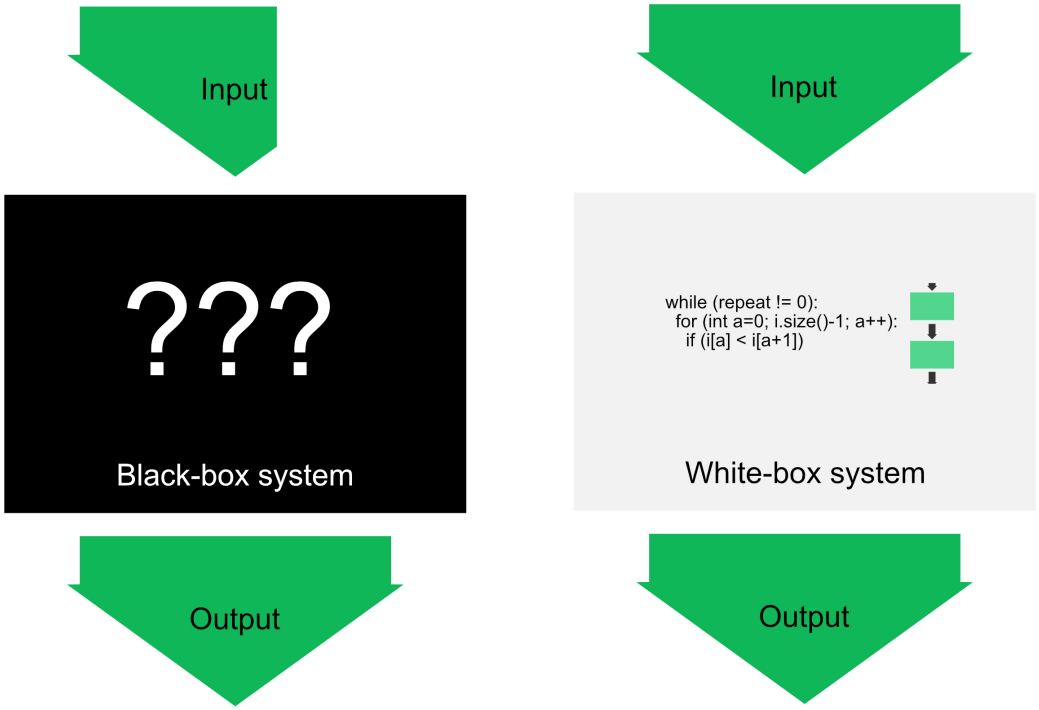


Figure 4.3: Auditors may have limited access (left) and/or limited insights (right).

things. There are two different understandings of the term 'auditing' that are relevant for this work (Lovelace & DataKind, 2020): regulatory inspections based on standardization – for example, the ISO standardization documents corpus (see Section 4.1.1), which is the basis of accredited certification, and bias audits of platforms (see Section 4.1.2). As both kinds of audits clearly address different forums (and there are further understandings; see, for example, Gaddis, 2018 and Vecchione et al., 2021), special care must be taken when representatives of these different forums communicate with each other.

4.1.1 Audit According to ISO

According to ISO 19011[1], an audit is a '*systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled*'. This definition is the basis of all references to audits throughout the whole ISO standardization documents corpus. It explicitly differentiates between 1^{st} party audit, 2^{nd} party audit, and 3^{rd} party audit (see Figure 4.4).

1^{st} party audits, also called *internal audits*, '*are conducted by, or on behalf of, the organization itself*'.[2] They are only of limited value for most forums, as they are carried out within the company and thus, there is an incentive and the possibility to ensure good results by manipulating the procedure when used for something different than internal quality assurance. They are therefore suitable for building internal trustworthiness but

[1] Guidelines for auditing management systems.
[2] ISO 19011 3.1 Note 1 to entry.

contribute only to a limited extent to external trustworthiness. However, such audits can already be performed before deployment of a system.

2nd party audits 'are conducted by parties having an interest in the organization, such as customers, or by other individuals on their behalf'. 3rd party audits 'are conducted by independent auditing organizations, such as those providing certification/registration of conformity or governmental agencies.'[3] Thus, 2nd and 3rd party audits are also called *external audits*. All these forms of audits are either based on a quality assurance process in the interest of an external company (e.g., a contractor) or on legal interests of a statutory body. This means that an auditor is granted access to all relevant information and systems. The validity of such audits, however, depends heavily on the trustworthiness of the auditing personnel and the audit process carried out.

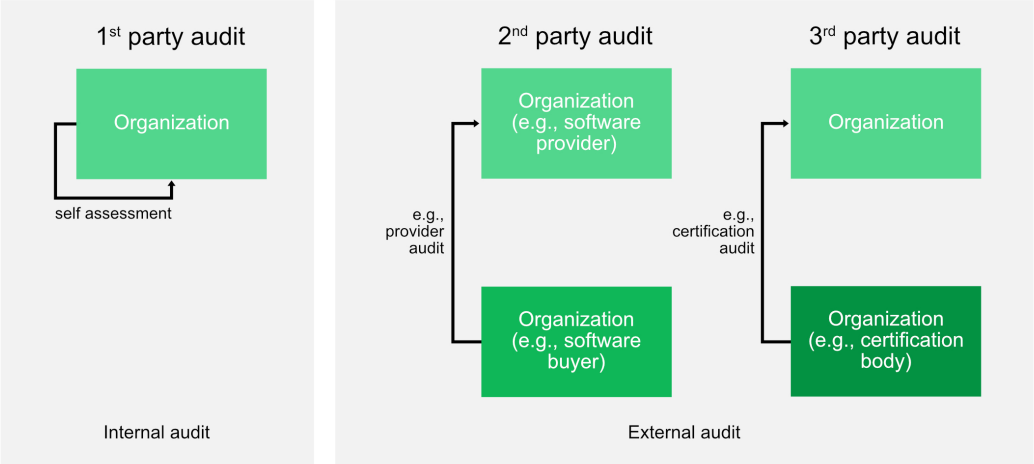


Figure 4.4: Depending on which organization performs an audit and how it is related to the organization to be audited, the audit is referred to as 1st, 2nd, or 3rd party audit.

Successfully completed audit processes (according to the ISO understanding of audits) with positive results can be confirmed with so-called certificates.

Certification

A certificate has as much meaning as there is confidence in the issuing institution and its methods. In theory, any individual can issue certificates at will. However, these do not have much meaning as long as the issuing person cannot at least prove to be sufficiently trained. Furthermore, it is difficult to judge whether the person, despite being sufficiently competent, carries out an auditing process with adequate care and passes on the results truthfully and impartially. Usually, certificates are issued by persons who work for organizations that have earned the necessary trust. A distinction is made between proprietary certificates (e.g., company certificates, producer certificates, or certificates aimed at the use of a specific product) and certificates issued by bodies whose competence

[3] ISO 19011 3.1 Note 2 to entry.

has been confirmed by a third, independent party. Accredited certifications (see Definition 17) are the most meaningful, as any institution that is allowed to issue accredited certificates (a so-called *accredited certification body*) has been attested by the respective national *accreditation body*.

Definition 17 (Accredited Certification)

According to EU Regulation No. 765/2008[4], accreditation is defined as *'an attestation by a national accreditation body that a conformity assessment body meets the requirements set by harmonised standards and, where applicable, any additional requirements including those set out in relevant sectoral schemes, to carry out a specific conformity assessment activity'*. The European legal framework ensures that there is exactly one accreditation body in each country of the European Union (in Germany, this is the DAkkS[5]), which confirms the competence of certification bodies.

Accredited certification has to be based on standards, as provided by DIN[6] and DKE[7] at the national level in Germany, CEN[8], CENELEC[9], and ETSI[10] at the European level, and ISO[11], IEC[12], and ITU[13] at the global level.

The states of the European Union have also created a legal framework[14] that demands accreditation by a conformity assessment body for sensitive areas, like the safeguarding of critical infrastructure (gas, water, and electricity supply) against IT-based attacks.[15] Therefore, accredited certifications are internationally recognized. This characteristic is particularly important when considering the modern global market in which IT systems are distributed and operated.

A distinction is made between the certification of processes (in the context of certification, this is referred to as the certification of *management systems*, ISO 17021[16]), products (ISO 17065[17]), persons performing audits (ISO 17024[18]), and testing and calibration laboratories (ISO 17025[19]). This means that in the context of an AI-based ADM system, for example, the ADM system itself, responsible actors, and the quality assurance process could be certified (Heesen et al., 2020).

The certification of software systems is particularly challenging, as a product is certified as is, but changes to software (updates and/or upgrades) are usually made on a frequent basis. To counter such problems, the validity of each accredited certificate is limited in time. Depending on the standard according to which certification is performed, its validity counts for three (e.g., ISO 17021) to five years (e.g., ISO 17065). Before the end of this period, a surveillance audit is required every year, as well as a reassessment at the end to maintain the validity of the certification. A particular challenge is posed by AI-based ADM systems that continue to learn as they are used, i.e., their decision-making structure changes continuously. The solution to this problem could lie in implementing a continuous monitoring strategy (see *Field Testing* in Section 5.5) as a certification requirement. Since the standards landscape necessary for accredited certification is still developing (see, e.g., DIN/DKE, 2020, 2023), there are no accredited certificates for AI-based ADM systems yet.

If non-statutory 3^{rd} parties, like NGOs, journalists, or scientific bodies, want to assess a digital product or service to verify a suspicion or accusation of misbehavior, they might not be provided with any transparency and inspectability mechanisms beyond those offered by the operator to

[4] Chapter I(10) and chapter III of Regulation (EC) No 765/2008 of the European Partliament and of the Council of 9 July 2008.

[5] German Accreditation Body ('*Deutsche Akkreditierungsstelle GmbH*' in German). In Germany, the mandate of the DAkkS is legally anchored by the Accreditation Body Act ('*Gesetz über die Akkreditierungsstelle (Akkreditierungsstellengesetz - AkkStelleG)*' in German).

[6] German Institute for Standardisation, registered association ('*Deutsches Institut für Normung*' in German).

[7] German Commission for Electrotechnical, Electronic & Information Technologies of DIN and VDE ('*Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE*' in German.

[8] European Committee for Standardization.

[9] European Committee for Electrotechnical Standardization.

[10] European Telecommunications Standards Institute.

[11] International Standardization Organization.

[12] International Electrotechnical Commission.

[13] International Telecommunication Union.

[14] Regulation (EC) No. 765/2008.

[15] In Germany regulated by the Energy Industry Act §11 ('*Energiewirtschaftsgesetz*' in German).

[16] Conformity assessment - Requirements for bodies providing audit and certification of management systems.

[17] Conformity assessment - Requirements for bodies certifying products, processes and services.

[18] Conformity assessment - General requirements for bodies operating certification of persons.

[19] Testing and calibration laboratories.

anyone else. To address such scenarios in the context of discrimination on Internet platforms, Sandvig et al. defined a different kind of audits.

4.1.2 Audits According to Sandvig et al.

Sandvig et al. understand (software) audits as some sort of field experiments tailored to a specific platform, containing multiple tests, and, most likely, an additional software apparatus for experimentation (Sandvig et al., 2014). They provide a categorization of five different kinds of audits based on the process of how the inputs are provided: (i) Code Audits, (ii) Non-Invasive User Audits, (iii) Crowdsourced Audits, (iv) Sock Puppet Audits, and (v) Scraping Audits (see Figure 4.5).

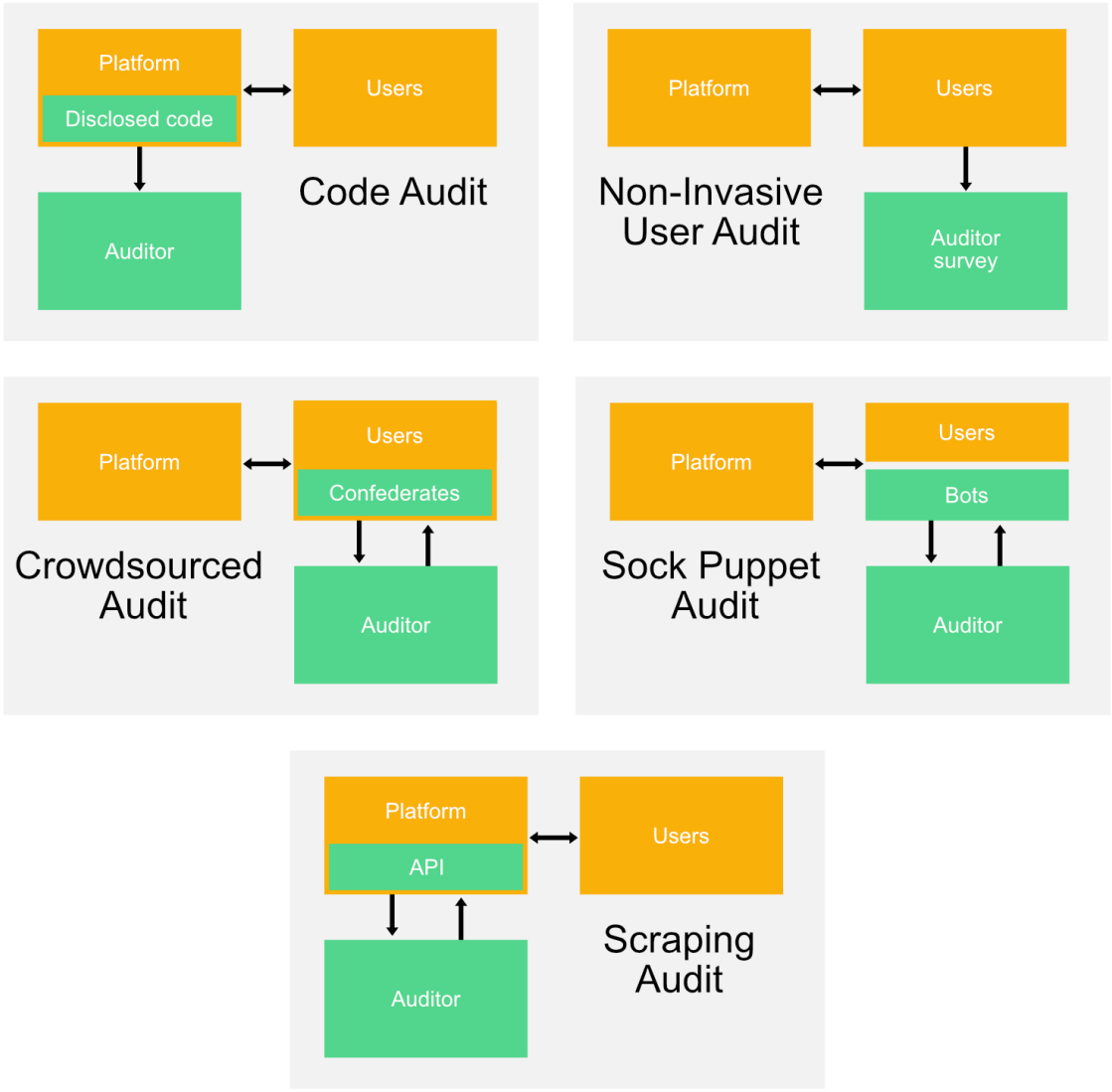


Figure 4.5: Visualization of the five different kinds of audits based on Sandvig et al., 2014.

Code Audit means that an auditor receives a copy of the relevant code. This can be considered to be a white-box audit, as the audit is based on internal information. In the context of standardization-based audits (such as ISO 19011); this is sometimes also called code review (Baum et al., 2016). The benefit of code audits is highly controversial. On the one hand, some argue that they introduce additional risk potential, as people might be able to scope the code for loopholes in order to attack the system or exploit it for their own benefit (Weller, 2019, p. 29). Additionally, algorithmic transparency is frequently not provided due to a claimed risk to trade secrets (Brauneis & Goodman, 2018, pp. 153-159). One solution might be to share the relevant code base only with trusted parties, for example, based on a non-disclosure agreement. In such a case, in particular, a code audit according to Sandvig et al., 2014, is the same as a code review during a 2^{nd} or 3^{rd} party audit according to ISO 19011. On the other hand, actual product code can be complex and hard to understand without investing an impractical amount of time, and may even be impossible for a single person (Camp, 2006). Furthermore, the system behavior might not only be defined by its code base alone, but also by the data fed into it. The system behavior could also be non-deterministic due to random elements, like the order of data to be processed.

These and other challenges lead to the idea of considering a software system as a black box, regardless of whether it really is one, and basing an audit only on system inputs and resulting system outputs. In many cases, there is no other choice anyway, as a software provider will not necessarily agree to making relevant parts of the code available. The other four audit procedures are explicitly designed for the inspectability of black-box systems.

In a *Non-Invasive User Audit*, the interactions of users with the system and how the system reacts to them are evaluated. This can take the form of user surveys or an analysis of a user's input and the system's subsequent output. In any case, the auditing entity has no influence on the user's input, meaning that the analysis of the data is limited to manual review of individual cases and statistical evaluations. This method is a last resort when other means of auditing a system are not possible. For example, it could be used to check whether men generally get more or better suggestions of open job positions than women.

It is also suitable for assessing how the system is perceived by users. However, it is not suitable for examining the reasons for unequal treatment. For example, if the cause of different treatments lies not in the user input at all, but in the user profile, which is not necessarily known to the auditor, this connection can hardly be detected with a non-invasive user audit. Such an analysis would require targeted experiments, which can be offered by the other three kinds of audits.

A *Crowdsourced Audit* is also based on the participation of actual users. Instead of real user behavior, participants enter predefined queries or let a program use their profile and interface to automatically enter predefined inputs. Thus, it can also be considered to be an *Invasive User Audit*. Reber et al. used this technique to assess whether patients were being actively

targeted with advertisements for unproven stem cell therapies by reviewing the advertisements after entering keywords such as 'Parkinson' into the Google search engine (Reber et al., 2020). Another example is the assessment of the impact of personalization on a search engine performed by Krafft et al., 2019. They developed a browser plugin for participants to install, which collected the first page results on Google for 16 search terms every four hours for around one month for more than 4,000 participants.

For a *Sock Puppet Audit*, human interaction is simulated by a program. Programs that behave as though they were human, or use the same interfaces as a human would, are called *bots*. The behavior of bots can be modified at any time, which allows a high degree of controlling inputs for experimental setups. However, this kind of audit has two limitations. First of all, information about typically human behavior, like mouse cursor acceleration and positioning, might be part of the input information considered by the system under test (Amazon, for example, saves at least every click, scroll, and mouse movement of registered users).[20] Second, many systems and platforms prohibit the use of bots for various reasons (e.g., as bots could be used to manipulate other users or affect the system behavior for all users; see, e.g., Orabi et al., 2020). Often, bot detection and prevention mechanisms are installed that block accounts that seem to be controlled by a program alone. Interestingly, the exact same input behavior would not be prohibited if it were to come from a human.

Scraping Audits are based on previously defined queries that are automatically transferred to the system, for example, via an API or a browser control system (e.g., Selenium[21]). If the system under test provides an API, this kind of audit is the easiest to implement; however, it does not use the same interfaces as human users. Thus, this form of audit is not appropriate if a specific user behavior, like mouse cursor movement or delay times due to a user reading and processing information, is part of the system input. It may also be the case that interactions via API show a different behavior than actual user interactions (Diakopoulos, 2014, p. 17). Since the query can be modified as desired (also depending on previous responses), it also allows a high degree of controlling inputs for experimental setups. Hauer et al., 2020, compared the h-indices of hundreds of the world's most renowned scientists as provided by various platforms. The data was retrieved by performing a scraping audit.

In general, there needs to be some way to feed input data to the system. Optimally, the auditor can feed the data and inspect the resulting outputs on their own. This would provide the highest degree of credibility and flexibility for testing purposes. For those test methods in which input queries can be fully defined before the test starts, it does not matter who feeds the inputs; it could also be an internal mediator performing queries on behalf of an external tester. On the one hand, the operator can then

[20] According to https://www.wired.co.uk/article/amazon-history-data (last accessed on February 08, 2023), I requested all information Amazon had collected from and about me and can confirm this.

[21] Selenium is a framework for automated software testing of web applications; see https://www.selenium.dev/, last accessed on February 08, 2023.

save the effort of creating a secure interface for external parties and may also see better protection of trade secrets. On the other hand, it must be noted that the testing entity can hardly check whether the tests have been carried out conscientiously and whether the correct results are provided for verification. Test processes thus become more credible if both the construction of a test data set and the execution of the tests based on it are carried out by an external, independent entity (e.g., via API). This requires the testing entity to have enough access to the model to submit input data and inspect the respective outcomes. Additionally, some methods are not suitable for being mediated, especially those that require the modification of queries depending on previous results.

In 2020, the private television station Rhein-Necker Fernsehen (RNF) told us[22] that they had received complaints from subscribers on Facebook that the news feed did not pass on the entire bandwidth of contributions to its subscribers. This is quite justified, as the quantity of contributions that appear on the news feeds must necessarily be limited. At the same time, however, they also complained that they primarily received contributions about accidents, crimes, etc., and not the usual mix, which also contains weather forecasts, politics, or curiosities. Whether this complaint was justified or not could not be easily verified. As described in Krafft et al., 2020, we wanted to perform a black-box audit according to Sandvig et al., 2014, to investigate the accusation. A *Non-Invasive User Audit*, a *Crowdsourced Audit*, and a *Scraping Audit* were ruled out from the start due to privacy concerns. Furthermore, a *Scraping Audit* was no longer possible at that time, as Facebook had only recently restricted the possibilities of accessing information via API for all those who were not explicitly authorized by Facebook to pursue research questions with the help of the API. A request for authorization remained unanswered. This left us only with the option of performing a *Sock Puppet Audit*.

To validate whether the approach works, we conducted a pre-study. Following the elaborations of Z. Yang et al., 2014, regarding bot detection practices, we manually generated 30 fake accounts from various IPs, based on email addresses from various providers, to increase the chances of our bots of remaining undetected. Each account was manually set up to follow only the RNF Facebook group. In parallel, we built a software that logged each account in, scrolled through the respective news feed of that account, and saved the displayed posts in a database. After four days, the first bots were recognized as such and banned. After ten days, all bots but one were banned. The data we had collected up to that time showed that for no pair of accounts was the same selection of news displayed on the same day, independent of their order. Further insights were not possible with such little amount of data. However, this use case shows how limited black-box auditing approaches might be if the provider of the system to be audited does not support the investigation.

RQ 4

What kinds of audits are applicable for analyzing AI-based ADM systems?

[22] In Section 4.1.2, 'we' and 'us' refers to the authors of Krafft et al., 2020.

Answer: 1^{st}, 2^{nd}, and 3^{rd} party audits all play a different, but relevant role regarding the analysis of AI-based ADM systems.

1^{st} party audits (internal audits) are conducted by an organization itself or on its behalf. They are suitable for judging a system within a company, for example, before it is released. 2^{nd} party audits are performed by parties with an interest in the quality standards of a particular organization (such as customers, or others on their behalf). Traditional 3^{rd} party audits are conducted by independent auditing organizations, such as those issuing certificates or registrations of conformity, or by government agencies. In all of these cases, the auditing personnel requires access to all relevant information and systems in order to examine a product.

A different kind of 3^{rd} party audits assumes that the auditing personnel does not have any access or insights beyond those offered by the operator to anyone else, such as NGOs, journalists, or scientific bodies. Audits that can be performed in such cases are non-invasive user audits, crowdsourced audits, sock puppet audits, and scraping audits. Which of these are applicable depends on various factors, such as whether enough supporting users can be acquired, whether bots can be deployed, or whether API access is available.

While the audit terminology of ISO considers the evaluation of documents (transparency mechanisms) and experimentation with the system under test (inspectability mechanisms), the audit understanding of Sandvig et al., 2014, clearly focuses on the latter. In both cases, some kind of judgment must be made as to whether or not the information gathered (or tests based on it, which will be discussed in detail in Chapter 5) meets the (non-functional) requirements. The Assurance Case framework can provide guidance to answer this question.

4.2 Assurance Case Framework

The Assurance Case framework is a structured method for arguing why a particular collection of evidences is suitable for assuring that a claim about the non-functional properties of a system is true. It is a method that is widely used in the field of safety engineering (see, e.g., ANSI/UL 4600[23]). In this context, it is sometimes also referred to as *Safety Case* (Kelly et al., 1999, p. 121). Accordingly, there are already many extensions, further supporting techniques, and testimonials for arguing safety claims (Rinehart et al., 2017).[24] Assurance Cases have a strong resemblance to *Argumentation maps*, a method used by philosophers for at least 180 years to structure critical thinking (see, e.g., Whately, 1834, p. 415).

[23] Standard for Safety for the Evaluation of Autonomous Products.

[24] Figure 1 and Section 5.1.3 of Rinehart et al., 2017 show that the development of Assurance Cases goes back to at least 1965 and has been used for assuring safety in diverse fields of application.

For some years now, there has been growing interest in using the framework for other non-functional and extra-functional requirements (see Definition 14, p. 50) as well. For example, the concept was introduced in the standardization roadmap AI 2.0 of the German Standardization Council (DIN/DKE, 2023). Porter et al., 2022, suggest building Assurance Cases to argue for conformity with general ethical values in the context of autonomous systems, and we[25] suggest using them to specifically argue the assurance of fairness of a system.

The main task of an Assurance Case (AC) is to justify why and under which assumptions a collection of evidences implies a claim. The argumentation is built as a hierarchically structured tree. The root node contains the *main-claim* to assure, for example, that '*The AI-based system is fair*'. What is understood by fairness can vary greatly between different stakeholders. It could refer exclusively to aspects of non-discrimination, as already discussed in Section 2.3. However, aspects of autonomy, stability, transparency, or control can also be understood as important facets of fairness (see, e.g., Hauer, Müller-Kress, et al., 2023, p. 5). To be able to argue that the whole AI-based system is fair, it is thus necessary for all aspects of fairness that are relevant for the stakeholders to be assured. Which aspects are, in fact, relevant could be investigated by interviewing the stakeholders. Based on such a so-called *argumentation* or *reasoning step*, the main-claim can be decomposed into multiple smaller *sub-claims*, such as '*The AI-based system is fair in terms of stability for the persons affected by decisions of the system*', '*The AI-based system is fair in terms of treating no individual significantly worse than others*', and '*The persons affected by the decisions of the system have the option to challenge the decision*'. The argumentation can be supported by additional *contextual information*, such as '*The system manages the assignment of auditors to audit requests for complex software systems. This sometimes requires the auditors to spend several days at the client's site, which is potentially far away. The auditors want to know when they need to go where early on, so that long-term planning of their private lives is possible*'. Such information allows anyone who inspects the Assurance Case to understand the argumentation, independent of what they knew about the system in advance. Furthermore, implicit *assumptions* made in the argumentation can be made explicit, such as '*Based on the stakeholder interviews, all relevant aspects of fairness have been identified*'. Such assumptions may be appropriate, perhaps even necessary, at the time the Assurance Case is prepared, but by making them explicit, they can be easily challenged at any time, for example, if the software is functionally upgraded or is to be used in a context not considered before.

Each of these sub-claims can be deconstructed into further sub-claims, based on an argumentation, until the sub-claims each can be supported by *evidences*. Evidences could be test results, such as the computation of statistical values and fairness measures (in Section 5.4.2, this will be discussed in more detail), the software specification that documents its functionality (e.g., the functionality to anonymously post unfiltered and publicly available feedback on the systems decisions), other documents

[25] Here, 'we' refers to the authors of Hauer, Adler, and Zweig, 2021.

and processes that are suitable for supporting a claim and can be falsified, or other documents that may serve as evidence (e.g., external assessment results).

In summary, the process of writing an Assurance Case is about arguing that the main-claim holds, given that the required evidences can be provided.

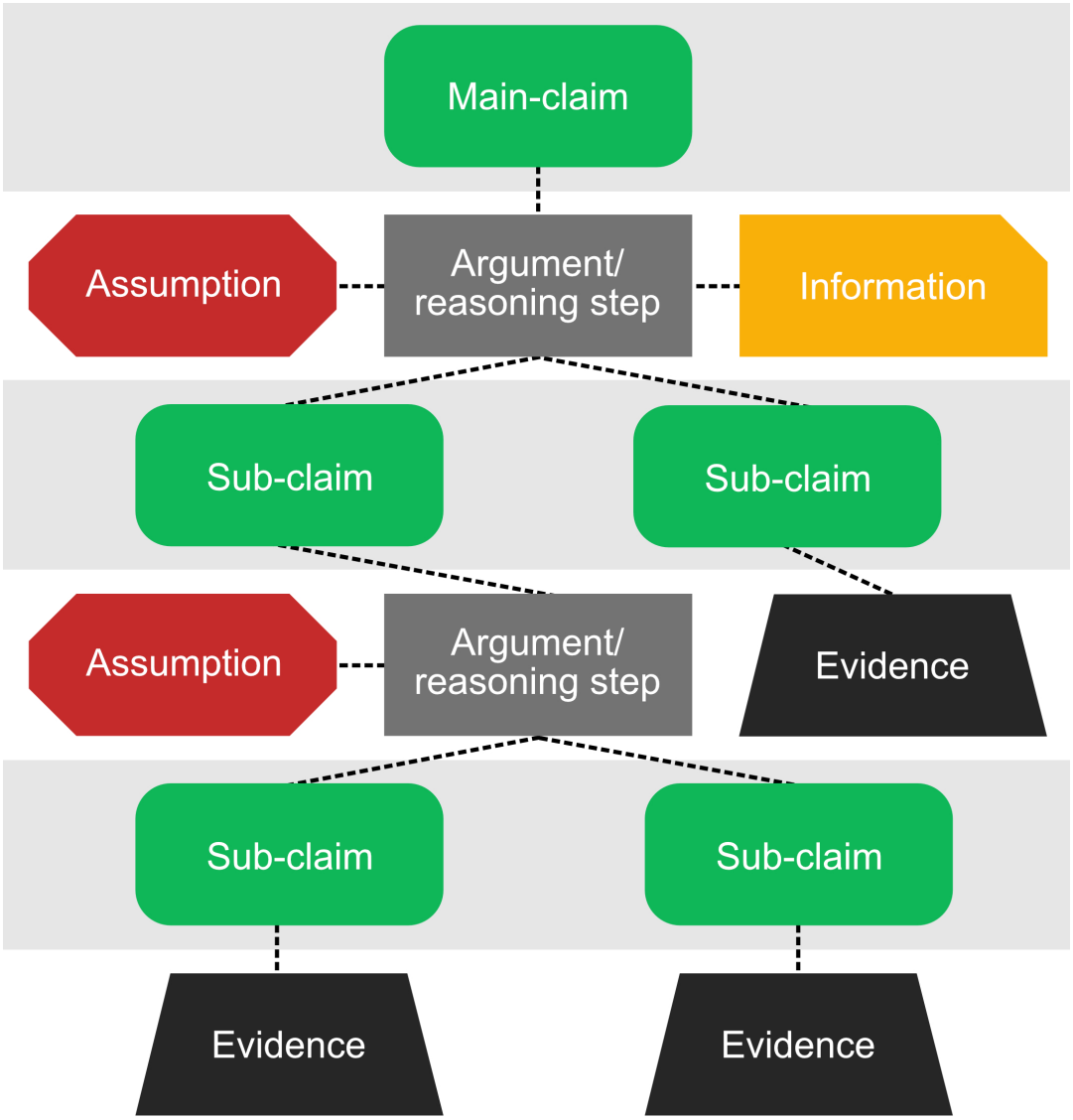


Figure 4.6: Generic representation of an Assurance Case.

Several notations are available for modeling an Assurance Case. Most prominent are variations of the *Goal Structuring Notation* (GSN) (Maksimov et al., 2018). The GSN is a graph representation that consists of at least three types of nodes (goals, strategies, and solutions). These can be translated into any number of specific elements required for a given task (developing an Assurance Case) (Wilson et al., 1996). The connections between nodes (so-called *edges*) can be directed (each edge has one source node and one target node) or undirected (each edge is connected to two nodes). Whether the edges point from the main-claim to the evidences or

vice versa is not important, as both versions tell the same story, just from different perspectives: If the edges point from the main-claim to the evidences, the Assuracne Case can be read as 'Based on the argumentation, the main-claim can be divided into sub-claims' and 'The sub-claim is supported by the following evidences'. If the edges point from the evidences to the main-claim, the Assurance Case can be read as 'Together, these evidences imply the fulfillment of this sub-claim' and 'Together, these subclaims imply the fulfillment of the main-claim'. Thus, undirected edges are a viable choice as well. Figure 4.6 shows a generic Assurance Case based on the GSN that includes every standard element at least once. As long as all relevant components are used in a meaningful and uniform manner, the details of the visual design (e.g., which shape represents which element) do not play a significant role. However, it makes sense to choose distinct shapes and colors to ensure a good overview in the Assurance Case. It also makes sense to add colored lines between levels of depth, which assist in keeping the positioning of elements consistent (Kunze et al., 2023).

When an assurance Case becomes too large to manage, each subclaim can be considered as a separate Assurance Case and can thus be developed independently by a second group of experts. Parts that are only relevant to a particular stakeholder group are particularly suitable for such a separation (Rinehart et al., 2017, p. 39).

As the idea of using this framework for non-safety engineering contexts is new, we[26] developed and tested an iterative Assurance Case construction process consisting of eight phases:

> **Phase 1 – Identify Relevant Stakeholders:** Building a thorough assurance argumentation requires awareness of each relevant stakeholder group. In this context, the term stakeholder is used in a broad sense, i.e., meaning any group of people who affect or are affected by the software product in some way. Usually, this means at least a subset of *business experts/customer representatives/product owners, developers, testers, customers, users, and other people affected*. In the sense of Bovens' accountability theory (see Definition 8, p. 28), the relevant stakeholder groups are all relevant actors (see Section 3.2) and forums (see Section 3.3). The most appropriate choice of stakeholders always depends on the specific application context.
>
> **Phase 2 – Assemble an Assurance Case Development Team**: In order to develop an Assurance Case, the identified stakeholders should be able to participate in the Assurance Case development process, as each stakeholder generally has a different perspective. Their needs and concerns can be explored in advance, for example, through interviews, if it is not possible for a stakeholder group to participate. In addition to the stakeholder groups already mentioned explicitly, other participants with specific expertise may be needed, if not already available, such as ethicists, lawyers, and public relations experts. In Section 6, the IEEE standard P7000[27] proposes several

[26] Here, and for the rest of this Section, 'we' and 'our' refers to the authors of Hauer, Müller-Kress, et al., 2023.

[27] IEEE Standard Model Process for Addressing Ethical Concerns in System Design.

key roles in ethical value engineering that may provide additional input and could thus be considered as well.

The support of a facilitator who manages the Assurance Case development process and is experienced in this role may also be useful. This person needs to ensure that the method is used correctly and that all elements of the Assurance Case are described adequately. The facilitator also needs strong communication and listening skills, the ability to facilitate discussion, resolve conflicts, and adapt to different situations, and a high degree of creativity to keep the discussion interesting. Last but not least, the facilitator must have strong organizational skills to keep track of tasks such as setting new meetings, preparing templates, and inviting all relevant persons. Preferably, the facilitator should not be involved in the development of the AI-based system, so that they are not tempted to deliberately avoid any discussion of known problem areas.

Phase 3 – Prepare a Platform: An Assurance Case may grow large, both in terms of width and depth. The right tools are needed to deal with these dimensions. Collaborative whiteboard platforms (such as *Miro*[28] or *Conceptboard*[29]) are particularly suitable for this purpose. For larger companies, it may make sense to develop their own tools, tailored to their specific needs.

Phase 4 – Prepare a Template: In the early stages, the task of developing an Assurance Case can be overwhelming, especially if a large number of people are trying to contribute their thoughts at the same time. Therefore, it is advisable to have a single person or a small group of people produce an initial draft which can be built upon later. It is likely to be heavily refactored sooner or later anyway, so discussing details with a larger group at the initial drafting stage is time-consuming and of limited value.

As of now, there are no best practices on how to systematically develop an Assurance Case for non-safety objectives. In our experience, a concern-driven approach works well. This means talking with all stakeholders about their concerns and handling each of them in a separate sub-claim right under the main-claim as a starting point. As refactoring is part of the development process, the initial structure may change later on: Multiple concerns may be addressed by the same sub-claim or by a sub-claim on a deeper level (e.g., employees and Works Council representatives will most likely express similar concerns). However, having all identified concerns already addressed in the template is a strong starting point for future collaboration with all participants.

If there are other Assurance Cases for other applications in the company, it may be possible to reuse parts of them in order to take advantage of the thoughts and experience underlying them. Preparing a short summary of what each visual element stands for within the Assurance Case can help to ensure their consistent usage.

[28] https://miro.com/, last accessed on June 26, 2023.
[29] https://conceptboard.com/, last accessed on June 26, 2023.

Phase 5 – Empower Participants to Contribute Asynchronously: Over the course of regular meetings, participants become increasingly skilled, first in creating assurance Cases and second, in understanding and incorporating the perspectives of the other stakeholders. Thereby, they become able to work on (at least parts of) the Assurance Case independent of the other stakeholders. This helps to increase productivity and allows for better management of time and people. Changes made by individuals without discussing them with the group can be highlighted, for example, by coloring in the visual elements. At its regular meetings, the group can discuss whether to keep, modify, or remove these changes. Such an asynchronous approach requires a clearly defined process that is known and followed by all participants.

Phase 6 – Meet on a Regular Basis: Regular workshops are required to gradually expand, concretize, question, and refactor the Assurance Case. They are also necessary to discuss and harmonize asynchronous contributions from individual participants. Limiting sessions to two to four people at a time might be a good choice, as too many people at once will make the sessions less productive. If more than four people express a need for discussion, it makes more sense to plan several sessions on smaller topics. This also helps to keep the sessions short, which is beneficial for the necessary concentration and critical thinking skills.

Phase 7 – Check the Required Evidences: As soon as the Assurance Case is ready, it can be assessed to what extent evidences can be provided and documented. If an evidence is not yet available, for example, because the Assurance Case was developed before or during system development, it is advisable to highlight it (and the (sub)claims that cannot be assured due to the missing evidence) visually, for example, by color coding. This makes it possible to see at a glance which claims are already supported by evidences and where evidences are yet to be provided.

Phase 8 – Revise: The Assurance Case can be revised periodically during system development or after deployment, as previously missing evidences become available, the software design is updated, updates are planned, the context changes, or a better argumentation for assuring a claim emerges. Regardless of whether the Assurance Case needs to be adjusted when an AI-based system is updated, after the update it must be ensured that the required evidences can still be provided.

We developed this process as part of the interdisciplinary project *fAIr by design* (see Section 1.2.1) together with the companies winnovation consulting gmbh, Rania Wazir e.U., and rotable to provide a strong fairness argumentation for the software that allows hospitals to automate their clinical rotation scheduling process for clinical clerkships and residents developed by *rotable*. Clinical rotation means that each doctor in training must complete their practical training for a minimum fixed period of time in multiple medical areas. This requires planning their training in specific medical departments at multiple hospitals.

Application of the Assurance Case Framework for Arguing the Fairness of a Medical Rotation Planner

At the start of our cooperation, the project members identified students, doctors, and administration personnel of hospitals as the relevant stakeholders, whose perspectives might not be sufficiently represented among the project members themselves (Phase 1). Interviews with those stakeholders revealed the following aspects of fairness as relevant:

- Autonomy, i.e., students want their preferences for hospitals and departments to be taken into serious consideration. They also want to be able to give feedback about the plan.
- Stability, i.e., students want to be able to rely on staying at a hospital for a minimum fixed period of time.
- Low waiting times, i.e., all students should have the opportunity to complete their education in roughly the same amount of time.
- Long-term planning, i.e., students want to be informed early enough about the further course of their training in order to be able to plan accordingly.
- Non-discrimination, i.e., students do not want to be treated at a disadvantage based on their gender, language, or personal relations with doctors and planners. There are no specific guidelines provided by laws or standards, just a vague obligation to protect against discrimination.
- Transparency, i.e., students want to have an idea of how plans are developed, what information is considered in their creation, which people have an influence on their creation, and what processes are in place to ensure their quality.
- Control, i.e., all stakeholders want access to statistical information that allows inspection of the degree to which the requirements are fulfilled. They also want to be able to give feedback about the information provided.

Note that from the very beginning, we went beyond the classical understanding of fairness in computer science, which is usually limited to non-discrimination (see Section 2.3.3).

In the next phase, we decided who was going to participate in the Assurance Case development process (Phase 2). As it relies on frequent intensive discussions, the number of participants should not be too high and the participants should be able to contribute to the Assurance Case development between meetings. We decided on two representatives from winnovation consulting gmbh as representatives of the stakeholders affected directly, two representatives from Rania Wazir e.U. as technology experts, two representatives from rotable as business experts, and me as the facilitator. A Miro board was prepared for use as the platform on which the Assurance Case was to be built (Phase 3).

After two joint meetings for getting used to the Assurance Case framework, the participants from winnovation consulting gmbh built a first draft

Assurance Case, for which each fairness understanding was represented by a sub-claim (Phase 4). In parallel, all participants met a few times to foster a common understanding of the use case and to define important technical terms that might have led to misunderstandings otherwise. At the same time, we reflected on the first draft of the Assurance Case to further empower all participants to contribute asynchronously (Phase 5).

After that, the participants from winnovation consulting gmbh and I met every three weeks to discuss the changes each of us had made asynchronously and to improve the Assurance Case in general. For example, we removed redundancies, merged branches with similar argumentation, and made formulations consistent (Phase 6). In the later stages, the technical experts from Rania Wazir e.U. joined the regular meetings as the focus shifted more and more to considerations of how evidences could be provided. Once we had a first final version, we handed it over to rotable for a thorough inspection. There were two follow-up meetings to adjust the Assurance Case according to their feedback.

Once all participants considered the Assurance Case to be finished, we checked the required evidences (Phase 7). It turned out that half of the evidences could not be provided yet, as the software still lacked the necessary functionality (e.g., to give the students the option to give feedback on their plan). Around a quarter of the evidences were not originally planned to be provided by the system (e.g., to provide statistical information to the students). The evidences were color-coded accordingly to support the remaining software development process. At this point, we published our results for external discussion (Hauer, Müller-Kress, et al., 2023) and my participation in the project ended. At this time, the Assurance Case consisted of ∼20 argumentation steps, ∼70 sub-claims, and ∼80 required evidences, ∼30 of which were unique (see Figure 4.7). At least until all the evidences can be provided, rotable plans to revise the Assurance Case on a three-month basis as part of their agile development process (Phase 8).

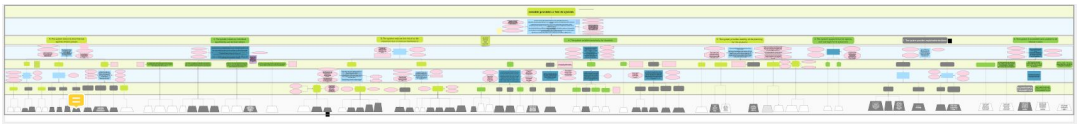


Figure 4.7: The Assurance Case we developed for the rotable clinical rotation scheduling software at the end of Phase 6. Due to confidentiality reasons, no detailed information can be disclosed. Thus, this figure only serves to get an idea of what a complete Assurance Case may look like. All gray elements refer to evidences and functionality that had not been provided yet at the stage of the software development process at the time.

The proposed approach is not a deterministic process and will thus not lead to a unique result. It can also not solve the main and principal problem of how to define ethical values in a quantified and widely accepted manner. Nevertheless, it describes a pragmatic approach to arriving at a well-documented argument about when and under which assumptions a system is deemed “ethical enough” to be used. Therefore, the framework

is suitable for supporting requirements engineering processes, even before the actual development starts (for more about this consideration, see Section 5.7.3). In this case, the proposed Assurance Case development process adheres to the precautionary principle (see Definition 18).

Definition 18 (Precautionary Principle)

The precautionary principle, as defined by the EU Commission[30] is one of the main principles on which EU policy is based.[31] Ricci and Sheng, 2013[32], explain that the precautionary principle is an approach to innovations focused on caution and prevention, applicable whenever policy has to deal with weakly understood causes of potentially catastrophic or irreversible events, or threats of harm to human life, property, and/or well-being. Based on this understanding, R. D. Taylor, 2020, argues for applying the precautionary principle at least to AI-based applications that fall under this delimitation.

An Assurance Case can also be developed during actual development (as we did in the project fAIr by design), for continuous deployment approaches, and after the software is deployed but continues to receive updates and upgrades on a regular basis. Additionally, an Assuracne Case can be easily communicated for an external review or audit, for example, in case of a lawsuit, or for a certification process[33] (see Section 4.1.1). As standards for certifying the fulfillment of ethical requirements on AI-based ADM systems are currently insufficient for accredited certification (see Definition 17, p. 87), companies have to rely on unaccredited auditing and certification processes if the fulfillment of certain ethical requirements is to be attested. Using an Assurance Case can help an auditor/certifier to search for gaps in an ethical assurance argumentation, keep track of the provided evidences, and challenge the assumptions made by the respective actors, at least until appropriate standardization documents are available. Last but not least, an Assurance Case can be used by a company to communicate the extent to which it complies with ethical requirements beyond what is required by law, as it enables the company to document and explain the effort made to decrease and manage ethical concerns in the best possible way.

[30] Communication from the Commission on the precautionary principle COM/2000/0001 final, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52000DC0001, last accessed on April 24, 2023.

[31] Consolidated version of the Treaty on the Functioning of the European Union - PART THREE: UNION POLICIES AND INTERNAL ACTIONS - TITLE XX: ENVIRONMENT - Article 191 (ex Article 174 TEC), https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX%3A12008E191%3AEN%3AHTML, last accessed on April 24, 2023.

[32] According to R. D. Taylor, 2020, who cites Ricci and Sheng, 2013.

[33] Brundage et al., 2020, p. 65, state that Assurance Cases are sometimes already required for certifying safety claims.

Assurance Cases could also be improved by applying some sort of *Common Task Framework* (CTF) methodology. To apply a CTF, a publicly available data set and a specific challenge to be solved are provided to a crowd of people interested in mastering this challenge. All solutions are made publicly available as well, sorted by some sort of scoring function that evaluates them (Wikle et al., 2017). The CTF has been applied for years on well-known data sets, such as the *MNIST handwritten digits database*[34] or those provided by *Kaggle*[35] to push the capabilities of data-driven technologies (Donoho, 2017, p. 752). While the framework is not suitable for use with Assurance Cases, the general idea could be adapted by providing a much discussed, legally and morally challenging task to be solved by a DDC (such as an AI-based system that suggests credit limits (see Example 1, p. 14), matches job candidates with open positions (see Example 2, p. 21), or assesses the risk of recidivism of criminal offenders (see Example 4, p. 38)). The task now is not to create a solution based on a data set provided but to develop a strong argumentation, represented by an Assurance Case, about what the most important claims to be assured are, how to formulate them precisely enough, and what evidences need to be provided. Multiple contestants could publish their variations and make improvements based on the solutions of others. The result may be a collection of great generic Assurance Cases for specific kinds of applications that can be used as strong templates for future Assurance Case development processes (see Phase 4) regarding similar applications. This is basically what Gauerhof et al., 2018, also argue for in the conclusion of their publication in which they introduce a generic Assurance Case that argues safety in autonomous driving.

RQ 5

How suitable is the Assurance Case framework from the field of safety engineering for use with extra-functional requirements such as fairness?

Answer: A finished Assurance Case is a documentation of why which evidences, including details of those evidences (e.g., such as test results), are considered sufficient to imply a claim. Therefore, it may be useful to communicate the effort made to manage legal demands and reduce ethical concerns as much as possible.

Applied in the context of an actual software product development process, the Assurance Case framework proved to be quite valuable for formulating relevant requirements to ensure fairness based on the various understandings of fairness of all stakeholders.

An Assurance Case may also be useful for auditing and certification practices. To assess the usefulness of the Assurance Case framework for such purposes, further research is necessary.

[34] http://yann.lecun.com/exdb/mnist/, last accessed on March 30, 2023.
[35] https://www.kaggle.com/datasets, last accessed on March 06, 2023.

The use of Assurance Cases to argue the fulfillment of ethical values is still in its infancy. Nevertheless, the potential benefits are already being recognized (see, e.g., Porter et al., 2022, DIN/DKE, 2023, Chapter 4). The extent to which the framework can live up to expectations will be demonstrated in the form of practical trials over the next few years.

Tests can be performed both in the context of audits according to ISO and in the context of audits according to Sandvig et al., 2014. Furthermore, test results can also serve as evidence for an Assurance Case. Which tests exist and under which conditions they are applicable will be the topic of the next chapter.

Chapter 5

Testing

With different auditing procedures, forums can get access to information or systems to ask their questions. With Assurance Cases (AC), there is a systematic approach for translating any question or vague requirement a forum might have into concrete requirements. However, validating that ADM systems actually meet specific requirements is a challenging task on its own, no matter how well-conceived and well-formulated they may be. Especially with non-functional (and extra-functional) requirements, some room for interpretation in the assessment of whether a software system complies with them can hardly be avoided. As long as the requirements relate to a non-critical product quality, this is not a problem. Non-compliance 'only' affects the marketability and thus the financial value of a product. However, if the requirements are intended to implement ethical or even legal standards – in the case of critical applications, this can also address product quality, such as in the case of safety-critical applications – this room for interpretation may represent an ethical risk for the people affected or a legal risk for a company. The task of computationally validating the fulfillment of requirements is addressed by the topic of software *testing*.

Thorough testing of software products has long been a cornerstone of the software development industry. Especially those software products that potentially have an immense impact on the lives of the members of a society need to be tested extensively. However, the use of AI technologies poses a major challenge to current testing practices, since rules according to which the system works cannot simply be found in the code and there is not always an expected behavior[1] that can be tested for. Thus, which tests are appropriate for an ADM system with a data-driven model (DDM, see Definition 4, p. 16) cannot be answered easily. Pre- and post-processing steps that incorporate a DDM into a data-driven component (DDC) can be addressed with traditional software testing approaches. Testing the implementation of a DDC in a specific context, including, for example, manual procedures that are part of that implementation, poses further challenges.

This chapter elaborates on the technical term "testing". It explains which different levels of abstraction are encompassed by the term (Section 5.1)

[1]The emergence of AI-based systems has led to increasing anthropomorphization of software systems in everyday language, which is why experts are increasingly criticizing the attribution of properties to software systems that are restricted to living beings. However, it is also common in technical jargon to speak of expected and actual behavior in the context of software (e.g., in IEEE Std. 829 (2008) - Standard for Software and System Test Documentation, Section 11.2.4), which is why this terminology is retained here.

and which approaches per level of abstraction are particularly interesting in the context of testing AI-based applications (Sections 5.2 - 5.7). There is hardly any right or wrong in defining different levels of abstraction, as many test-related terms are interconnected. A strict ontology is only necessary to a limited extent, but it is important that the terms be understood in the same way when communicating with each other. For this reason, this chapter also deals with varying interpretations of terms.

The collection of test-related terms for each level of abstraction is not based on a systematic literature research, as the literature on the subject is too extensive and inconsistent in terms of definitions and ontologies. Instead, I collected and structured test-related terms[2] based on some of the most cited books on software engineering and testing,[3] added relevant terms to the list that I came across during my detailed research, and made them available online[4] with a request for additions and critical feedback. Based on the feedback, I extended the list with specific suggestions and the contents of the ISTQB[5] glossary. This is the basis for the most relevant tests for ADM systems containing a DDM I worked with for this chapter.

Special focus lies on testing methods that aim to assess the fulfillment of fairness requirements and how these can be implemented in the context of the various audit concepts according to Sandvig et al., 2014 (see Section 5.4.2).

5.1 Test Terminology

The very general terms '*test*' or '*testing*' and extensions like '*test method*' or '*test concept*' are used for many different kinds of methods, concepts, and general ideas. At the same time, there are many things that could be called a test but which are not designated as 'test'. In the following, the term test as well as other related terms will be defined as precisely as possible to avoid ambiguities.

The definition of testing in this thesis is based on ISO 29119-1[6], which states that testing is a '*set of activities conducted to facilitate discovery and/or evaluation of properties of one or more test items*', while a test item is '*the result of an activity, such as management, development, maintainance, test itself, or other supporting processes*'.

In other words, test(ing) is an umbrella term that can be examined at different levels of abstraction. In the context of this work, a distinction is made between the following six levels:

[2] Including, for example, test levels and test development processes that are not tests themselves, and also terms that do not contain the word 'test', like code coverage.

[3] Most importantly: Ammann and Offutt, 2017; Jorgensen, 2014; Myers et al., 2004; Van Vliet et al., 2008.

[4] See https://www.linkedin.com/posts/marc-hauer-288a15b2_testconcepts-activity-6916642155130220545-8CU8?utm_source=share&utm_medium=member_desktop and https://twitter.com/hauer_p/status/1510874827418226690, both last accessed on June 26, 2023.

[5] International Software Testing Qualifications Board, https://glossary.istqb.org/en/search/, last accessed on February 17, 2023.

[6] Software and systems engineering - Software testing - Part 1: Concepts and definitions.

- **Test Level**. The test level addresses different test stages in the software development process (Ammann & Offutt, 2017, pp. 22-23). Each stage has completely different objectives. Section 5.2 does not make any novel contribution to this abstraction level but helps to better understand the test concepts and methods presented in the following sections.

- **Test Concepts**. There are some principal types of tests that are particularly well suited for testing certain properties or for testing in certain situations. Section 5.3 explains test concepts for traditional software that can be transferred for testing DDMs and test concepts specifically tailored for testing DDMs.

- **Test Methods**.[7] There is a large number of test methods described in the technical literature, most of which can also be implemented in different variations, depending on the specific test object. Most of these tests address traditional requirements of a software system. These aspects are also important for testing AI-based systems, but there are several significant peculiarities that necessitate extensions to traditional test activities. For one thing, AI-based systems are often so-called black-box systems (see Section 4.1), which means that many test activities are not feasible. In addition, AI-based applications often do not have an *expected behavior* to test for, which makes statistical test methods particularly relevant. These peculiarities open up the field of so-called *Black-Box Testing*, which will receive increased attention in Sections 5.4.2 and 5.6.

  It should be noted that the focus of Section 5.4 is on test methods that only become relevant through a DDM. This does not mean that traditional software tests lose importance. Traditional unit tests, for example, are still necessary to ensure that the code actually does what the programmer intended.

- **Test Case Generation**. For testing activities based on inputs, it is rarely possible to test the entire input space (and even if possible, it is a waste of resources). This is because the number of possible inputs and the combinations of inputs are usually too large to consider every possible case (this problem is also called *combinatorial explosion*; see Marijan et al., 2019). Therefore, there are various optimization methods for facilitating the generation or selection of test cases, as shown in Section 5.5, some of which place different weights on specific objectives. In practice, test case generation must be considered in the context of the test method to be used. As understood by the ISTQB, test case generation is even an explicit part of a test method. As the goal of generating test cases is fundamentally different from that of applying test methods, test case generation is discussed separately in this thesis.

[7] In many documents, this is simply referred to as *tests* (e.g., in ISO 29119-1 Software and systems engineering - Software testing, Pat 1: Concepts and definitions, Section 4.47). To prevent ambiguities, the term *test methods* was chosen for this thesis, as it is not uniformly used in the literature. The American Society for Testing and Materials (ASTM) defines a test method as '*a definitive procedure that produces a test result*' (ASTM, 2022), which corresponds to the understanding in this thesis.

- **Test Schemes**. Various test schemes will be explained in Section 5.6 that are often considered test concepts or methods on their own, but do, in fact, involve one or more other test methods to inspect, for example, whether tests cover a certain percentage of code or how well a system under test performs compared to an older version of itself, an alternative version of itself, or another similar system. To the best of my knowledge, no term has been coined for this level of abstraction. Therefore, I choose the term *test schemes* as it seems to fit the general idea and is not being used in a different, widely accepted meaning.

- **Test Development Processes**. Determining under which circumstances which tests should be implemented at which point in the development process is the subject of much debate. For traditional software tests and software development processes, numerous books exist that deal extensively with the subject under different terms (e.g., Jorgensen, 2014, pp. 207-219, refers to the discussion under the term **Life Cycle-based Testing**), which is why this topic is deliberately not addressed holistically in this thesis. Since the question of what exactly should be tested, as well as the selection of suitable tests and the determination of their benefit, is significantly more complicated for AI-based systems, modern test development processes are being continuously refined and new test development processes are regularly emerging in the field. How the development, execution, and fulfillment of test requirements can be implemented within some of these novel processes will be discussed in Section 5.7.

A test is successful in the literal sense if it detects an existing error for which the test was written. In everyday language, however, it is also said that a test is successful if it runs without detecting an error. To avoid this ambiguity, a test that runs without finding an error is also said to pass and a test that finds an error is said to fail.

Automated tests must be written by hand but can be run completely automatically on a regular basis, for example, with the help of a so-called test server. Examination of the test results must be done manually. If versioning is also used (e.g., with Git or Apache Subversion (SVN)), the code can be compared with an earlier version if an error is discovered in order to quickly find the cause. Building an automated testing process often requires the use of external tools/frameworks, potentially introducing third-party dependencies and new sources of errors. Test automation is therefore only worthwhile beyond a certain code complexity and size.

Tests for DDMs differ from tests for traditional software in some essential points (Jöckel et al., 2021). In traditional software, there is a clearly specified expected behavior for every possible input. This expected behavior is developed during the requirements engineering process and recorded in a so-called *specification* document.[8] It *'specifies, in a complete, precise,*

[8]There is a lot of discussion online about the difference between requirements and specifications. There seems to be a vague consensus that requirements focus on the goals to be implemented, whereas specifications address technical details. The technical literature also shows this tendency, without being in complete agreement on this point (e.g., Myers et al., 2004, p. 125).

verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and, often, the procedures for determining whether these provisions have been satisfied' (IEEE Std 610.12-1990, 1990). A *test object* is any executable component to be tested against the specifications. A test objective is defined as '*an identified set of software features to be measured under specific conditions by comparing the actual behavior with the required behavior described in the documentation or specification of the test object*' (IEEE Std 829-2008, 2008).

With DDMs, the expected behavior is only specified for a small subset of possible inputs (ground truth; see Definition 3, p. 13). Therefore, a specification can only be incomplete. For new inputs without previously specified correct output, the behavior of the DDM can only be assessed to a limited extent. Automated evaluation is not possible without a ground truth, but a human could look at the output and determine whether it is appropriate. Depending on the specific ADM system (e.g., a prediction model such as credit score assignment; see Example 1, p. 14), the input can also be used for an actual application to compare the real output with the model output. Another approach may be to use the ADM system output for a practical application (e.g., to assign students a place at university; see Example 3, p. 33) to check whether the impact is desired. Furthermore, regardless of whether an output is correct or not, it is possible to check whether the outputs for many inputs correspond to a desired distribution or not.

A DDM can be built by dividing the ground truth into a *training data set*, (optionally) a *validation data set*, and a *test data set* (see Definition 19).

Definition 19 (Training, Validation and Test Data Set)

A very common approach to building DDMs is to divide the ground truth data into three data sets: training data set, validation data, set and test data set.[9] The training data set is used to train a DDM. The validation data set is used to '*assess the performance of one or more candidate [...] models*' (analysis). The test data set is used for quality assurance of the selected model. Only if testing activities provide sufficient evidence of quality is a product deployed and put into operation.

The training data set has already been introduced (see Section 2.1). The training process continues until the DDM performs well on the training data set. If the given combination of training data, (untrained) model structure, and training process does not result in a DDM that performs well enough (e.g., after n training iterations), the model structure and/or training process is adjusted. In the next step, the validation data set is used to analyze how the trained model performs on previously unseen data. If performance is bad, it may be necessary to tune the model structure or

[9] See ISO/IEC DIS 22989 - Information technology - Artificial intelligence - Artificial intelligence concepts and terminology.

the training process (e.g., by adjusting the hyperparameters) and restart construction (Ripley, 1996, p. 354). The analysis is often also called *'validation'*. This step is distinct from the definition of validation in the context of system and product development, as validation in the ML context is merely an intermediate step in the training process and not an immediate validation of the final system.

Construction and analysis are repeated (sometimes with multiple models in parallel) until the model performs well enough on the validation data set. In this case, the model structure itself is considered to be fitting to make predictions on the available data. Then the test data set is used to thoroughly test how well the final model performs (Ripley, 1996, p. 354). Each data point of this data set can be seen as a test case, providing the model input and the expected outcome. Due to the analysis step, the test results should not be all too bad, so if the results are not satisfactory, the model structure itself is kept and only training is resumed or adjusted (fine-tuning). Note that adjusting an AI model based on tests or test cases is a delicate process, as it may specialize on just these cases while results for other inputs remain unsatisfactory.

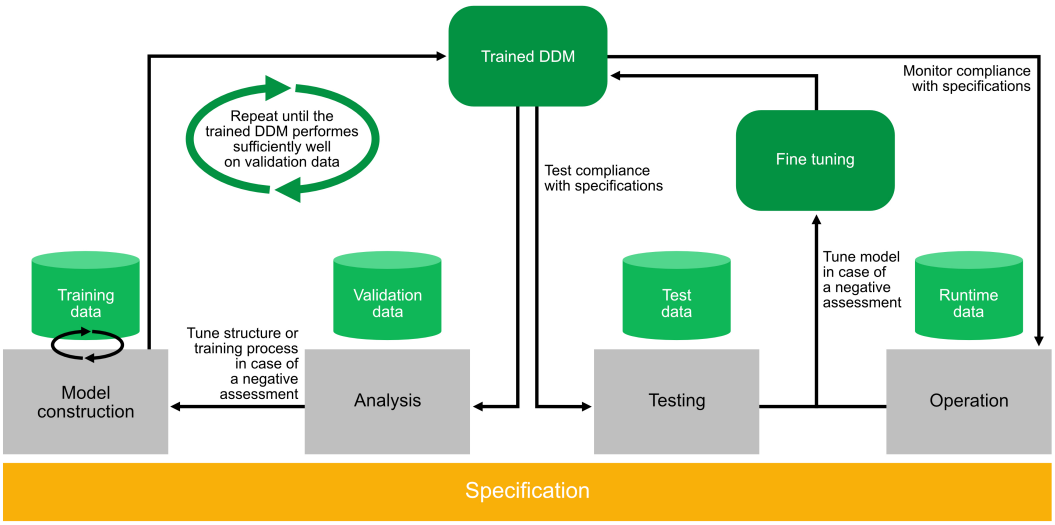


Figure 5.1: Visualization of the use of training, validation, and test data before the deployment of a DDC and runtime data after deployment. The workflow assumes that only minor issues may arise during testing and operation as a result of a proper model construction and analysis process.

Analysis and testing activities are sometimes considered to be the same (M. Kuhn, Johnson, et al., 2013, p. 67). Additionally, there are multiple cross-validation techniques that replace the need for a separate analysis step (M. Kuhn, Johnson, et al., 2013, pp. 69-78); therefore, splitting a part of the ground truth into a validation data set is often deemed obsolete. For the sake of completeness, Figure 5.1[10] also shows a fourth

[10]Note that this lifecycle model (as well as many other similar models) basically shows the same process as the long chain of responsibility (see Figure 3.2). Different lifecycle models are designed to communicate different ideas. While the focus of the long chain of responsibilities is on illustrating the challenges of assigning responsibility, Figure 5.1 shows how different data sets are used for quality assurance in different steps of the development process of a DDM.

step, operation (see also Sections 3.1.7 and 3.1.8). In this step, runtime data that is generated during the application of an ADM system is monitored without using any of the three data sets that result from splitting the ground truth. This is also called *Field Testing* or *Post-Market Monitoring* (see *Field Testing* in Section 5.5).

Testing the training process itself does not provide any information about the behavior of a DDM. Since the functionality of a DDM is derived from and evaluated on data, it makes sense to perform testing activities on them (see *Data Set Testing* in Section 5.3). Another test object is the DDM itself. Unlike in traditional software testing, requirements on functional correctness need to be given a probabilistic sense, as the input-output relationship cannot be fully specified and uncertainty in the DDM outcomes (and DDC outcomes, if there are any pre- or post-processing steps, see Definition 4, p. 16) cannot be fully eliminated. Aspects to test for in this way are, for example, prediction quality and fairness (see Section 2.3.3). Further test objectives might be performance under specific conditions (e.g., environmental influences or replacement of sensory hardware) or explainability (see Definition 15, p. 52).

RQ 6

How does traditional test terminology differ from test terminology in the context of DDCs?

Answer: For traditional software systems, the test objects are individual components of the system or the system as a whole. Each component contributes to a specific function or set of functions of its associated system, which allows a clear definition of the requirements to test for. The definition of test objectives can be derived directly from the requirements.

For DDCs, the test object can be the data, the DDM, the process in which it is used (e.g., when there is an obligation to use the DDC only for recommendations), and, depending on how it is intended to be used, many other things. For most of these test objects, it is rather challenging to define clear test objectives, as test requirements range from quality aspects (e.g., prediction quality, fairness, and robustness) to insight and control (e.g., documentation and explainability) to adherence to manual processes and protocols (e.g., a human in the loop). The test objectives can rarely be formulated as strictly defined expected behaviors, but as required confidences in terms of probabilities.

5.2 Test Levels

The explanations of the test levels are largely based on Myers et al., 2004, Chapter 6. They are largely consistent with what is stated in the broader software testing-related standard literature. *Function Testing* is excluded, as there are too many discrepancies in the literature explaining this term

and it remains unclear whether it is the same as *Functional Testing*.[11] *Installation Testing* is not of interest for this thesis as it deals with testing installation processes, which are not within the scope of this work.

Acceptance Testing refers to testing activities performed by the end user to show that the system does not provide the agreed range of functions or is defective. If the end user fails, the acceptance tests pass.

System Testing tries to determine whether a system does not meet its objectives. For this purpose, measurable objectives must be defined. If such objectives are not available, they need to be extracted from the planned application context or the requirement documents.

As software testing is traditionally an activity governed by software system developers and providers, the common test level terminology follows a generic development lifecycle that does not consider testing activities beyond the deployment of a software product, i.e., after *Acceptance Testing* is finished. Testing activities after deployment, for example, by customers, NGOs, scientists, etc., can be considered *System Testing* or *Acceptance Testing*, as test concepts and methods that are usually considered at any of these two levels can be applied.

Integration Testing '*focuses on the interaction of components and systems*'[12]. Due to the combinatorial explosion of possible inputs, not every combination can be tested at the system test level. Therefore, integration tests have a spot-checking character, with the interaction of critical components to be tested in particular detail. At least when component(s) and system(s) are developed by the same organization, *Integration Testing* is an integral part of *Module Testing*. Therefore, *Integration Testing* is sometimes not considered a separate test level.

Module Testing (also called **Unit Testing**[13], **Component Testing** (Utting & Legeard, 2010) or **Developer Testing** (Myers et al., 2004)) is about finding errors in a functional code segment by comparing the expected behavior with the actual behavior in terms of correct inputs, edge cases, wrong inputs, and non-processable inputs. Those testing activities can also be performed when a module is integrated into a more complex system. In this case, *Module Testing* and *Integration Testing* overlap. Since only individual modules are tested, module tests are particularly suitable

[11] See, for example, the definition of *Function Testing* according to Myers et al., 2004 vs. the definition of *Functional Testing* according to Jorgensen, 2014. It seems to be clear that *Function Testing* and *Functional Testing* are different levels/concepts, though neither refers to the other level/concept. Other literature does not seem to provide any help in this regard.

[12] https://glossary.istqb.org/en_US/term/integration-testing-3-2, last accessed on May 15, 2023.

[13] Sometimes *Module Testing* and *Unit Testing* are considered to be the same (e.g., Myers et al., 2004, Chapter 5); sometimes the distinction is made that *Unit Testing* focuses on the smallest units of code while *Module Testing* focuses on testing an isolated 'module' that implements a specific functionality and the interaction of the units the module contains (e.g., Ammann and Offutt, 2017).

for monitoring potential errors when changing the code of a running system. For the same reason, they also provide a good basis for building a reusable code base.

There is already a great deal of literature on testing traditional software at all levels (e.g., Jorgensen, 2014; Myers et al., 2004; Utting and Legeard, 2010). Therefore, the following sections will focus only on test concepts and test methods that are suitable for testing the special properties and requirements of ADMs with DDCs. On the one hand, this refers to traditional test concepts whose basic idea can be transferred to testing DDCs and, on the other hand, to test concepts that were explicitly developed for testing DDCs. At the same time, it excludes module testing, which does not differ from traditional software systems.

5.3 Test Concepts

Test concepts are principal types of tests that are particularly well suited for testing certain properties or for testing in certain situations.[14] Each test concept encompasses various possible test methods (see Section 5.4) and/or test case generation methods (see Section 5.5). As there is a great number of test concepts, this section only lists those that are especially interesting to explore to test a DDC. Nevertheless, most of these concepts are also used for non-data-driven software systems and are discussed in detail in the respective technical literature. Thus, it is possible to benefit from the experience gained there when using these concepts to test DDCs.

Data Set Testing is necessary to ensure that data collection and pre-processing (see phases B and C of the long chain of responsibilities in Sections 3.1.2 and 3.1.3) have been performed well and to identify the need for (further) data collection and/or pre-processing. Some methods focus on statistical evaluations, either by computing statistical values or by using data visualization techniques (Vartak et al., 2015). Others check whether values are within expected ranges. Both approaches try to identify outliers, which trigger a manual validation check. Such an evaluation can be very subjective and should only be carried out with great caution. Even if values seem to be unrealistic, they might result from events whose inclusion in the data may be important (e.g., the COVID-19 crisis had a huge impact on health-related data sets; see, e.g., Giuntella et al., 2021) or from gradual drifts in trends over a longer period of time (Polyzotis et al., 2017). However, if many values are found outside an expected range, it is worth investigating the reasons. Mistakes may have been made in data collection or processing. These challanges can also be partially automatically adressed by manually setting a value range for which the respective feature is automatically checked. If too many values are found outside the range, the test runs red (Polyzotis et al., 2017). If data stems from multiple sources, another approach is to validate that

[14] ISTQB refers to the term *'Test Type'* with a similar understanding: *'A group of test activities based on specific test objectives aimed at specific characteristics of a component or system'*, https://glossary.istqb.org/en_US/term/test-type-4-2, last accessed on May 15, 2023.

all sources are valid; for example, if 9 of 10 sources show a similar trend and one does not, inspecting whether the deviation has a viable reason or not might be advisable. It is also important to check whether specific values are ascribed a special meaning (e.g., -9999 to represent missing data) and to ensure that training, validation, and test data are encoded in the same way (e.g., that categorical data is represented by the same numerical values; see Hynes et al., 2017). Some of the listed approaches are independent of a DDC and are also discussed in the database management system literature (see, e.g., Ramakrishnan et al., 2003). *Data Set Testing* also involves simply reflecting upon the available features and whether they are relevant for a given task. Last but not least, the training data can be tested for bias (see, e.g., Hynes et al., 2017; Polyzotis et al., 2017).

Penetration Testing refers to all testing activities aimed at finding the vulnerabilities of a software system. There are many generic approaches (e.g., crashing the application and dumping the system to search for unencrypted passwords, or brute-force approaches), but *Penetration Testing* also comes down to good intuition about a software system and creativity. **Red teaming** describes the process of having both internal programmers who know the code and external security experts or hackers who are familiar with sophisticated software for *Penetration Testing* attempt to find and exploit vulnerabilities (Potter & McGraw, 2004). Due to the increasingly large role played by AI-based systems in everyday life, more and more attempts to attack these systems are to be expected. Therefore, it is becoming increasingly important to take an attacker's perspective in order to detect vulnerabilities and eliminate them. If finding discriminatory system behavior is understood as an attack, *Penetration Testing* methods can also be used to detect and mitigate such behavior. This will be explained in detail using the example of a *Penetration Testing* method called *Adversarial Testing* in Section 5.4.2.

Performance Testing is concerned with testing several aspects of performance. Mostly, it is about the speed or accuracy with which tasks can be executed (Myers et al., 2004, p. 137). This can often be automated and is especially useful for *Regression Testing*, which will be explained in Section 5.6. However, how exactly performance is best evaluated is potentially subject to constant change, which is why automation is only partly useful. In addition, aggregated information across many subsystems is not useful, as the focus is on detecting particularly underperforming (e.g., slow) subsystems. Performance can also refer to how 'usable' a system is, for example, in terms of the user interface. Here, only manual checking is possible. **Benchmark Testing** is a special form of *Performance Testing* that involves the (automatable) comparison of performance metrics. Measuring quality or fairness metrics and comparing them to predefined benchmarks (thresholds) is a standard procedure for testing DDCs (more on this in Section 5.4.2).

Testing with different operating systems, hardware devices, etc., is called **Configuration Testing**. The purpose of this test concept is to check

whether a hardware component is insufficient for the purpose of a software system. Many ADM systems, regardless of whether they contain a DDC or not, obtain the data on which decisions are based in an application context from various sensors (e.g., cameras, microphones, bumpers, etc.). *Configuration Testing* can be used to determine the minimum requirements that hardware must meet to be used for such a system.

Just as it is important to test the software and the hardware, it is also important to test the documentation, which includes *'user manuals, online help, design features and specifications, source code comments, test plans, test reports, and anything else written that explains how something should work or be used'* (Mamone, 2000, p. 26). In the context of this thesis, two aspects of **Documentation Testing** are especially relevant. First, each documented example is expected to be an implemented test case (Myers et al., 2004, p. 142). This also includes negative examples, such as specific discrimination scenarios. Second, all test results should be documented, for example, to allow carrying out an evaluation in the context of *Regression Testing* (see Section 5.6), or to enable an assessment in the context of an audit (see Chapter 4).

Procedure Tests refer to activities for testing manual processes that are part of using the software system to be tested (Myers et al., 2004, p. 142). As probabilistic decision-making processes are always at risk of erring, regardless of how small the process may be, operators often decide to use such systems only to get recommendations; the actual decisions are made by humans. In some cases, there is even a legal obligation that there be a *'human in the loop'* (e.g., according to Art. 22 GDPR). Procedure tests aim to find out whether manual processes are suitable for their purpose, whether they work, and whether they have any undesirable side effects.

Since most ML methods are black-box systems, it is often difficult to understand how exactly an output is generated, i.e., which factors have a particularly large influence on the result in individual cases or in general. Some **explainability methods** attempt to provide insights here, while others attempt to make the decision structure of a model comprehensible or provide reference samples for a given label (see Definition 15, p. 52). Explanation approaches are often considered to be a specific kind of testing, but have a fundamentally different goal, which is why the general idea is seen as a separate concept here.

(Technical) **Reviews** refer to any manual process by one or more individual(s) who search for defects or possible improvements in the code by hand. Based on this definition, they can be understood as a test concept according to Jorgensen, 2014, p. 417, ISTQB[15] glossary, and ISO 29119-1, or even as a test method according to Myers et al., 2004, p. 22. At the same time, what all of them describe also matches the definition of code

[15] International Software Testing Qualifications Board: https://glossary.istqb.org/en_US/term/technical-review-1-3, last accessed on February 17, 2023. Reviews are static tests carried out by experts on work products. In this sense, *Documentation Testing* and *Reviews* are the same thing according to the ISTQB.

audits as described by Sandvig et al., 2014, which were already discussed in Section 4.1.2.

5.4 Test Methods

Test methods describe concrete test activities that provide a measurable result that can be compared to an expected behavior. As mentioned earlier, such an expected behavior is often not available in the context of AI-based applications, especially when testing a black box for fairness (Krafft et al., 2023). In the context of testing, the collection of expected behaviors is also called a (test) *oracle*. Depending on whether an oracle is available or not, different test methods may be applicable; therefore, Section 5.4.1 elaborates on the so-called *Oracle Problem*. Section 5.4.2 focuses on test methods that are either particularly suitable or adaptable for testing black-box ADM systems for bias (which is also one definition of *Black-Box Testing*; see Section 5.6). Section 5.4.3 explains how these methods are related to the black-box audit forms described in Section 4.1.2.

5.4.1 The Oracle Problem

An oracle is any method that allows validating whether the system output can be considered correct (Howden, 1978). A ground truth, for example, is a frequently used special form of a test oracle, as the correct output for a limited set of inputs is known. There are also probabilistic oracles for which a certain error rate is tolerated (Barr et al., 2015), for example, an algorithm or process that provides a probability that the test result is correct. As already mentioned, the information about what can be considered as 'correct' output for most of the possible inputs is often missing for DDCs. This is the so-called '*oracle problem*'.

To counteract this problem, there are test methods for assessing the bias of DDCs that do not require a traditional oracle at all. Some test for a certain ratio or distribution in the outputs, regardless of whether the respective outputs are correct or not. Others are based on an oracle substitute (Krafft et al., 2023). Depending on the definition, both could also be considered oracles – the literature is not clear regarding this classification (Barr et al., 2015). To avoid confusion, such determinations (whether test result distributions are accepted as correct or not, and substitutes) are explicitly not considered oracles in this thesis.

5.4.2 Test Methods Suitable for Assessing the Bias of DDCs

Various test methods have been specifically developed to assess the bias of DDCs, and some traditional test methods can be applied to this task. These methods aim to evaluate the performance of systems regarding a protected attribute in the data. While none of these methods is suitable for guaranteeing discrimination-free decisions if the corresponding tests pass, they are suitable for showing discriminatory decisions if the corresponding tests fail. Note that most of the described test methods do not necessarily require an oracle. If the corresponding tests fail, however, it is

impossible to determine without an oracle whether the disadvantaged individual or group is getting excessively poor results or whether the advantaged individual or group is getting excessively good results. Thus, these test methods serve to detect bias but do little to support investigation of the problem. Also, note that each of the described methods relies on the availability of information about protected attributes regarding which fairness is to be tested. However, in many contexts, such information may not be available for testing purposes due to legal restrictions (e.g., due to the GDPR). Thus, fairness testing may need to be based on artificial data, which might not sufficiently represent the real data.

Testing Based on Quality and Fairness Measures. The general idea of quality and fairness measures has been thoroughly introduced in Section 2.3.3. There are various ways to transform the plain computation of such measures into test methods. All of them are based on comparison with a threshold ϵ, which is interpreted as the expected behavior to which the tests result are compared (Haeri Amir et al., 2023).

To test prediction quality for example, that of a binary classifier – at least any quality measure Q based on the confusion matrix can be computed and compared to a previously specified ϵ:

$$Q \leq \epsilon \tag{5.1}$$

In the case of group fairness measures (see Section 2.3.3.1) that are based on the comparison of the quality measure values of two sensitive groups A and $\bar{A}$, there are at least two ways to compute the level of fulfillment. The first is to compute the absolute difference:

$$-- \mid (Q_A - Q_{\bar{A}}) \mid \tag{5.2}$$

The closer the difference is to zero, the fairer the system is considered to be. Similarly, the ratio between the two measures can be used as an indicator of the extent to which the parity requirement is met:

$$\frac{Q_A}{Q_{\bar{A}}} \tag{5.3}$$

In this case, the system is considered to be fairer the closer the ratio is to one. Note that the ratio is only helpful for comparing high values, as expected, for example, from correct classification rates. False classification rates might differ in magnitudes but will still be very small for both sensitive groups.

Given the difference or the ratio of the measures in the two groups, a threshold ϵ can be added that specifies to what degree the quality measure values are allowed to differ, i.e.:

$$\mid (Q_A - Q_{\bar{A}}) \mid \leq \epsilon \tag{5.4}$$

$$\mid \frac{Q_A}{Q_{\bar{A}}} - 1 \mid \leq \epsilon \tag{5.5}$$

As common quality measures are computed on confusion matrix values, which again require an oracle (see Table 2.2, p. 36), testing based on fairness measures that compare quality measure values requires an oracle as well. However, for group fairness measures not based on a comparison of quality measures, such as *Statistical Parity* (see Section 2.3.3.1), a threshold can also be defined. In this case, the measure is referred to as a **Relaxed Statistical Parity** version (Barocas et al., 2017, p. 55):

$$\frac{\Pr(\hat{Y} = 1 \mid A = \alpha)}{\Pr(\hat{Y} = 1 \mid A = \beta)} \geq 1 - \epsilon \tag{5.6}$$

Individual fairness (see Section 2.3.3.2) in its raw form is only suitable for testing the fairness of single decisions, as the results for individuals do not provide information about the behavior of the system as a whole. However, *individual fairness* can be reformulated by calculating it pairwise for all data points in a larger data set to automatically check whether dissimilar inputs often receive similar outputs and vice versa. If the fairness of a classifier (see Definition 7, p. 23) is to be tested, a function is required that states whether two inputs are similar or how similar two inputs X_1 and X_2 are. The outputs $\hat{Y}_1$ and $\hat{Y}_2$ are either the same (class) or not. In the case of a scorer (see Definition 7, p. 23), a second function is required that states whether the outputs are similar or how similar they are. Common functions for computing the similarity of vectors and scalars are the *Minkowski distance* (see Equation 5.7) and the *cosine similarity* (see Equation 5.8) (Ontañón, 2020, pp. 5312-5313). In its basic form, testing based on *individual fairness* assumes that all input information is equally important. If certain information is more important, it can be weighted or potentiated in similarity functions.

$$Sim_{Mink} = \Big(\sum_{i=1}^{n} \big|X_1(i) - X_2(i)\big|^p\Big)^{\frac{1}{p}}, \text{ where } p \in \mathbb{N} \tag{5.7}$$

$$Sim_{cos} = \frac{\sum_{i=1}^{n} X_1(i) \cdot X_2(i)}{\sqrt{\sum_{i=1}^{n} (X_1(i))^2} \cdot \sqrt{\sum_{i=1}^{n} (X_2(i))^2}} \tag{5.8}$$

Counterfactual fairness (see Section 2.3.3.4) can also be used for bias testing, by checking for every data point in the test data set and their counterfactuals whether they are counterfactually fair. It may be unlikely that an ADM system can fully satisfy *counterfactual fairness*, so a threshold must be specified, such as the percentage of the test data set that must satisfy *counterfactual fairness*, for the test to pass. In the case of a scorer, a second threshold must be defined, stating whether the outputs for an original input and its counterfactual are sufficiently similar.

Differential Testing compares the system under test S with another system A having the same or very similar functionality (Evans & Savoia, 2007); for example, two systems suggesting open job positions. A could also be an older version of the system under test S'; in this particular

case, the method is called **Back-to-Back Testing** (Vouk, 1988). When *S* predicts people's future behavior, it can even be compared to the judgment of human experts. In each of these cases, the outputs of *A* (or the human experts) are an oracle substitute. The test method consists of taking a large set of inputs to both systems and comparing the respective outputs (McKeeman, 1998). Depending on the specific implementation, different variations of the procedure are possible (see Figure 5.2). There could be a purely statistical evaluation of the number of cases in which the two systems contradict each other. If there are too many discrepancies for a given type of input, the test fails (Petsios et al., 2017a). In this case, the test can be automated. However, discrepancies can also come from the fact that the systems do not have exactly the same functionality, but only very similar functionality. Thus, the test method can only be used to detect discrepancies in behavior, which then have to be evaluated by hand (Groce et al., 2007). This approach is particularly interesting when multiple (potentially also black-box) systems are used as *A* for comparison (Pei et al., 2017).

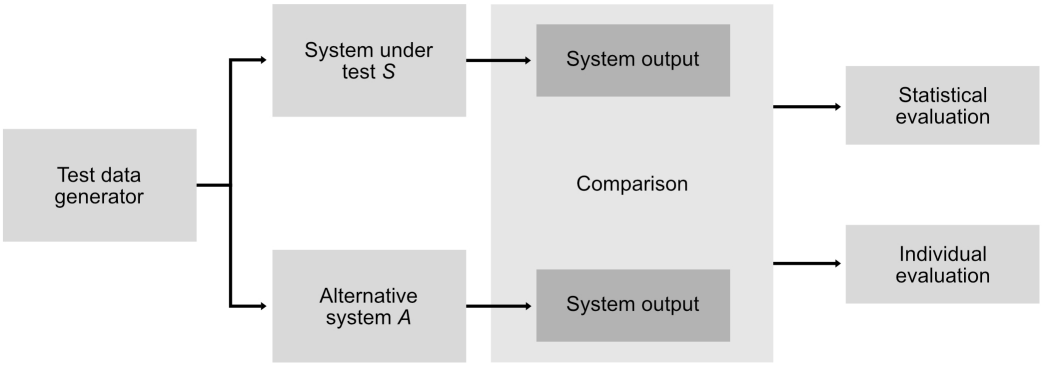


Figure 5.2: Differential Testing results can be evaluated statistically or individually (Krafft et al., 2023). The less similar S and A are, the less meaningful the statistical evaluation.

The advantage of *Differential Testing* is that the effort for manually labeling test data can be limited to those cases where the results contradict each other. Furthermore, these edge cases can then be added to the training data to improve the system under test *S*.

The disadvantage lies in the danger that all compared systems could have been created by developers who tend to make the same mistakes or have the same bias (Knight & Leveson, 1986). This may produce faulty outputs that cannot be detected based on this kind of test. Also, there needs to be an alternative system or an older version of the system under test to which the results can be compared.

Adversarial Testing is an approach based on the success rate of finding so-called '*adversarial examples*'. An adversarial example is a slightly modified example input that was created specifically with the purpose of resulting in incorrect model output (Goodfellow et al., 2014). Therefore, this is a kind of *Penetration Testing* (see Section 5.3). When this method is used with malicious intent, it is also called an '*adversarial attack*'. ANNs, in particular, are considered especially susceptible to this

type of attack (Szegedy et al., 2013). There are different methods for finding adversarial examples, depending on how much information a tester (attacker) has. In the case of black-box systems, the tester only has information about the system input and the system output. To the best of our[16] knowledge, this leaves only the method of *decision-based attacks* for finding adversarial examples (Brendel et al., 2018). In this case, the tester queries the system to be tested with synthetic inputs selected by a heuristic process to train a substitute model (also called a *surrogate model*; see Definition 15, p. 52) that mimics the decision boundaries. To find such decision boundaries, the system needs to be probed with input data that is iteratively adjusted depending on the respective system output. As the internal mechanisms of this substitute model can be observed, various algorithms can be applied to find adversarial examples that also work on the original model (Papernot et al., 2017). According to Brendel et al., 2018, this method works well for systems with low intra-class variability, but for other systems, there is a lack of experience. While the process of building a substitute model can be automated, the initial setup and its refinement need to be done by hand. This process can be quite difficult and time-consuming. It might take multiple iterations of trial and error to build a substitute model that shows sufficiently similar behavior as the system under test.

While with traditional *Adversarial Testing*, the goal is to identify inputs that lead to wrong outputs, this method can also be used in the context of bias detection by identifying input vectors for which a change to a protected parameter results in a change to the system output (see Figure 5.3). Based on the assumption that the protected property should not change the result, such a change is an indicator of an unwanted (unexpected) behavior, regardless of which of the two decisions was correct. Thus, there is no need for an oracle.

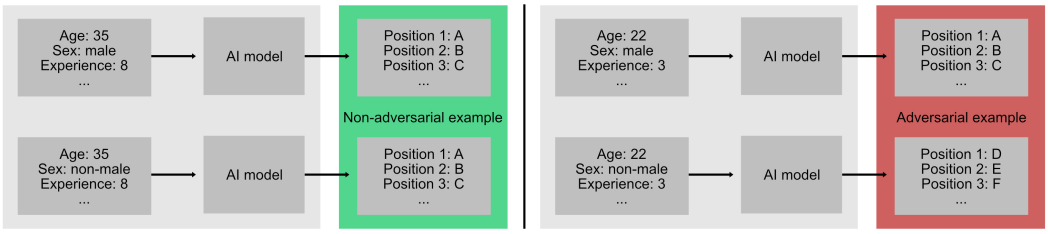


Figure 5.3: Comparison between a non-adversarial (left) and an adversarial (right) example pair. The persons in each pair differ from each other only in their gender. This should not affect the result in any case (left). However, the automatic testing process has found an example where the output differs between the persons (right).

There can be no absolute protection against wrong individual cases. Therefore, a measure is required that states under which circumstances different results as a consequence of a change to a protected parameter are acceptable. This is called an exit criterion. It may be the time until an adversarial example has been found or the number of attempts needed

[16] In Section 5.4, 'we' and 'our' refers to the authors of Krafft et al., 2023.

to find an adversarial example. In either case, the evaluation is based on individual results and therefore of a qualitative nature.

The basic concept of **Metamorphic Testing** is to develop assumptions about the input-output relationship of the system, so-called metamorphic relations, and then to adapt the test cases in order to test these assumptions (T. Chen et al., 1998; Segura et al., 2016). The metamorphic relations describe as formally as possible how changes in (individual) inputs should affect the outcome of the system under test (T. Y. Chen et al., 2003). For example, in the context of a system for selecting applicants for a job, if two applicants A and B differ only in terms of gender and years of experience, the person with more years of experience is expected to get a higher score. The metamorphic relation in this case states: If A and B are equal but A has more years of experience, it should get an equal or higher outcome, independent of gender.

More sophisticated metamorphic relations can be defined as well, e.g.: If A has more years of experience than B but is younger, A should generally have a higher outcome, independent of any other input information, including gender. It is also possible to define multiple (non-contradictory) metamorphic relations that all have to hold. In this way, the design flexibility of metamorphic relations allows investigating more complex questions around bias without increasing the complexity of the test.

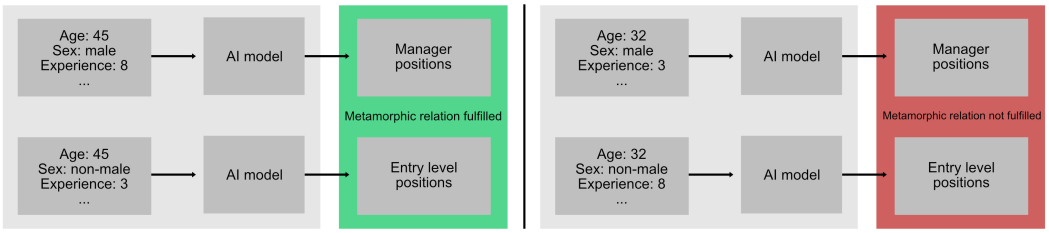


Figure 5.4: Example of the metamorphic relation *'If A has more years of experience than B and both have the same age, A should generally have a higher outcome, regardless of any other information'*. In this case, all candidates of the same age can be compared pairwise.

In practice, it is almost impossible to completely satisfy metamorphic relations addressing bias. A threshold is needed that states how much results may differ to still be considered as fulfilling the metamorphic relation. Additionally, a second threshold might be necessary that states the maximum number of test instances for which the metamorphic relations may not be fulfilled for the test to pass. This also means that a quantitative evaluation is performed. Once a metamorphic relation and a function to compare the results have been specified, the test method can be completely automated. While performing *Metamorphic Testing* approaches involves only little effort, appropriate metamorphic relations need to be specified in advance. This task is usually performed by a domain expert or experienced programmer and may introduce considerable effort (Kanewala & Bieman, 2013; Segura et al., 2016).

Dealing with Negative Test Results regarding Bias

Regarding the handling of negative test results, a distinction must be made between two scenarios: internal tests for quality assurance and external tests to challenge the system.

If internal tests are not passed, the focus is on improving the ADM system so that the tests are passed. If tests reveal functional errors, these can simply be fixed. There is no difference between traditional software systems and DDCs. Depending on the complexity of the code, this is not always easy, but generally feasible. In the case of non-functional errors, it depends on the non-functional property that is insufficiently fulfilled as to what exactly possible improvements could look like. For example, if the system reacts too slowly to inputs on a selected test machine (the non-functional requirement of *time behavior*, according to ISO 25010[17]), changes are necessary to make the system more performance-oriented, for which there are potentially numerous possibilities.

Dealing with failed fairness tests on DDCs is particularly difficult because it is hard to determine the causes. There may be a problem with the data. Training data that does not meet the desired ideal of fairness, whether or not it is accurate and representative, is likely to produce a corresponding DDC that meets the implications of the training data. Even if the training data appears to be unproblematic, proxy variables may be the cause of indirect discriminatory behavior. In addition, it is also possible that discriminatory behavior arises from the training process or a combination of factors.

If there are errors in the training data set (see *Data Set Testing* in Section 5.3), these can be corrected. In the worst case, the data collection process needs to be changed and/or repeated. However, if the data is correct and representative, but still disadvantageous for a protected group of people, it gets tricky. The DDC resulting from the data will most likely reflect the discriminatory information in its decision structure and thus fail the corresponding fairness tests, even though the quality of the information corresponds to the specifications. In order to specifically establish fairness without reducing the meaningfulness of a DDC, a large number of different methods are discussed in the literature. These can be divided into three categories: pre-processing methods, in-processing methods, and post-processing methods.

Pre-Processing Methods transform the data set in a way that unfair data is mapped onto a fair subspace while attempting to preserve correctness (see, e.g., Kamiran and Calders, 2012; Zemel et al., 2013; Feldman et al., 2015; F. Calmon et al., 2017; F. d. P. Calmon et al., 2018; Lahoti et al., 2019). *Reweighing*, for example, assigns weights to the data instances to adjust the probability that a data point is used for training. For this technique, all samples can be selected multiple times. Therefore, data points that provide more fairness are selected more frequently (Calders et al., 2009).

In-Processing Methods focus on the DDC or a combination of DDCs

[17] Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

(see, e.g., Kamiran et al., 2010; Kamishima et al., 2011; Bechavod and Ligett, 2018; Grgić-Hlača et al., 2018; Iosifidis and Ntoutsi, 2019; Grari et al., 2020). One approach consists of building group-specific classifiers to achieve higher fairness for each group (Calders & Verwer, 2010; Ustun et al., 2019). Another approach represents an adaptation of the training process, for example, by setting a threshold for a fairness measure as a secondary exit criterion for the training process (Kamishima et al., 2012; Grgić-Hlača et al., 2016; Grgić-Hlača et al., 2018; Goel et al., 2018). Since most training concepts of DDMs are optimization problems, partly even non-deterministic ones, this latter approach does not necessarily lead to an accuracy-fairness trade-off. The result of the training is generally not the best possible DDC, but one of a large number of 'very good' ones. This is called the '*Rashomon Effect*' (see Definition 20). By incorporating a fairness measure into the training process, it is possible, at least in theory, to find a DDC that has about the same classification quality as a DDC that was not optimized for fairness at the same time (D'Amour et al., 2022, pp. 30-32). In practice, however, the literature shows at least a small reduction in accuracy (Kamishima et al., 2012; Grgić-Hlača et al., 2016; Grgić-Hlača et al., 2018; Goel et al., 2018).

Definition 20 (Rashomon Effect)

The '*Rashomon Effect*' describes the situation of having a multitude of different functions (DDCs) with about the same minimum error rate in a test data set that are completely dissimilar in their decision-making structure and can thus produce completely different outputs on new data (Breiman, 2001). A set of such functions is also called a '*Rashomon Set*' (Fisher et al., 2019). The effect is particularly significant for training and testing complex DDCs, since it is possible, in principle, to find an alternative DDC within a Rashomon set that is just as good as the one already found with respect to a primary goal but performs better with respect to a secondary goal.

Post-Processing Methods aim at correcting decisions made by the classifier (see, e.g., M. P. Kim et al., 2019; Lohia et al., 2019) or modifying the classifier after training to accomplish fairness (see, e.g., Fish et al., 2016; Woodworth et al., 2017; Dwork et al., 2018). Kamiran et al., 2012, for example, propose two easy-to-understand methods: *Reject Option based Classification* aims not to reject labels with high uncertainty, but to determine them in such a way that they contribute to more fairness. The second method, called *Discrimination-Aware Ensemble*, uses an ensemble of classifiers to find data instances on which the classifier to be adjusted does not match the others. Then the result of a classifier that improves fairness is selected. It is important to note that increasing the fairness of any learned classifier post-hoc inevitably reduces predictive accuracy at least for some individuals (Bechavod & Ligett, 2018). It can

also happen that the advantaged group is treated worse by such methods, for example, by achieving rather bad quality measure values. This is not in line with the basic idea of fairness or quality. Mittelstadt et al., 2023, Chapter 2, refer to this phenomenon as *'levelling down'*. As a solution, they suggest allowing only fairness-promoting modifications that result in better treatment of the disadvantaged group (*'levelling up'*, Mittelstadt et al., 2023, Chapter 6).

If the test results are unsatisfactory during external tests, the auditor can pass a negative judgment and thus induce consequences for the actor.

5.4.3 Applying Test Methods for Assessing Bias in the Context of Black-Box Audits

The results of tests alone are not sufficient to provide evidences for quality and/or fairness. Test methods can be incomplete or poorly executed; the test data can be produced, selected, or pre-processed in a way that hides unwanted (unexpected) behaviors; the test results can be manipulated or outright faked. That is why it makes sense to have external parties carry out an audit process. If an external auditor has full access to the system under test, for example, during a 2^{nd} or 3^{rd} party audit according to ISO 19011[18] (see Section 4.1.1), they can assure that all tests are performed thoroughly and the results are not manipulated. However, if the external auditor has no access to the system under test beyond the access everyone else has as well and an audit according to Sandvig et al., 2014, is to be performed (see Section 4.1.2), the list of applicable bias test methods may be limited:

Table 5.1: Compatibility of test methods and auditing concepts for black-box analyses (Krafft et al., 2023). The yellow boxes indicate restricted compatibility, which will be elaborated further in the following section.

	Non-Invasive User Audit	Crowdsourced Audit	Sock Puppet Audit	Scraping Audit
Group fairness measures based on quality measures	✗	✓/✗	✓/✗	✓/✗
Group fairness measures not based on quality measures	✓	✓	✓	✓
Individual fairness	✓	✓	✓	✓
Counterfactual fairness	✗	✗	✓	✓
Adversarial Testing	✗	✗	✓/✗	✓/✗
Differential Testing	✗	✓/✗	✓/✗	✓/✗
Metamorphic Testing	✓/✗	✓/✗	✓	✓

In the context of a *Non-Invasive User Audit* or Crowdsourced Audit, the influence of user profiles can only be determined statistically, as the profiles of the participants already have a history. On the one hand, it is hardly

[18] Guidelines for auditing management systems.

possible to make specific changes to the test setup in order to take the influence of this history into account or to test in a more targeted manner. On the other hand, as long as users are selected uniformly at random, it may be possible to detect bias with respect to the user profiles, given the profiles are also available for testing purposes. *Metamorphic Testing* may also only be a valid option if sufficient user profile information from the participants is available in addition to the submitted data to evaluate the metamorphic relations of interest. *Counterfactual fairness* is not an option, at least not in most cases, as it may not be possible to construct and submit sufficiently accurate counterfactuals, especially considering the possible impact of profiles. Further applicable test methods depend on whether an oracle is available or not (see Table 5.1).

In a *Non-Invasive User Audit*, the auditor has no influence on the inputs made by the user. Thus, fairness measures based on a ground truth can most likely not be computed, as the auditor cannot specifically select or construct test inputs for which an oracle is available. The evaluation of the data (in the context of bias) is limited to a manual examination of individual cases and statistical evaluations, such as *individual fairness* and *group fairness* measures not based on a ground truth.

To perform *Differential Testing* in a *Crowdsourced Audit*, the users must submit their input to both systems and retrieve both outputs. The results can then be used for further investigation.

When applying a *Sock Puppet* or *Scraping Audit*, the behavior of the bots or the queries to scrape with can be modified at any time. This allows the application of any of the previously discussed test methods. Still, the selection may be limited depending on whether an oracle is available, whether it is possible to interact directly with a system, and whether the test should be automatable (see Figure 5.5):

- If no oracle is available, it is not possible to calculate fairness measures based on a ground truth (F).
- If the system can only be reached via a mediator, it is not possible to make the adjustments to the input at runtime that are necessary for *Adversarial Testing* (A).
- If the tests are to be performed on a regular basis and are therefore to be automated, individual evaluation based on *Differential Testing* (D) is not an option.

Consequently, *Metamorphic Testing* (M) or fairness measures not based on a ground truth (N), including *individual fairness* and *counterfactual fairness*, can always be applied. In general, a *Sock Puppet Audit* is useful when the influence of human interaction is to be part of the evaluation (like typing or clicking) or if there is no API by which a system can be systematically tested (see, for example, Krafft et al., 2020). The challenge of creating bots that are not recognized as bots by the system under test should not be underestimated (Krafft et al., 2020).

In the context of a *Scraping Audit*, access to an API may be limited to a given number of submissions per time or to a certain maximal number

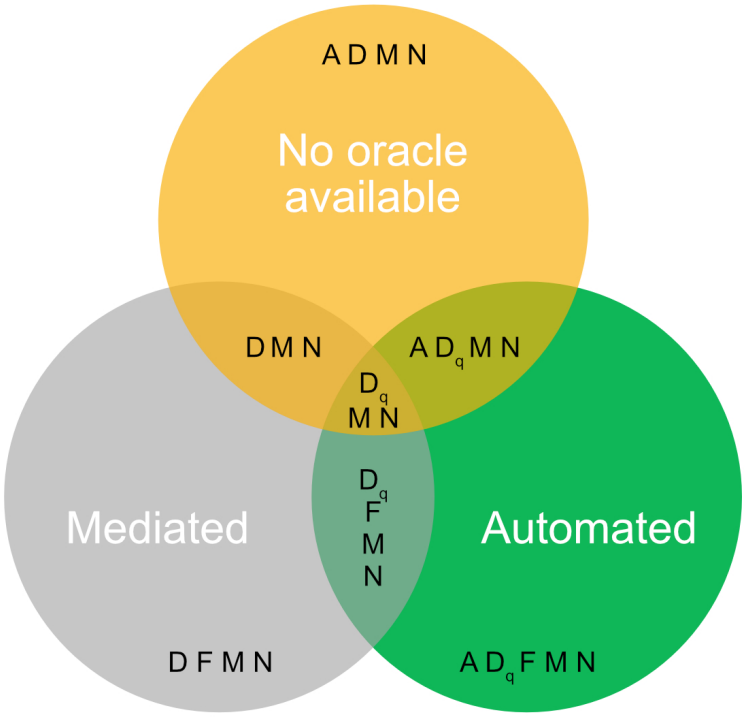


Figure 5.5: Venn diagram showing how the questions of whether an oracle is available, whether interaction with the system can only take place via a mediator, and whether the test should be automatable limit the selection of possible test methods in the context of a *Scraping Audit* or *Sock Puppet Audit* (Krafft et al., 2023). A = *Adversarial Testing*, D = *Differential Testing*, D_q = *Differential Testing* (quantitative evaluation only), F = *Fairness measures based on a ground truth* and *counterfactual fairness*, M = *Metamorphic Testing*, N = *Fairness measures not based on a ground truth* including *individual fairness*.

of input data. This problem can be addressed if the audit is extended to include a crowdsourced approach, by having many participants who either make their own queries and donate the results for evaluation or transmit prepared queries.

RQ 8

Which test methods for fairness are applicable in the context of which audit procedures according to Sandvig et al., 2014, and under which conditions?

Answer: Which test methods for fairness are applicable in the context of which audit procedures according to Sandvig et al., 2014, is summed up in Table 5.1.

The calculation of *group fairness* measures based on quality measures is generally only possible if there is an oracle. In the context of a *Sock Puppet Audit* or *Scraping Audit*, inputs for which this is the case can be selected specifically.

Adversarial Testing can only be implemented meaningfully if iterative adjustments of inputs at runtime are possible, i.e., if there is no mediator. This generally rules out its implementation in the context of a *Non-Invasive User Audit* or *Crowdsourced Audit*.

The individual evaluation of *Differential Testing* procedures is not an option if tests are to be automated, regardless of the audit procedure.

Metamorphic Testing may only be a viable option if sufficient user profile information is available or if user profiles are not relevant.

Further details will be elaborated in Section 5.4.3.

5.5 Test Case Generation

There are two general kinds of test case generation. For statistical evaluations (which is the case for all *group fairness measures*), the goal is to make a selection of test data with respect to a certain criterion or distribution. For example, the selected data should reflect real-world distributions as well as possible, or different subgroups should be equally represented, regardless of the real-world distribution (Moser, 1952).

When a test method aims to find single erroneous results (which is the case for all tests that make qualitative evaluations, but also for *Metamorphic Testing*; see Section 5.4), however, the goal is either to achieve the best possible test coverage, which means testing for all relevant eventualities, or to systematically search for inputs that result in unwanted outputs, like *Adversarial Testing* does. For real-world applications, it is usually not possible to test the entire input space, regardless of which test is to be performed, due to the combinatorial explosion of possible inputs. Therefore, there are various methods for reducing the number of test cases by limiting them to certain input parameter combinations, which are discussed under the broader term of **Combinatorial Testing** (Nie and Leung, 2011; D. R. Kuhn et al., 2013). Various methods for test case generation that are to be used for tests aiming to find single erroneous results will be discussed in the following.

Manual testing of a complex system is bound to be incomplete, no matter how many people are involved. However, there are test case generation approaches that aim to support more thorough manual testing. **Exploratory Testing** refers to generating test cases based on human intuition. The idea is based on the assumption that a human tester gains experience over time and sooner or later is able to guess which test cases are good for finding specific defects. Itkonen and Rautiainen, 2005, Chapter 3, review various nuances of this concept. The ISTQB glossary additionally describes **Checklist-Based Testing** as *'an experience-based test*

technique whereby the experienced tester uses a high-level list of items to be noted, checked, or remembered, or a set of rules or criteria against which a product has to be verified'.[19] To further support *Exploratory Testing*, so-called **Test Tours** can be defined. A *Test Tour* is a description of a typical user or user behavior the tester simulates. A *Test Tour* can be defined for each expected user group, but also, for example, for a user who tries to play the system, a user who tries to favor a certain group of people, or a user who does not try to complete a task, but instead tries to break[20] the application (e.g., through constant distortion of partial windows of an application) (Whittaker, 2009, Chapter 4). *Exploratory Testing* approaches have a random component, as different people do things differently and even the same person will not always test the same way in such an approach. Kroll et al., 2017, pp. 653-656, provide a strong argumentation that random components in testing increase the validity of tests.

Pairwise Testing is based on combining any two factors at least once. If there are three input parameters, each of which can have three discrete values, then 27 unique value combinations are possible. Pairwise, however, each combination will occur three times in this way (e.g., 1Aα, 1Aβ, 1Aγ, etc.). By selecting the inputs in such a way that each pairwise combination of parameters occurs at least once (if possible, exactly once), the number of value combinations to be tested is reduced to nine (see Figure 5.6).

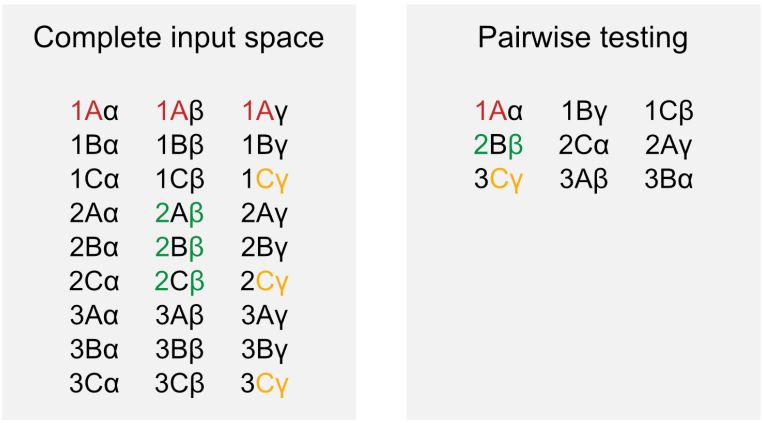


Figure 5.6: Considering an input size of three parameters x_1, x_2, x_3, each of which each can take on three discrete values $x_1 = 1, 2, 3$, $x_2 = A, B, C$ and $x_3 = \alpha, \beta, \gamma$, the complete input space consists of 27 combinations. With *Pairwise Testing*, this space is reduced to nine combinations. The complete input space contains each input pair three times, whereas with *Pairwise Testing*, the same pair occurs only once.

In the case of continuous values, there needs to be some sort of discretization that divides the value space into ranges. This can drastically reduce

[19] https://glossary.istqb.org/en_US/term/checklist-based-testing, last accessed on April 26, 2023.

[20] Traditionally, 'breaking' the system refers to triggering errors. In the context of bias testing, this also means triggering discriminatory behavior or outputs.

the number of tests to run while still providing well-distributed coverage of the underlying decision rules (Cohen et al., 1996). **Orthogonal Array Testing** is a special form of *Pairwise Testing* that allows the combination of any n factors (instead of just two) and requires each possible combination of parameter values to be tested equally often (Hedayat et al., 2012, p. 2). The choice of n is a test design decision.

Traditional **Fuzz Testing** is used to find test cases that result in system crashes. For this purpose, a collection of initially random input vectors is generated and passed to the system for execution. In each iteration, the queried input vector is mutated, which means that the initial random input is changed within a certain range (for example, on a limited number of parameters or in a predetermined order of magnitude). If the system output encounters an *interesting behavior*, the corresponding input is stored. In the context of debugging, an interesting behavior could be, for example, an error message that does not lead to a system crash or longer delays in the computation. The generation of further input vectors focuses increasingly on the mutation of inputs that previously triggered interesting behavior (Klees et al., 2018). This method of systematic test case generation can be used for bias tests on a scoring system. First, a random input vector is generated and duplicated with different values of the protected attribute ($X_A = X_1, \ldots, X_n, A$ and $X'_A = X_1, \ldots, X_n, A'$), for which the system computes the outputs ($\hat{Y}$ and $\hat{Y}'$). The procedure is repeated with the mutated inputs ($X_\epsilon = X_1 + \epsilon_1, \ldots, X_n + \epsilon_n, A$) until the difference between the outputs for inputs that only differ in the protected attribute exceeds a certain threshold ($|\hat{Y}_\epsilon - \hat{Y}'_\epsilon| > \tau$), even if this deviation would not yet lead to a different categorization based on the score. These input vectors can now be used as test cases or as inputs that result in interesting behavior for generating additional, randomly mutated input vectors.

Such test case generation methods are discussed especially in the context of *Differential Testing* and *Metamorphic Testing* (Petsios et al., 2017b; Zhu, 2015; see Section 5.4.2).

5.6 Test Schemes

Which test activities should be performed and how many is difficult to determine. While certain test concepts are particularly suited for testing for certain goals, there is no guarantee that the chosen test activities will be sufficient. By following different testing schemes, more holistic testing can be achieved.

For **A/B Testing**, two or more systems (or variants of the same system) are compared by performing the same test activities on them (Kohavi & Longbotham, 2017; Young, 2014). This procedure is also known as **Bucket Testing**, **Split Testing**, or **Controlled Experiment** (Xu et al., 2015). *A/B Testing* does not assess *'what would happen if a data point had a different value'*, but rather *'what would happen if the system had a different decision structure'*, which is why it is also known as **Counterfactual Reasoning** (Gilotte et al., 2018). *A/B Testing* is frequently used to improve different variations of configurations of online platforms in terms of user

experience (Cruz-Benito et al., 2017; Siroker & Koomen, 2013). It is a meta-testing framework based on any test activities to evaluate a property. Breck et al., 2017, also discuss the concept of testing a complex model (e.g., an ANN) against a simple one (e.g., a rule-based approach) to assess the need for a complex model. Thus, *A/B Testing* is also useful for testing whether a white-box model or even a rule-based approach would be sufficient. Any test method that can be applied to a given system under test can also be used in the course of *A/B Testing* (Kohavi & Longbotham, 2017).

Differential Testing (see Section 5.4.2) can be understood as a special form of *A/B Testing* in which the system under test is compared to a reference system. Instead of keeping the system that performs better as a consequence of the test result, the system under test is kept in any case and improved in the case of a negative result.

In traditional software engineering, the term **Black-Box Testing** has been coined for testing activities that are intended to treat the system to be tested as a black box, regardless of whether it actually is one (Krafft et al., 2021). In some literature, this kind of *Black-Box Testing* is synonymous with *Functional Testing* (Nidhra & Dondeti, 2012). In the context of an external audit as described by Sandvig et al., 2014 (see Section 4.1.2), it does not matter whether an ADM system has been constructed as a white-box system (e.g., a decision tree) or a black-box system (e.g., an ANN) because the auditor is not able to perceive this information. The system is treated as a black box in any case. Therefore, all test methods explained in Section 5.4 can be used for *Black-Box Testing*. Based on this consideration, *Black-Box Testing* could also be understood as a test concept according to the taxonomy presented in this thesis.

Code Coverage describes a measurement of how much of the code of a software system is functionally tested (Y. W. Kim, 2003). There are various approaches that measure different coverage aspects; for example, **Decision Coverage** checks whether there is at least one test for each conditional branch in the code (Myers et al., 2004, p. 45). 100% **Test Coverage** means that every path in the code is covered by automated tests. This is often not possible and often not useful (Marick et al., 1999). In many cases, testing certain branches does not add any value. Too many unnecessary tests quickly lead to a time-consuming or even unmanageable process, making test execution impractical. The scientific literature refers to code coverage more as an analysis tool or metric than as a test method in its own right (e.g., Shamshiri et al., 2018, p. 250; Sevinchan et al., 2020, p. 449). However, if test collection is considered as a test object, *Code Coverage* can certainly be understood as a separate test method for assessing test coverage. For **Mutation Testing**, faulty variations of a program are generated. Then the implemented test methods are used to check whether the faults are detected or not (Papadakis et al., 2019, Chapter 2: Background). Thus, it is also a procedure for checking test coverage. **Extreme Mutation Testing** even goes one step further and analyzes whether a test suite recognizes when the functionality of a method is removed (Betka & Wagner, 2021). **Neuron Coverage** is a rather novel approach, tailored to assess the test coverage of (deep)

ANNs (see Section 2.1.2). A common definition of *Neuron Coverage* of a test suite is the ratio of the number of neurons activated at least once for all test inputs and the total number of neurons in an ANN (Pei et al., 2017, p. 6). However, whether *Neuron Coverage* really represents a significant measure is still the subject of critical debate (e.g., Harel-Canada et al., 2020; Z. Yang et al., 2022).

Modern software systems are subject to constant change. With **Regression Testing**, the test results of an older version can be compared with the test results of a newer version. This makes it possible, for example, to determine the time frame in which new errors have appeared, and thus which changes may be associated with them (Leung & White, 1989). As classification systems rarely achieve 100% correctness on larger test data sets, *Regression Testing* can be used to check how a change of the code or the model (e.g., through continuous learning in operation) affects the classification of the test data set. This allows weaknesses in the system under test to be identified and addressed better. *Regression Testing* can only be applied to test cases that have been carried out on at least one earlier product increment and are thus static. Testing based on static test cases is also called **Offline Testing**. Testing based on dynamically generated test cases (also called **Online Testing**), such as *Adversarial Testing* (see Section 5.4.2) is not suitable for *Regression Testing* (Utting et al., 2012, p. 306).

Testing under laboratory conditions (as opposed to testing in the field) is important, but the results may not fully reflect the system behavior in operation (e.g., due to artificial test data, a system that is blocked for inputs from outside the testing procedure, or even manipulations; see Example 9, p. 132). In addition, in the long run, an increasing number of AI-based systems can be expected that continue to learn in use, which means that test results at the time of release lose their significance. In order to counteract this problem, it makes sense to implement tests to be executed during application of the system. Tests can also be executed automatically on a regular basis, which allows inspection of the collected results at fixed intervals, for example, during regular audits. To further improve the process, the documentation can be limited to failed tests, error aggregations (such as an error rate), or the computation of carefully selected performance measures based on batches of inputs (e.g., after n inputs, compute for the last n outputs the fairness measure of *Separation* (see Table 2.3, p. 37) with regard to gender fairness and document it if it is below 95%). Another option would be to alert the operator every time a test fails or to continuously document and display test results on a dashboard. The ISTQG glossary uses the term **Field Testing**,[21], and in Art. 61 AI Act (proposal)[22] the term **Post-Market Monitoring** is used to refer to such schemes. Strictly speaking, the two terms need to be distinguished. A test ends with a clear result – it is temporally complete. Monitoring is a continuous process whose current results can be inspected at any point

[21] https://glossary.istqb.org/en_US/term/field-testing, last accessed on April 27, 2023.

[22] European Commission, Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts COM(2021) 206 final.

in time. Even if *Field Testing* approaches only take effect after the deployment of the software, they can be implemented long before, just like all other test activities. When exactly to implement tests highly depends on the test development process.

Example 9 (Dieselgate)

In 2015, the German automaker Volkswagen was accused of defrauding emission tests, as field tests showed major discrepancies compared to the laboratory test results (Fracarolli Nunes & Lee Park, 2016). Later, it turned out that software had been implemented that was able to detect a test environment and then switch to a mode that emits less pollutants (Zhang et al., 2021, p. 81), but consumed much more AdBlue, a liquid used to neutralize harmful nitrogen oxides and thus clean diesel gas emissions (Demir et al., 2022). As it was not possible to install a sufficiently large tank for AdBlue to enable continuous operation of the emission-friendly mode, the actual emissions on the road were significantly higher than legally permitted. The Volkswagen emissions scandal became commonly known as Dieselgate affair (Di Rattalma, 2017).

With a mandatory *Post-Market Monitoring* approach, for example, where emission and consumption values are measured continuously (or at least regularly), a comparable fraud approach would be much more difficult to implement and cover up. However, this would also require a corresponding continuous (or regular) measurement to be technically feasible.

5.7 Test Development Processes

The most naive test development process is to write tests only when an error or problem occurs. This approach obviously contradicts the general idea that tests are written to find errors before they occur. So the test-on-demand approach has more to do with debugging than actual testing.

Traditionally, the right time to perform test activities is once the components to be tested have been implemented. This seems intuitive at first, as it is only possible to carry out very limited tests in advance, at least if they are to pass. The *test-first approach*, however, proposes doing exactly that, namely, to write and execute tests before the code to be tested has been implemented. This approach is intended to prevent the tendency to write tests to match the code. Developers are forced to think about all eventualities in advance and take them into account in the code. If the code is written first, it can easily happen that scenarios are not considered and the corresponding test cases are also (consciously or unconsciously) forgotten. At the same time, concerns of adapting the

code to be tested later are reduced, since test cases (should) exist for all eventualities. There are two major test development processes that follow the test-first approach: *Test-Driven Development* (TDD, see Section 5.7.1) and *Acceptance Test-Driven Development* (ATDD, see Section 5.7.2).

5.7.1 Test-Driven Development

In agile software development processes, *Test-Driven-Development* (TDD) is often considered the obvious choice (e.g., Shore and Warden, 2021, p. 353). TDD is a cycle in which a test is first written to check whether the code does what it is supposed to do, even before the code itself is written. If the test passes ('the test runs green'), the functionality tested for is already available and the development of new code is unnecessary. If the test fails ('the test runs red'), the next step is to try to run the test green with as little code as possible. Finally, the last step of the cycle begins: *refactoring*. This is an attempt to optimize the code, although it is not specified whether the optimization is an improvement in performance or in readability (these two types of improvement are sometimes contradictory to each other at a certain level of functional complexity). Once the refactoring phase is complete, the test might fail again due to mistakes in the changes; thus, the cycle begins again until the code is no longer improved and the test passes (see Figure 5.7). To test a functionality, a single test is usually not sufficient. In the sense of TDD, 'one test' means a collection of tests for a single functionality.

Following the TDD approach yields various benefits. Developers are required to think carefully in advance about the architecture of the code to be written. Interfaces between components used for testing are assumed to be the way a developer wants them to be. This forces developers to produce clean code, as it should be easy to use in further development and not, initially, easy to program (these two concepts are also often antagonistic to each other, at least for inexperienced developers). The simple interfaces also make it easier to replace components later on and, for persons not involved in the development process, to audit the code (e.g., for *Black-Box Testing*; see Section 5.5). Simple interfaces may also make the creation of a governmental test database for governmental reviews of DDCs more interesting (e.g., governmental benchmarks in the context of fairness testing).

Writing code as efficiently as possible from the start often carries the risk of making mistakes due to complexity. Since the process demands that the code be written only functionally first and then optimized, this risk is preempted. In addition, the process of improving code incrementally helps programmers to develop their skills.

Nevertheless, TDD also has some limitations. The idea is to write only 'small-scale tests' with this approach (Beck, 2003, p. 199). Since there is no concrete definition of this term (Beck, 2003, explicitly mentions that the term does not match the accepted definition of Unit Tests), there is some disagreement about how it should be understood.[23] Special care must be taken to write meaningful, beneficial tests. Developers may tend

[23] According to Gärtner, 2012, p. 118, TDD is used for Unit Tests only. Shore and Warden, 2021, p. 300, also include various forms of Integration Tests.

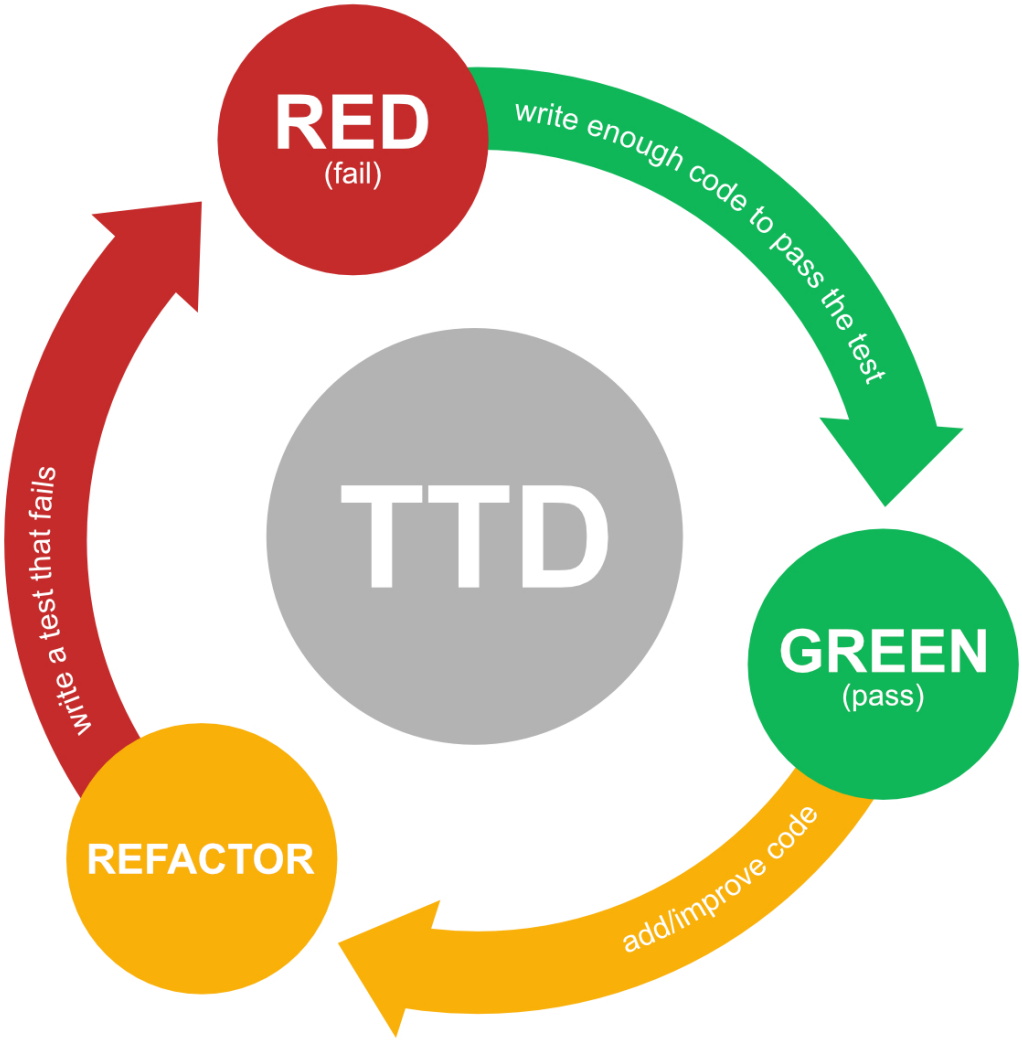


Figure 5.7: The Test-Driven Development (TDD) cycle.

to test every class or even every method because of the TDD framework. Especially with very large software products, both wrong tests and an extremely large number of tests can lead to tests taking a very long time to complete (Marick et al., 1999). This can also be to the detriment of the long-term time savings that are bought with the additional time invested in applying the TDD framework. The focus on functional sections usually means the implementation of unit tests and integration tests. TDD could also be used for system tests, although here again, there is some disagreement among experts since above a certain complexity, it is no longer possible to recognize every case to be tested and refactoring is also only feasible to a limited extent.

As already discussed in Section 5.1, unit and integration tests are not suitable for testing the results of a DDC. To apply the idea of test-first approaches at the system test and acceptance test level, the concept of Acceptance Test-Driven Development (ATDD) has been introduced.

5.7.2 Acceptance Test-Driven Development

The goal of Acceptance Test-Driven Development (ATDD) is to prove that a user story[24] or a requirement has been fulfilled. It is hard to say since when the idea of ATDD has been around, as there are various other names for comparable ideas (e.g., **Behavior-Driven Development**, **Specification by Example**, **Agile Acceptance Testing**, and **Story Testing**; see Gärtner, 2012, p. XV). The reason could be that there is hardly any scientific literature on the topic (e.g., Andersson et al., 2003); there are, however, a lot of blog posts from software developers, training courses, and certifications for prospective testers (e.g., by ISTQB[25]), and some books (e.g., Pugh, 2010), which do not necessarily meet scientific standards. Therefore, the following explanations follow the interpretations of Gärtner, 2012.

In order to verify compliance with requirements, it is necessary for the requirements to be well formulated. There are various guidelines on how to write good requirements, one of which is the '*SMART*' approach. SMART stands for specific, measurable, attainable, realizable and time bounded (Mannion & Keepence, 1995).[26] Despite such guidelines, it happens that '*software projects fail to deliver what their customers request*' (Gärtner, 2012, p. XIII) due to a divergent understanding of the task. However, if tests for verifying compliance with requirements are written in advance together with the customers, the risk for divergent understanding diminishes greatly.

There are many descriptions, online and in the literature, of how ATDD is performed as a process. Personally, I prefer those that emphasize the connections with TDD, as Ken Pugh does (Pugh, 2015). In the following, his process is described and supplemented with my personal experiences with the concept (see Figure 5.8, p. 137):

1. **Elicit Requirements**. As so often in this context, many ways have been described how to do this (Sommerville & Sawyer, 1997, Chapter 4). One frequently named option is to work with *specification by example*. This technique requires developing examples that show how the system to be tested is expected to work and examples that show how the system is expected not to work. The examples do not need to be limited to strictly functional aspects but could also contain legal requirements (such as protection against discrimination) or (legally optional) moral requirements (what this may mean will be addressed in Section 6.2). The examples are documented in such a

[24] A user story is a short, informal explanation of a specific software feature written from the perspective of the user. Traditionally, user stories provide an informal context for the developers to discuss why they are building what they are building and to develop requirements based on these insights (Shore & Warden, 2021, p. 130). However, the general idea is often (mis)used to let users define testable requirements.

[25] https://glossary.istqb.org/en_US/term/acceptance-test-driven-develop, last accessed on April 27, 2023.

[26] SMART was originally meant as an aid for managers to describe meaningful goals. At that time, the acronym stood for specific, measurable, assignable, realistic and time-related (SMART, 1981). To date, many alternative interpretations of the acronym have developed, but their meanings are very similar. Mannion and Keepence, 1995 appear to be the first to have applied the term to requirements engineering in software development processes. Accordingly, the focus here is on their interpretation.

way such that anyone who might later inspect any part of the quality assurance process will be able to understand them (Gärtner, 2012, Chapter 9). A popular format for writing down examples is provided by so-called *controlled natural languages* (T. Kuhn, 2014), such as the simple description language Gherkin. It follows a syntax based on few keywords, like: given [...] when [...] then [...]. No matter how the requirements are derived, it is important to ensure that they are formulated in a testable manner.

2. **Analyze Requirements with Tests**. No matter how carefully requirements are formulated and communicated, there can always be inaccuracies and misunderstandings. By writing acceptance tests based on the requirements (core cases, edge cases, error cases, etc.) that state how fulfillment of the requirements is validated, the requirements are analyzed precisely. The tests are discussed with all internal stakeholders (business user, product owner, etc.) to ensure that all requirements/user stories are covered and that they are in line with the stakeholders' expectations. During the first execution, all tests should fail.

3. **Design**. Once all acceptance tests for validating that the system behaves as expected have been prepared, traditional software design processes can start. This step is performed independent of whether ATDD is applied or not.

4. **Code with TDD**. Once the system design has been prepared, the actual coding and training of a DDC starts. If the code consists of many functional blocks, the process starts again from 1 for each block. As soon as the process gets to the unit test level, TDD is applied. This means that the requirements can be fulfilled incrementally. If requirements that have already been fulfilled are no longer fulfilled due to later changes in the code or further training, this will be noticed by the tests that have already been written. This is another benefit of test automation.

5. **Deploy**. Once all acceptance tests have passed and all TDD processes (including refactoring) are completed, the software is ready to be deployed.

Based on this process, tests are written before there is a concrete idea of how the system should comply with the requirements. This is exactly the strength of ATDD. It is not for testing code, but behavior. This is especially important in the context of DDCs. If the training data is selected first, it can easily happen that scenarios are not considered and corresponding test cases are therefore also forgotten (consciously or unconsciously). At the same time, it might become necessary sooner or later to retrain the DDC with modified data; for example, because the circumstances of the application of a system have changed, more current or reliable data is available, or data of the same type that previously triggered unwanted behavior is available. Retraining a model is a delicate process, as it introduces new potential for unwanted behavior. However, the acceptance test cases developed with ATDD are generally valid and independent of any training data or specific ML methods. The fear of consequences following

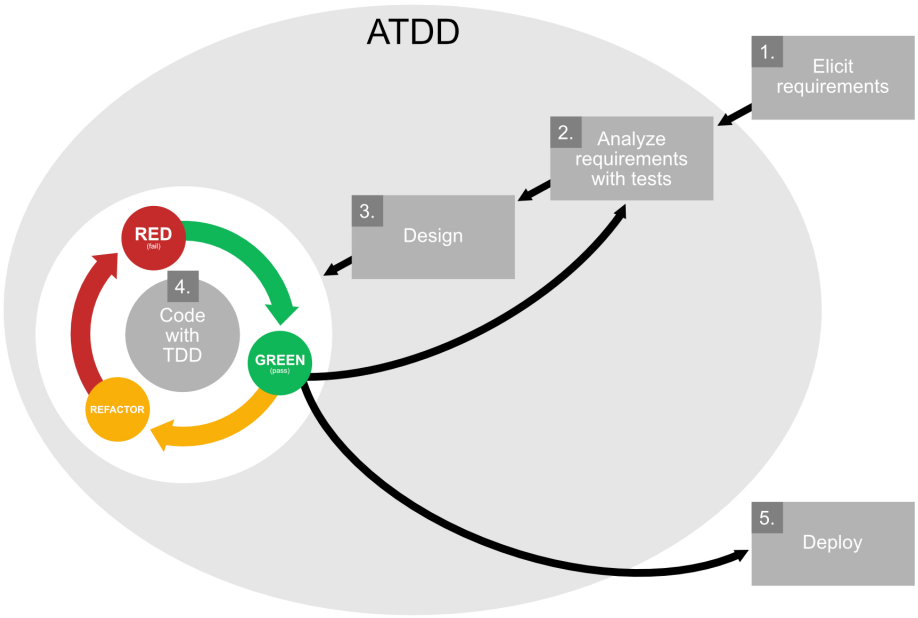


Figure 5.8: The Acceptance Test-Driven Development (ATDD) cycle.

changes to the training data can be avoided by performing such tests. Therefore, this also facilitates the introduction of certain testing schemes, such as *A/B Testing* or *Regression Testing* (see Section 5.6). If the tests are automated, they even serve as a warning bell for systems that keep learning during their application and thus serve as a strong basis for *Field Testing* (see Section 5.5) as well. Most of these benefits are also described in the (scarce) scientific literature reporting about hands-on experiences with the framework (e.g., Andersson et al., 2003).

In the context of the accountability considerations from Chapter 3, ATDD might help to resolve the question of which actor (stakeholder) is responsible for what effects of applying the system under test (independent of wanted or unwanted behavior) and should thus face the consequences resulting from the forums' judgments. Consider a client who wants a DDC and a developer team that demands specifications for their tests in advance. For all requirements, it can be specified which actor is responsible if this requirement is not met in operation. This also ensures that all actors take their role in eliciting requirements and analyzing them with tests (Phases 1 and 2 of the described ATDD process) seriously. At the same time, the framework can also help actors defend themselves against unjustified accusations. This aspect will be discussed in more detail in Section 5.7.3.

Developing software with ATDD, however, increases the effort of requirements engineering. All stakeholders have to be brought together several times, or at least an efficient way has to be found to create consensus between them. Furthermore, it may require an iterative process of successively specifying requirements, developing tests for them, and checking fulfillment of the requirements through tests (together with the stakeholders). It is difficult to determine whether the advantages described sufficiently compensate for this additional effort; after all, it is

not economical to develop the same product twice in parallel, once with and once without ATDD, just for comparison.

RQ 7

Which test-related terms are especially relevant in the context of AI-based ADM systems?

Answer: The literature on test-related terms is extensive and often inconsistent. Sections 5.2- 5.7 present different abstraction levels of test-related terms and describe which test-related terms that are especially relevant for AI-based ADM systems can be categorized under each. The explanations are compatible with most of the literature and explicitly point out ambiguities and contradictions.

5.7.3 Acceptance Test-Driven Development and Assurance Cases

ATDD has some desirable advantages (at least in theory). However, the framework does not provide a structured approach of how exactly requirements can be determined. This is where a combination with the Assurance Case framework comes in handy. Building an argumentation structure before the actual development starts corresponds to the first phase of the ATDD process (elicit requirements). The description of testable or validatable evidences is the necessary preparation for the second phase of the ATDD process (analyze requirements with tests), in which automated procedures for the provision of evidence are implemented (except for evidences that are just documents that need to be kept). The discussion of the tests and whether they cover the requirements is an implicit part of the Assurance Case development process (phases 5 and 6), as all stakeholders (or appropriate stakeholder representatives) can ensure during their contribution that the evidences that are important to them are included. It is worth noting that Assurance Cases are based on a broader definition of stakeholders, which should be kept for the combined approach.

When arguing the fulfillment of extra-functional requirements such as fairness, the translation from evidences to tests may be quite challenging. To solve this problem, the evidences need to be formulated so precisely that the software developers (tester) have no room for interpretation when writing the tests. However, the stakeholders can hardly be expected to express concrete thresholds to be achieved by fairness tests (see Section 5.4.2). In order to arrive at such thresholds, it is necessary that both the stakeholders and technical experts (possibly a product owner is sufficient, but the developers themselves may also be needed), jointly define concrete test cases, which the developers can then put into practice. In the context of testing AI-based applications, interdisciplinary background knowledge (e.g., in the areas of law, standards, statistics, etc.) could be required as well.

Another solution could be to have the stakeholders also review the tests implemented by the software developers based on possibly underspecified evidences and their results, as intended by the stand-alone ATDD

process. However, if the stakeholders see that their requirements are not adequately addressed, an iterative process of refining evidence descriptions and tests that implement those descriptions starts, which delays the development process. Of course, it could make sense for the stakeholders to take a look at the final tests in any case, as an additional layer of validation.

In Hauer, Adler, and Zweig, 2021, we[27] elaborate the general idea of combining ATDD with the development of an Assurance Case using the example of assuring gender fairness in an unspecified application. To the best of our knowledge, no one has attempted any practical experimentation with our idea yet.

RQ 9

Can the Assurance Case framework be integrated into a test-driven development process?

Answer: The development of an Assurance Case is an iterative process in which all stakeholders (or their representatives) are involved directly or indirectly. As a requirements engineering technique, this process can begin even before the actual product development starts. This makes it an obvious candidate for embedding in agile development and testing processes, such as ATDD.

While ATDD aims to find acceptance criteria based on specific examples, elaborated by a diverse group of stakeholders, and to test for them, an Assurance Case argues that the criteria found with the help of ATDD are adequate and makes otherwise implicit assumptions visible. It provides a framework for structured argumentation and documents the results of discussions, whereby it reveals misconceptions and shortcomings.

[27] In Section 5.7.3, 'we' and 'our' refers to the authors of Hauer, Adler, and Zweig, 2021.

Chapter 6

Regulation and Corporate Digital Responsibility

In Chapter 3 of this thesis, a broad selection of different transparency and inspectability mechanisms is discussed that are suitable for achieving accountability in AI-based systems. Various implementation strategies, as presented in Chapters 4 and 5, can involve considerable effort and therefore costs. Nevertheless, companies have an intrinsic motivation to offer good AI-based products. After all, a bad product may perform worse on the market than a comparable, good product. Accordingly, it can be assumed that most companies establish various quality assurance mechanisms and conduct technology impact assessments on their own initiative. Nevertheless, there is always a balance between the financial value added by the additional effort and the associated costs. This balance is not necessarily in favor of customers or the people affected by a product.

The question therefore arises as to how additional incentives can be created to promote product quality and benevolence. There are at least two basic approaches that complement each other. The first approach is the regulation of AI-based products (see Section 6.1). The aim is to enforce minimum standards by adding extrinsic motivation, i.e., the threat of punishment. The other approach aims to expand intrinsic motivation by creating additional incentives for self-commitment (see Section 6.2). The discussions and findings presented in this section can also be transferred to regimes that fall in between these approaches, such as self-regulation (e.g., defining regulatory goals without prescribing the means to achieve them). Since a discussion of such approaches must be highly implementation-specific and I am not aware of any concrete implementation in the context of AI-based systems, this type of approach is not explicitly discussed here.

6.1 Regulation of AI-based ADM Systems

The challenges of making decisions and decision structures of ADM systems transparent and inspectable, and thereby making unfair or unjust decisions identifiable and contestable, as discussed in detail in Chapter 3, are one of the main justifications for risk-based regulation approaches (Krafft and Zweig, 2019; European Commission, 2021, Section 2.3; Orwat et al., 2022, p. 258; Krafft et al., 2022). This is an attempt to achieve an appropriate balance between a society's need for protection and entrepreneurial freedom (Orwat et al., 2022, p. 259) while adhering to the

precautionary principle (see Definition 18, p. 101). Risk-based regulation approaches are opposed to classical rights-based approaches, which define specific obligations to avoid risks that are applied equally to all subjects. Orwat et al., 2022, p. 279, argue that neither a purely rights-based approach nor a purely risk-based approach alone can meet the demands of regulating AI-based systems: Rights-based approaches assume that static rules are sufficient to achieve protection goals, ignoring the rapid advancement of socio-technical developments and the very different regulatory needs of AI-based systems depending on their intended use, performance, robustness, and safety.[1] Risk-based approaches are characterized by a large number of design decisions (see, e.g., van der Heijden, 2019), the concrete effects of which are difficult to assess in advance. Various forms of risk-based regulation approaches that are commonly discussed in the context of regulating software systems in general and AI-based systems in particular will be discussed in Section 6.1.1. Section 6.1.2 explains the risk-based regulation of AI-based systems as proposed in the AI Act[2], which is still being negotiated, and shows an assessment of how its first draft would affect the AI landscape in Germany.

6.1.1 Risk-based Regulation

The operationalization of risks to describe a risk level, often referred to as *risk assessment*, is the basis of every risk-based regulation. Different rules are assigned to each risk level, with stricter rules generally assigned as the risk level increases. The rules of the lower levels are retained, supplemented, and/or made more stringent.

One of the most commonly known methods to assess risk is defined as '*a combination of the consequences of an event (including changes in circumstances) and the associated likelihood [...] of occurrence*'.[3] These two aspects can be used as axes in a two-dimensional matrix, where the intersection of specific values for these two aspects states the level of risk (see, e.g., Dawson et al., 2019). When it comes to assessing the risk of AI-based ADM systems, though, there is a lack of practical experience and scientific knowledge that allows thoroughly anticipating consequences and the likelihood of their occurrence. To anticipate possible consequences, contexts, vulnerabilities, protection mechanisms, and threats (including faulty data and models) could be identified first. However, there may be very different types of potential (negative) consequences of an event involved, such as physical injury, damage to property, environmental damage, financial damage (private or corporate), impairment of informational self-determination, and/or impairment of critical infrastructure. This makes it necessary to carry out one risk assessment per consequence or to offset the different consequences against each other.

[1] Translated from the report of the German Data Ethics Commission. '*Aus regulatorischer Sicht legt die Tatsache, dass algorithmische Systeme je nach Einsatzzweck, Leistungsfähigkeit, Robustheit und Sicherheit sowie mit Blick auf ihre Wirkungen ethisch sehr unterschiedlich zu bewerten sind, einen risikoadaptierten Regulierungsansatz nahe*', Datenethikkommission, 2019, p. 173.

[2] European Commission, Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts COM(2021) 206 final.

[3] ISO Guide 73:2009, Risk management — Vocabulary.

Due to the interdisciplinary character of many AI-based applications, as well as systemic consequences due to emergent effects, the assessment of the associated likelihoods of occurrence poses additional challenges. Therefore, alternative parameters are being discussed among experts on the basis of which a risk assessment of AI-based systems could be more practicable. In the GOAL project (see section 1.2.1), the possible aspects discussed internally were (i) information asymmetry, (ii) monopoly position of a system vs. choice for the user/affected party, (iii) individual risks, (iv) societal risks, (v) damage potential (both worst-case scenarios and a collection of all conceivable risks), (vi) affected parties, (vii) chance of problems becoming known to the affected party, (viii) correction in the event of error/damage, (ix) duration until correction, (x) chance of asserting claims for correction, and (xi) whether there is a human in the loop. Many of these aspects are similar or have a direct influence on each other, but they are not the same. Furthermore, this is not an exhaustive list.

6.1.1.1 Risk Matrix

Krafft and Zweig, 2019 present a risk matrix based on similar discussions that is tailored more to the societal effects of ADM systems. It takes two aspects into consideration: the intensity of potential harm (x-axis) and the dependence of a person affected on the decision (y-axis; see Figure 6.1).[4] In our contribution to Hallensleben et al., 2020, we[5] break down which factors are at least relevant for the two dimensions. The intensity of potential harm addresses any potential harm to people, organizations, and society, independent of the likelihood. That means that the different consequences are offset against each other. Dependence on the decision addresses the options to avoid that potential harm. Relevant aspects are whether a human is in the loop, whether a person affected can choose to be affected by a different ADM system (or none at all), and the possibility of challenging or correcting an algorithmically made decision and its impact (see Example 10).

[4] In the referenced paper, the risk matrix has different axis labels. This is because the concept has been slightly adapted and refined over the years. The most recent version is published in Krafft et al., 2022, which shows another different label of the y-axis. I refer here to the version published in Hallensleben et al., 2020, as I only contributed to that one. The general idea is the same in all versions.

[5] In this section, 'we' refers to the authors of Hallensleben et al., 2020, Chapter 3, namely Tobias D. Krafft and I.

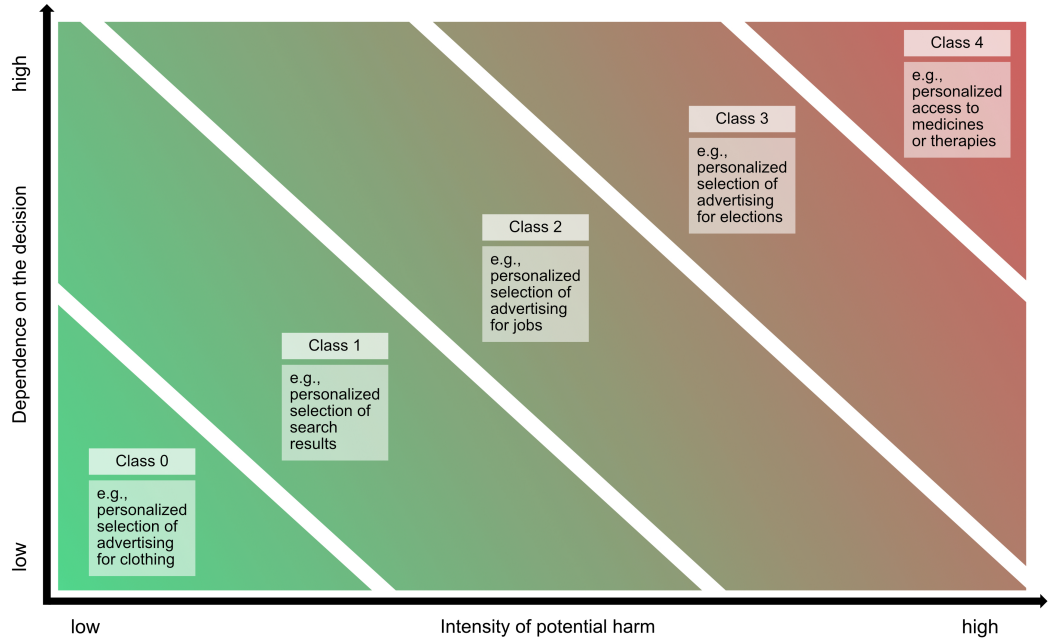


Figure 6.1: A risk matrix with five classes, including some examples of what application might fall under which class and would need to comply with the respective demands as adapted from Hallensleben et al., 2020, p. 36.

Example 10 (Robodebt)

From mid-2016 onwards, an ADM system in Australia called Centrelink automatically identified those who had wrongly received unemployment benefits or social assistance (Braithwaite, 2020; Cosier, 2017). In cases of suspicion, the system automatically sent out reminders without a human in the loop. More than 200,000 reminders were sent out, of which more than 20,000 were unjustified. Not only was too much money demanded back, but in some cases, more money was demanded than had actually been paid out. In various individual cases, this had fatal consequences for the affected persons (see, e.g. Karp, 2020). The issue became publicly known as Robodebt. Correcting the error sometimes took months and, in a few cases, even more than a year. During this time, affected persons, who as welfare recipients were already among the most vulnerable in Australian society, were additionally burdened instead of supported.

As the two dimensions are inherently complex in their internal structure, assigning values can be challenging. Application of this kind of risk assessment to real ADM systems has shown that in some cases, there are objective value conflicts that can affect both axes (e.g., private data protection vs. security) and thus further increase complexity. Therefore,

defining risk classes in individual cases makes the participation of actors with a broad, interdisciplinary perspective indispensable.

For risk-based regulation of AI-based ADM systems, a classification must be made into at least three different classes: a lowest class for ADM systems that are so uncritical that no regulation is necessary for them,[6] a highest class for AI-based systems with fatal consequences,[7], and at least one class between these two extremes. The risk matrix according to Krafft et al., 2022 proposes a division into five classes, with the specific demands in each class being deliberately kept vague for the most part. Krafft and Zweig, 2019 provide some recommendations on which actions should be taken depending on how a system is categorized. In Hallensleben et al., 2020, we further expanded these recommendations. A specific implementation of that recommendation could look like this:

> Class 0: No preventive action is necessary.
>
> Class 1: Voluntary and auditable (see Chapter 4) self-commitment (the benefits and caveats of self-commitment concepts will be discussed under the term *Corporate Digital Responsibility* in Section 6.2) is sufficient.
>
> Class 2: Accredited certification (see Section 4.1.1) is a prerequisite for product approval.
>
> Class 3: An approval procedure including regular *Field Testing* (see Section 5.5) is necessary. The ADM system may only be used as a recommendation system; a human in the loop is mandatory.
>
> Class 4: Under the given conditions, the deployment of the ADM system cannot be permitted.

The boundaries between the classes are only defined vaguely for the risk matrix in order to leave room for discretionary decisions in the context of concrete applications. A major drawback of such an approach is that the complex definitions of the dimensions and the lack of clear boundaries between the classes prevent companies from achieving legal certainty or eliciting specific requirements for compliance checks.

6.1.1.2 Risk Graph

Use of a risk graph, rather than a risk matrix, might be more suitable for dealing with the complexities involved. This approach is used, for example, by ISO 13849[8] to consider three dimensions in the classification of risks (severity of injury, frequency and/or exposure to hazard, and possibility of avoiding hazard or limiting harm) instead of only two. Figures 6.2 and 6.3 show example generic risk graphs, one for risk of physical damages and one for risk of non-physical damages. For a specific implementation, clear definitions of all the values each variable can take on, including an operationalization, need to be provided. To the best of my knowledge, such an approach has not been thoroughly discussed in the context of AI-based systems yet.

[6] For example, personalized product recommendations (excluding medical products).
[7] For example, lethal autonomous weapon systems.
[8] Safety of machinery — Safety-related parts of control systems.

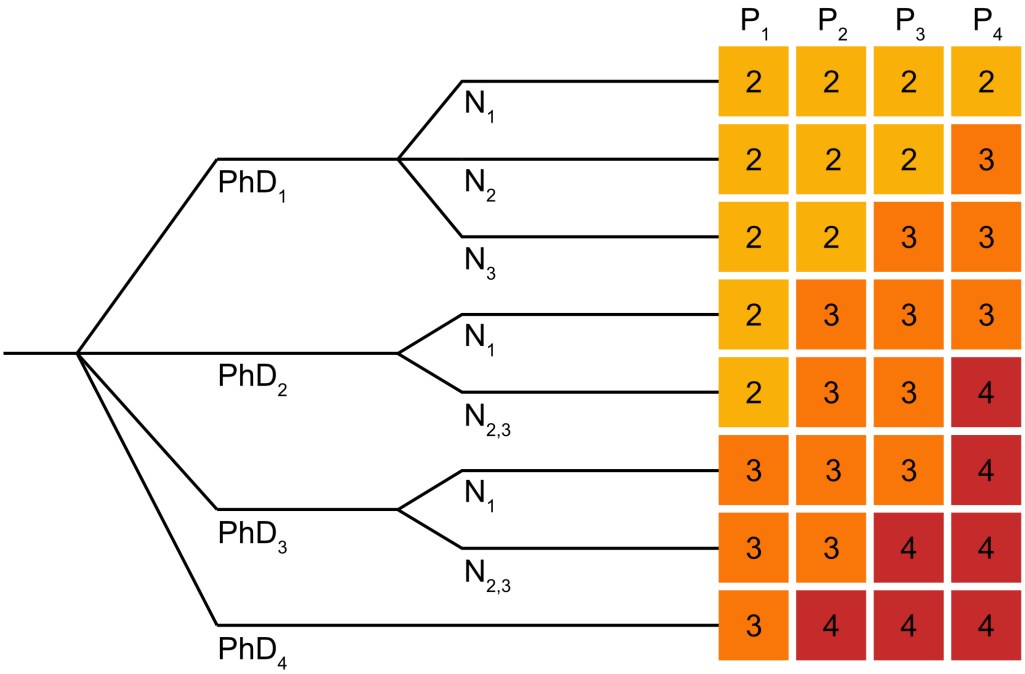


Figure 6.2: Example risk graph for AI-based systems that may result in physical damage for persons. Relevant aspects in this example are the physical damage (PhD, e.g., 1 = very small damage and 4 = life-threatening), the number of persons affected (N, e.g., 1 = few and 3 = many), and the probability of occurrence (P, e.g., 1 = unlikely and 4 = very likely). The suggested paths and values are only supposed to promote the general idea. In practical application, measurable values need to be specified.

Another consideration is to make the details of a risk-based regulation dependent on specific sectors. For example, risks in the context of medical products should potentially be assessed differently than risks in the insurance industry.[9] Such a risk-based approach to regulating AI-based systems is also proposed by the AI Act.

[9] In the GOAL project (see Section 1.2.1), media/advertising/social networks, labor market/performance evaluation/employment, medicine/health, education, lending/finance, military, law, mobility/transport, industrial automation, data protection, facial recognition in the private sector, product safety, sustainability, tourism, energy, migration, and state vs. private applications were internally discussed as possible sectors.

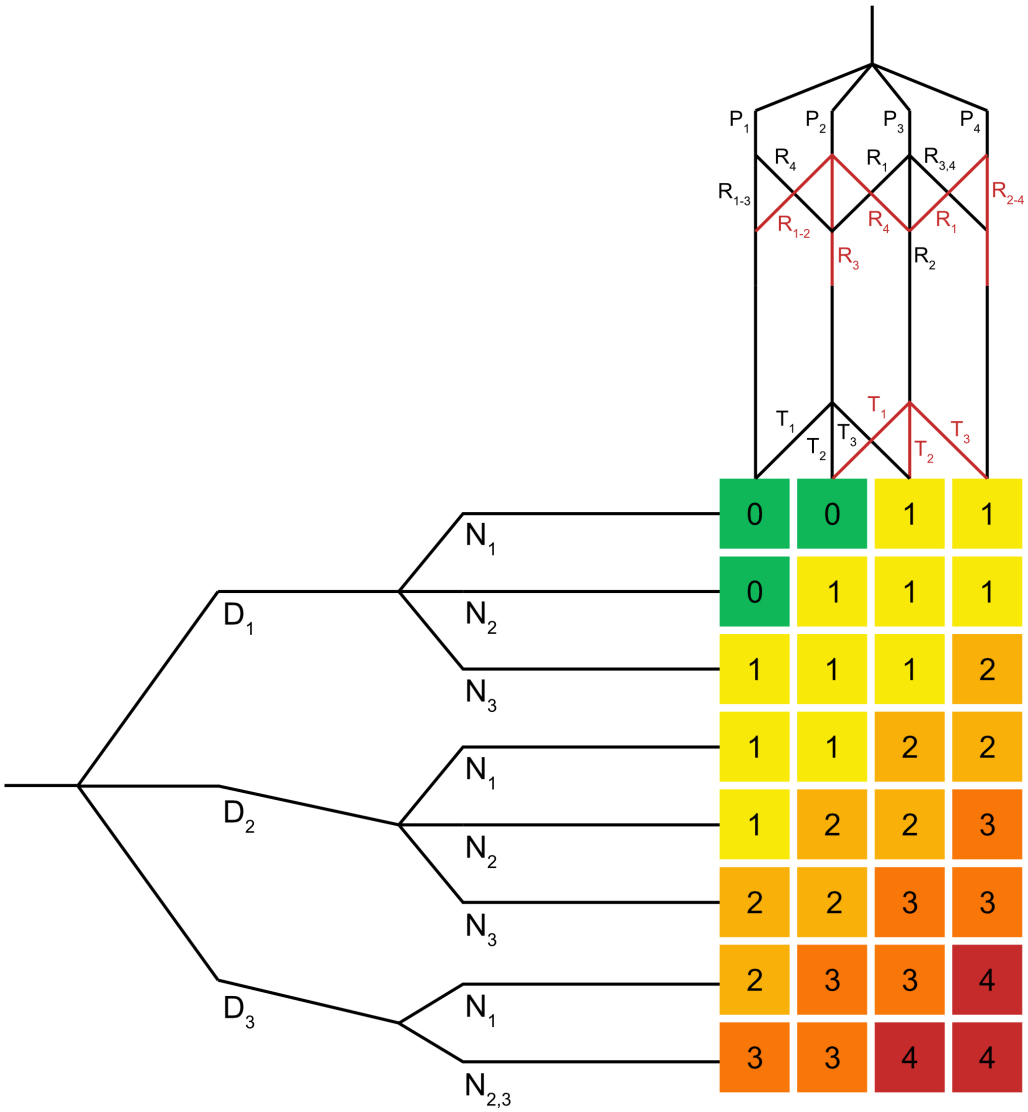


Figure 6.3: Example risk graph for AI-based systems that may only result in non-physical damage. Relevant aspects in this example are the non-physical damage, like environmental or financial damage (D, e.g., 1 = none and 3 = high), the number of persons affected (N, e.g., 1 = few and 3 = many), the probability of occurrence (P, e.g., 1 = unlikely and 4 = very likely), the probability of remedy (R, e.g., 1 = very high and 4 = very low), and the time to remedy (T, e.g., 1 = soon and 3 = late). The suggested paths and values are only supposed to promote the general idea. In practical application, measurable values need to be specified. The red lines are only meant to facilitate readability.

RQ 10

What are the challenges of a risk-based regulation approach for AI-based ADM systems, and which aspects are particularly relevant for assessing their risks?

Answer: A frequently used way to assess risk is to assess the consequences of an event as a number together with the associated likelihood of occurrence. The respective values are plotted on a two-dimensional matrix. Each area in the matrix represents a certain risk class. However, when it comes to assessing the risk of AI-based ADM systems, there is a lack of practical experience and scientific knowledge that could be used to thoroughly anticipate consequences and the likelihood of their occurrence. Furthermore, it has to be decided whether all possible consequences of an event are taken into account at once, or whether a risk assessment is necessary for each possible consequence.

In order to circumvent these challenges, there are various proposals for alternative parameters in risk assessment, including (but not limited to): affected parties (e.g., individual risks, risks for groups of people sharing specific characteristics, or risks for society as a whole), damage potential (both worst-case scenarios and a collection of all conceivable risks), chance of consequences becoming known to the affected party, extent of information asymmetry, monopoly position of a system vs. choice for the user/affected party, correction in the event of error/damage, duration until correction, chance of asserting claims for correction, and whether there is a human in the loop.

Boundaries between risk classes cannot be strictly defined in general, as there needs to be room to judge according to the specific circumstances. Additionally, it could be necessary to provide sector-dependent risk classifications, which would allow taking application-specific contexts into consideration. However, it could also make sense to define at least some strict conditions for a risk classification. A risk graph, for example, could be a valid choice to define rules such as: *'If there is any risk that people are severely physically harmed, the system needs to be at least in risk class 3, independent of any other conditions'*. Further research is needed to assess the use of risk matrices and risk graphs for risk-based regulation of AI-based ADM systems.

6.1.2 AI Act

With the AI Act, the European Union (EU) is currently working on a significant regulatory tool for AI-based applications. It will be the first law in the world to regulate AI in all areas of life.[10] A first proposal for the regulation was published on April 21, 2021. It set in motion a worldwide feedback process that resulted in numerous suggestions for improvements. In the meantime, individual member states of the EU have developed their own proposals for adaptation, which have also been subject to criticism from the general public. As no final version of the regulation is available at the time of submitting this thesis, the following details refer to the initial

[10] According to the European Parliament: https://www.europarl.europa.eu/news/en/headlines/society/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence, last accessed on June 29, 2023.

proposal. Explicit explanations are given for particularly relevant changes that have already been adopted.

The regulation proposal follows a risk-based approach for the classification of AI-based systems. In the first version, three risk classes are mentioned explicitly: (i) prohibited AI-based systems (Art. 5 para. 1), (ii) high-risk AI-based systems (Art. 6 para. 1 and Art. 6 para. 2), and (iii) 'other AI-based systems', which can be described as AI-based systems with special needs for transparency (Art. 52). All AI-based systems not affected by this categorization thus fall into a fourth, implicit category, that of AI-based systems without a need for regulation. In later versions of the AI Act,[11], a fifth class is also named, that of 'general-purpose' AI-based systems.

In general, it is very difficult to estimate how a new regulation will affect the respective market. In order to get an idea of how the initial draft of the AI Act would affect the AI-based software product market, we[12] classified the collection of 760 AI-based software products and projects in Germany provided by the '*Platform Learning Systems*' (PLS)[13] based on the presented risk classes according to the criteria of the AI Act. At the same time, we proposed an evaluation concept for future versions of the regulation. The team conducting this analysis consisted of experts from the fields of law and computer science to ensure that both the legal background and the implications of a described AI-based product were adequately understood by all participants.

After several pre-processing steps (removal of, among other things, empty entries, duplicate entries, non-AI applications, abstract project descriptions, and consulting activities), 514 use cases remained that we could classify in agreement. Of these, 160 use cases fell into the high-risk category, 39 into the category with a need for transparency, and 315 into the category with no need for regulation (see Table 6.1). It is particularly noteworthy that 98 of the high-risk use cases would be regulated by Art. 6 para. 1. annex II, section A, No. 11 AI Act (proposal). This passage deals with medical devices, which are already subject to regulation.

Table 6.1: Classification of the 514 cases that could be classified in agreement.

Risk-class	514	(100%)
Prohibited	0	(0%)
High-risk	160	(31.13%)
Need for transparency	39	(7.59%)
Low-risk	315	(61.28%)

In 21 cases, we were unable to agree on a category despite extensive discussions, due to the imprecision in some wording of the AI Act. For example, the regulation refers to 'safety components' in the definition of

[11] Version Brussels, November 25, 2022, 14954/22 2021/0106(COD).

[12] In this Section, 'we' refers to the authors of Hauer, Krafft, Sesing-Wagenpfeil, Zweig, et al., 2023.

[13] https://www.plattform-lernende-systeme.de/ki-landkarte.html, last accessed on March 31, 2023.

high-risk systems according to Art. 6 para. 1 AI Act (proposal).[14] According to Art. 3 No. 14 AI Act (proposal), a *'safety component of a product or system means a component of a product or of a system which fulfills a safety function for that product or system or the failure or malfunctioning of which endangers the health and safety of persons or property'*. Whether a use case contains a safety component according to this definition could only be roughly estimated on the basis of the description alone. In six cases, we could not derive a final conclusion. Another example is the unclear term *interaction* used in Art. 52 para. 1 AI Act (proposal).[15] For another six use cases, we could not agree on whether or not they fall under that paragraph.

In this study, we showed that the majority of AI-based products will probably not be affected by the regulation and that there is, therefore, no reason to fear over-regulation. In addition, we identified a number of needs for clarification in the formulations and amendments to the first draft of the AI Act. Some of these refinements have been implemented in more recent versions, in particular the integration of general-purpose AI-based systems in Art. 4a– Art. 4c of the most recent draft of the AI Act.

RQ 11

How will the upcoming AI Act affect the AI landscape, and how can this be determined in advance?

Answer: The effect of the upcoming AI Act on the AI landscape cannot be predicted with certainty, but it can be methodically estimated. Based on the description of an existing or at least planned AI-based ADM system, it is possible to estimate with some accuracy into which risk category under the AI Act it would fall. Such an assessment can be carried out for a large number of descriptions of AI-based systems. The collection of assessments corresponds to the foreseeable effect on the current AI landscape, provided that the selection of these systems roughly corresponds to the different types of all AI-based systems and their expected market distribution.

Based on the descriptions of the AI-based systems listed on the PLS,

[14] *'Irrespective of whether an AI-based system is placed on the market or put into service independently from the products referred to in points (a) and (b), that AI-based system shall be considered high-risk where both of the following conditions are fulfilled:*
(a) the AI-based system is intended to be used as a safety component of a product, or is itself a product, covered by the Union harmonisation legislation listed in Annex II;
(b) the product whose safety component is the AI-based system, or the AI-based system itself as a product, is required to undergo a third-party conformity assessment with a view to the placing on the market or putting into service of that product pursuant to the Union harmonisation legislation listed in Annex II.'

[15] *'Providers shall ensure that AI-based systems intended to interact with natural persons are designed and developed in such a way that natural persons are informed that they are interacting with an AI-based system, unless this is obvious from the circumstances and the context of use. This obligation shall not apply to AI-based systems authorized by law to detect, prevent, investigate and prosecute criminal offences, unless those systems are available for the public to report a criminal offence.'*

our study indicates that about 30% of all AI-based systems will be classified as high-risk, about 8% as systems with special transparency requirements, and about 62% as low-risk systems (Hauer, Krafft, Sesing-Wagenpfeil, Zweig, et al., 2023). However, these numbers should be treated with caution, as the influence of various sources of error can only be estimated to a limited extent.

The study presented here offers a methodology not only for assessing the impact of the AI Act, but also the potential impact of any regulatory proposal on the AI landscape. It is therefore also suitable for analyzing different forms of risk-based regulations, also in comparison with each other.

6.2 Corporate Digital Responsibility

As already mentioned in Section 3.4, it is increasingly perceived as a competitive advantage if companies accommodate the non-functional and non-statutory requirements and wishes of various forums. The hypothesis is that the power imbalance between actors and forums is still too great despite statutory regulation. However, if an actor gives the forums more power through voluntary means, for example, through public commitments, non-compliance with which would lead to reputational and thus financial damage, the power imbalance is at least reduced. These means are being discussed under the term *Corporate Digital Responsibility* (CDR) (Mueller, 2022). It originates from the close relationship with *Corporate Social Responsibility* (CSR) (C. Mihale-Wilson et al., 2022). Both concepts deal with ideological requirements for companies from a social point of view, without standing in the way of their economic growth or, in the best case, even facilitating it. The main difference is the explicit focus of CDR on today's novel challenges posed by digital products and services, which justifies its own separate exploration (Lobschat et al., 2021). In short, the *International CDR Manifesto*[16] defines CDR as *'a set of practices and behaviors that help an organization use data and digital technologies in ways that are perceived as socially, economically, and environmentally responsible'*.[17] It has to be noted that CDR, as a form of voluntary self-regulation, should only cover optional aspects. At least all minimum societal requirements should be covered by legislation or other mandatory forms of regulation, independent of CDR efforts. As societal requirements covered by law are country-specific, CDR implementation strategies might also need to be developed in a country-specific way.

[16] https://corporatedigitalresponsibility.net/cdr-manifesto-english, last accessed on March 11, 2023.

[17] Herden et al., 2021 define CDR as *'an extension of a firm's responsibilities which takes into account the ethical opportunities and challenges of digitalization'*, based on a collection of definitions in the scientific literature. This definition is more abstract and thus less useful for this thesis. However, I personally like the explicit relationship to ethics. The definition is fully compatible with the definition of the CDR manifesto.

The idea of discussing CDR on its own is relatively new.[18] Accordingly, there are still only a few actors active in this discipline, little academic literature exists on the subject, and only a few organizations are trying to bring CDR into the consciousness of companies. Furthermore, most publications deal with abstract considerations and goals. While this creates a broad basis for future work, there is still a lack of concrete implementation strategies. Developing these is indeed a great challenge, as very different CDR measures make sense for the wide range of very different digital products and services.

A very common implementation strategy that could be considered as CDR is the development of codes of conduct/ethics and the self-commitment to adhere to them. Many large companies and organizations have published their own codes of ethics in recent years. Often, however, the content of these codes is limited to the minimum legal requirements or describes very abstract goals, just as CDR does. There is a general concern that such codes are mechanisms that mainly serve to improve the image of a company without having any real effect (so-called *ethics-washing*, see Definition 21).

A different approach is the introduction of a label or seal. The idea behind this is to commission a third party to audit a company, product, or service for CDR-relevant aspects (in the sense of the ISO definition of audit; see Section 4.1.1). The prerequisite is that the auditing company is sufficiently qualified to carry out such an audit and that this qualification is also recognized by the market. Otherwise, there is a risk that the label or seal will not be acknowledged as meaningful by the rest of the market. Therefore, such labels and seals are a non-accredited form of certification (see Section 4.1.1). Paeffgen and Perdrisat, 2021, Chapter 3, present an overview of initiatives dealing with ethical challenges. All of them either offer some form of ethics label or aim to provide the basis for such a label.

Elliott et al., 2021, p. 185, propose establishing a digital responsibility code in the context of AI/FinTech-enabled financial services that entails privacy-protecting open data practices providing more inspectability for non-statutory forums. They also suggest that companies should share their CDR experiences with each other in order to benefit from each other's progress and experience, and to accelerate the development of the societal added value of CDR. In Germany, there are already several organizations and initiatives that put this idea into practice, for example, the CDR Lab[19] led by the consulting agency *dimension2*, the annual CDR Awards[20] organized by *Bayern Innovativ* and the *Bundesverband Digitale Wirtschaft (BVDW) e.V.*, and the CDR initiative[21] of the *Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection* (BMUV).

[18] Thorun, 2018 appears to be the first who used the term CDR in a scientific publication, but he refers to other scientific publications discussing the idea. The oldest mention goes back to 2015 (Adi et al., 2015). Cooper et al., 2015 appear to be the first who used the term CDR in general.

[19] https://www.cdr-lab.de/, last accessed on March 30, 2023.

[20] https://www.cdr-award.digital/ueber-uns/, last accessed on March 30, 2023.

[21] https://cdr-initiative.de/, last accessed on March 30, 2023.

Definition 21 (Ethics Washing)

The term *ethics washing* is an allusion to the term *greenwashing*, which is used to describe corporate efforts to pretend to act in an ecologically responsible manner without establishing any effective mechanisms. By analogy, ethics washing is about only pretending to act ethically or introducing ineffective measures and thereby appearing ethically responsible in the process (Wagner, 2018).

No matter what implementation strategies are pursued, CDR is always an investment first. Its implementation potentially costs a lot of time and resources. It can also entail risks, for example, if the release of information as part of a CDR process reveals that a company is not properly living up to its responsibilities. But CDR also comes with economic advantages, as it creates a unique selling point. Meeting the needs of users and stakeholders beyond functionality and law is increasingly seen as a competitive advantage. Consumers are willing to pay more money to support ethical enterprises and thus cover at least part of the extra costs (Carl et al., 2023). To fully utilize this added value, it is also important to communicate CDR well. Optimally, this is done in a uniform way, which is why some kind of operationalization of CDR is needed. In the last few years, more and more proposals have emerged for such operationalization (e.g., the CDR Building Bloxx[22] introduced by the Bundesverband Digitale Wirtschaft (BVDW) e.V.).

Herden et al., 2021 suggest and elaborate 20 CDR topics categorized into environmental, social, and governmental (ESG)[23] concerns. Thorun et al., 2017, Chapter 3, propose eight dimensions that are relevant to consider when thinking about CDR: (i) access, (ii) economic interests, (iii) product safety and liability, (iv) privacy and data security, (v) information and transparency, (vi) education and awareness, (vii) dispute resolution and redress, and (viii) governance and participation. Based on these, other researchers propose extensions or adjustments and elaborate subdimensions for further concretization and specific implementation strategies (e.g., Carl, 2021; C. A. Mihale-Wilson et al., 2021). Such fine-grained sub-dimensions also allow for a checklist-based approach as an implementation strategy, similar to the default data science ethics checklist proposed by the command line tool deon.[24] Currently, we[25] are working on a tool to measure the CDR level in the different dimensions based on

[22] https://www.cdr-building-bloxx.com/, last accessed on June 29, 2023.

[23] The UN Environment Programme – Finance Initiative, 2004, seems to be the first that used the term ESG and built a framework around it. Since then, it has often been used to help stakeholders understand how an organization is managing risks and opportunities related to environmental, social, and governance criteria. The term does not appear to be used by the scientific community.

[24] https://deon.drivendata.org/, last accessed on May 23, 2023.

[25] Here, 'we' refers to Kim Valerie Carl, Dr. Thomas Arnold, and me.

a set of checklists for each of these sub-dimensions. With such a tool, consumers indirectly become forums in an accountability process. Actors who do not comply with CDR principles potentially lose customers to companies that do comply with these principles. They thus create the consequences that are necessary for an accountability process.

Despite everything, CDR is supposed to be voluntary. A company does not necessarily have the time or the resources to comply with the principles or even put trade secrets at risk. However, advertising with CDR creates an indirect obligation because if consumers discover that CDR aspects are being advertised but not fulfilled, a loss of image, and thus economic damage, is to be expected. Therefore, the idea of CDR has the potential to shift the power structure between companies and consumers a little more toward consumers. It also allows consumers to be seen as forums in an accountability process.

RQ 12

What role does the increasingly recognized concept of Corporate Digital Responsibility (CDR) play?

Answer: Corporate Digital Responsibility (CDR) promotes the idea of establishing practices and behaviors in the context of developing and using digital technologies that are perceived as socially, economically, and environmentally responsible, beyond what is required by law. Apart from a good conscience, the added value for companies lies in a unique selling point with regard to ethical accountability. Currently, only a few concrete implementation strategies exist, but the number of scientists and company representatives working on this topic is growing.

One of the biggest challenges is to capture the added business value of CDR through effective customer communication. In recent years, the scientific literature has elaborated sophisticated dimensions that are relevant for CDR. Based on these dimensions (and sub-dimensions that go into further detail), a tool is currently under development that measures the CDR level in the different dimensions based on a set of questions for each of these sub-dimensions. How well this tool is accepted and how useful it is needs to be investigated after its release.

Chapter 7

Bringing it All Together

The large number of different types of applications and application contexts for AI-based ADM systems is accompanied by a large number of different needs and options regarding their accountability. A one-size-fits-all solution cannot suffice. Therefore, the aim of this thesis was to collect, process, and extend the possibilities of promoting accountability in software development processes and software systems in order to support the achievement of the European goal of 'trustworthy AI'.

The basis is an understanding of accountability, which, first of all, requires that a system can be analyzed. In Chapter 3, a generic software development process is divided into several sections, based on the long chain of responsibilities. Each section is examined separately in terms of what information can be disclosed for external evaluation (transparency mechanisms), and what accesses to the system (or its partial components) can be granted for analysis purposes (inspectability mechanisms, see RQ 2, p. 72). The chapter also introduced John Austin's Speech Act theory as a thought experiment to determine which actors are to be held accountable (see RQ 3, p. 79)

Based on the information disclosed or retrievable through external access, various auditing procedures can be carried out (see Chapter 4). There are different understandings of the term audit. Two of them are particularly relevant for this thesis: regulatory inspections based on standardization – for example, the body of ISO standardization documents that form the basis for accredited certification – and bias audits of platforms (see RQ 4, p. 92). However, the significance of an audit heavily depends on the trustworthiness of the auditing personnel and the audit process being carried out. This trustworthiness, in turn, can be measured with certificates.

The audit understanding of Sandvig et al., 2014, assumes that a trusted party (e.g., an NGO or oneself) carries out a 3^{rd} party audit, but without the legitimation of the company whose product is being audited. This means that the auditing procedures to be considered depend greatly on the information accessible to a forum. This refers not only to the information about the system itself but also, for example, to the question of whether the system works with information of which the forum is unaware. In many cases, only limited transparency and inspectability mechanisms are available to the forum, which, in turn, restricts the choice of feasible analysis approaches, potentially to a huge extent.

Regardless of what type of audit is being carried out, the question

arises as to what exactly the expectations are that the system being audited should meet. In order to answer this question, taking into account all relevant stakeholder groups (including users and those affected), an Assurance Case can be developed (see RQ 5, p. 102). An Assurance Case is a reasoned argument supported by a body of evidence, which states that a system operates as intended for a defined application in a defined environment. In order to gain evidence from published information or system access, concrete, structured analytical procedures are needed, i.e., tests.

The term *testing* is very broad and not uniformly defined (see Chapter 5). Some subjects have multiple labels, and many terms are discussed under multiple meanings. As a result, communication around testing activities can be difficult and misunderstandings may not be recognized until long after deployment (or not at all). To address this problem, test-related terms in the context of testing data-driven components can be divided into different levels of abstraction and related to each other (see RQ 6, p. 111 and RQ 7, p. 138). In addition, data-driven components and AI-based applications bring new challenges that cannot be fully addressed with traditional testing approaches. This is particularly the case in the context of fairness testing, which is why special attention is given to this type of testing in this thesis and to how it can be applied in the context of the audit procedures described by Sandvig et al., 2014 (see RQ 1, p. 42 and RQ 8, p. 126). At the end of that chapter, it is discussed how the development, execution, and fulfillment of test requirements can be implemented within agile test development processes and how Acceptance Test-Driven Development can be extended by implementing the Assurance Case framework (see RQ 9, p. 139).

All approaches for promoting accountability require sufficient incentives to implement them. Regulations provide extrinsic incentives in the form of legal requirements that must be adhered to under threat of consequences. However, traditional regulatory approaches are relatively inflexible and accompanied by a lengthy process when it comes to adapting them to new regulatory needs. As a result, they are hardly able to keep up with the rapid developments in the software industry, especially in the context of AI-based applications. Risk-based regulation is intended to provide faster response times and better adaptability to changing regulatory needs as a result of new developments (see RQ 10, p. 147). The AI Act is a first large-scale implementation of this idea. Corporate Digital Responsibility is a novel approach to fostering intrinsic incentives; on the one hand, by calling on the good in people, and, on the other hand, by recognizing the ever-increasing importance of benevolence as a competitive advantage.

For both approaches, there are only indicators for assessing their effectiveness so far (see RQ 11, p. 150 and RQ 12, p. 154). However, as the AI Act is about to become legally binding, forcing the EU member states to develop national implementations, and as CDR has gained significant importance and attention in recent years, analyses of their effectiveness will be possible in the foreseeable future.

Taking all these approaches and considerations together, a large toolbox of options emerges that can be applied to a wide range of possible

products, services, and situations in order to establish accountability of AI-based ADM systems. Furthermore, this thesis can serve as a framework for linking all these aspects together and as an incentive for examining synergies more thoroughly. In the process, it also shows the limitations of current approaches and presents further research needs.

7.1 Limitations and Future Work

The answers to many of the research questions pursued in this thesis point to further research needs:

- Section 2.3.3.2 introduces a hierarchical view of fairness that helps to resolve conflicts between group fairness and individual fairness. How useful this approach is for evaluating the fairness of DDCs or as a secondary goal for training DDCs, and what new challenges it brings, for example, in defining appropriate subgroups, needs to be evaluated in practical trials (see RQ 1).

- This thesis elaborates on a broad selection of transparency and inspectability mechanisms based on a generic software development process model. The considerations are supported by discussions with other experts in the field, various use cases, and relevant technical literature (see RQ 2). Nevertheless, other important mechanisms for specific use cases are potentially missing. Therefore, an extension on the basis of practical trials would make sense. In addition, it might be a good idea to build up a database of general and specific concerns from various stakeholders, similar to the AI incidence database[1]. In the future, this could provide a multi-stakeholder perspective on all possible aspects for which someone should be accountable.

- In Section 3.2, the considerations regarding how to determine which actors are to be held accountable based on Austin's Speech Act theory lead to the idea of making the developing company accountable for everything concerning the system it develops and deploys. On this basis, the company could then contractually hand over accountability for specific aspects to other actors who also contribute to the substitution of the speech act. Whether this consideration would lead to the market regulating which actor is to be held accountable based on a balance of trust, control, and costs, whether such an approach would lead to individual actors being manipulated into taking over accountability for aspects for which they do not want to (or should not) be accountable, or whether other undesirable side effects would arise, needs to be investigated in practical trials (see RQ 3).

- In the context of this thesis, our adaptation of the Assurance Case framework for extra-functional requirements, more precisely fairness, could be tested in the continuous development process of a real industrial software product (see Section 4.2). In addition, a second trial is currently underway, which cannot be discussed in more detail for contractual reasons. The experiences made so far show that

[1] https://incidentdatabase.ai/, last accessed on September 5, 2023.

real-life situations are much more complex than could be covered by theoretical considerations. Although the results are promising and have been perceived as added value by the clients in both cases so far, there is still a lot of room for improvement, especially regarding the necessary effort. Research in the context of compatible software development processes such as Accepted Test-Driven Development (see RQ 9) seems to be particularly relevant for this. Also, a collection of generic Assurance Cases for specific kinds of applications could be developed that could be used as strong templates for future Assurance Case development processes regarding similar applications. Furthermore, the framework has primarily been used for requirements engineering and continuous development. Its use for testing compliance requirements, for communicating with stakeholders not involved in the process, and as a basis for auditing and certification processes cannot yet be assessed without further research (see RQ 5). For this, further industry trials and experience-based adjustments to the process are needed.

- Discussions with software testing experts from the Fraunhofer Institute for Experimental Software Engineering IESE, DIN, ISO, ISTQB, and various companies have shown that a lot of miscommunication takes place around test-related terms, especially in the context of testing AI-based applications (see RQ 7). The standardization of such terms has only been successful to a limited extent, as different standardization bodies sometimes provide different definitions and these are also not always compatible with the already inconsistent technical literature. It cannot be ruled out that the differences between the various definitions and understandings will be resolved sooner or later, but it is questionable whether this should be the goal of further research, as the challenge does not lie in a lack of knowledge or understanding. Nevertheless, scientists can contribute to identifying, detailing, and communicating the discrepancies. The AI Glossary[2] tries to contribute to this; however, it needs to be continuously maintained, expanded, and revised.

- The AI Act is a combination of a risk-based and a rights-based regulatory approach (see Section 6.1). An alternative approach would be to build a risk graph that could implement a risk-based regulatory approach within rights-based boundaries (see RQ 10). This approach could be interesting for the national implementation strategy of the AI Act, but it would first need to be assessed in terms of suitability for this task.

- It should be noted that the suitability of the AI Act itself can only be conclusively assessed through its implementation and then only after some time based on its impact (see RQ 11). At the same time, the approach we present in Section 6.1.2 provides a systematic procedure for a rough pre-assessment of the impact of a new regulation on an entire product family, which is also capable of assessing the suitability of other regulatory efforts still being negotiated and revised.

[2] https://ai-glossary.org/index.php?l=en, last accessed on June 29, 2023.

- The CDR movement as an approach to promoting intrinsic motivation for acting "good" or "benevolently" is generally to be welcomed, but the movement is still in its infancy. A better elaboration of concrete implementation strategies and possibilities of also communicating the efforts undertaken to customers in order to generate a unique selling point that can be perceived by the market is required (see RQ 12).

All considerations in this thesis are based on procedural regularity. This means that all data points are processed by the same system according to the same rules. In contrast, hard-coded behavior for a certain kind of information could be to assign a fixed output to the corresponding data point. Such cases can hardly be detected with dynamic testing and auditing concepts, i.e., everything except code reviews. Since manual code reviews are only possible to a limited extent beyond a certain code size and complexity, static code analysis methods might be suitable for recognizing such cases automatically in the source code. However, this requires further research, ideally on the basis of the source code of real AI-based ADM systems.

7.2 Final Remarks

In the scientific literature on algorithmic accountability (and by this, I also implicitly mean most of my own publications), most considerations are of a theoretical nature or only practically validated under model assumptions and ideal conditions. As researchers, we make every effort to think of all relevant facets of a problem. At the same time, we try to create conditions for practical validations that allow similar research approaches to be compared with each other. However, these conditions are rarely found in practice. In this respect, the results of many practical validations are only helpful to a limited extent. I have experienced this problem from the perspective of industry as well as from the perspective of research. As developers, we experiment a lot with different test formats, and all too often, ideas are discarded from the beginning by superiors because the experience has already been made that they *'only work that well in theory'*. Conversely, there are hardly any opportunities for most academic scientists to test their own research in commercial practice. In the few cases where I got such a valuable opportunity (for example, in the context of applying the Assurance Case framework to the development process of a piece of software that automates clinical rotation schedules; see Section 4.2), the applicability of the theoretical considerations turned out to be more complex than originally assumed. In this specific case, however, it fortunately turned out that the framework is flexible enough to deal with processes in a commercial enterprise that are not necessarily model-like or linear. That is why I conclude this thesis with a personal appeal: We need more field testing of independent research results on commercial applications and thus more cooperation between universities, research institutes, and companies, to create a foundation that is as solid as possible to achieve the European goal of *trustworthy AI*.

Scientific Publications in the Context of this Thesis

Haeri Amir, M., Hauer, M. P., & Zweig, K. (2023). Equality of quality: The relation between quality measures and fairness measures for evaluating machine learning models.
Author's contribution: All authors contributed approximately equally to this publication. The paper is still under review.

Hallensleben, S., Hustedt, C., Fetic, L., Fleischer, T., Grünke, P., Hagendorff, T., Hauer, M. P., Hauschke, A., Heesen, J., Herrmann, M., Hillerbrand, R., Hubig, C., Kaminski, A., Krafft, T. D., Loh, W., Otto, P., & Puntschuh, M. (2020). From principles to practice - an interdisciplinary framework to operationalise ai ethics. *iRights. Lab, Tech. Rep.* https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/WKIO_2020_final.pdf.
Author's contribution: Tobias Krafft and I wrote Chapter 3 with equal participation. In terms of content, my own novel contribution primarily involved dividing the dimensions of the risk matrix into individual sub-aspects. My collection of AI-based systems that had negative effects on individuals, groups of people, or society as a whole due to errors or unintended side effects served as the basis for the considerations.

Hauer, M. P., Adler, R., & Zweig, K. (2021). Assuring fairness of algorithmic decision making. *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 110–113.
Author's contribution: Rasmus Adler and I developed the idea to extend the Assurance Case framework from safety engineering with Acceptance Test-Driven Development from agile software development methodologies to make it suitable for arguing extra-functional properties of AI-based systems together. I was responsible for the first draft, communication and organization.

Hauer, M. P., Hofmann, X. C., Krafft, T. D., Zweig, K. A., et al. (2020). Quantitative analysis of automatic performance evaluation systems based on the h-index. *Scientometrics*, *123*(2), 735–751.
Author's contribution: This paper is based on my Master's thesis and Xavier Hofmann's Bachelor's thesis. I made the main contribution to the writing and composition of the paper. I was responsible for communication and organization.

Hauer, M. P., Kevekordes, J., & Haeri, M. A. (2021). Legal perspective on possible fairness measures–A legal discussion using the example of hiring decisions. *Computer Law & Security Review*, *42*, 105583.
Author's contribution: I had the idea for this paper and moderated the interdisciplinary discussions between the legal perspective (Johannes Kevekordes) and the computer science perspective

(Maryam Amir Haeri). I was responsible for providing a first draft, communication, organization, and quality assurance.

Hauer, M. P., Krafft, T. D., Sesing-Wagenpfeil, D., Zweig, P., et al. (2023). Quantitative study about the estimated impact of the ai act. *arXiv preprint arXiv:2304.06503*.

Author's contribution: Around half of the publication is based on my written contribution. However, the main effort clearly was the study itself, to which I contributed around 30%. A journal submission is currently being processed.

Hauer, M. P., Krafft, T. D., & Zweig, K. (2023). Overview of transparency and inspectability mechanisms to achieve accountability of ai systems. *Data and Policy*.

Author's contribution: The paper is a product of close collaboration with Tobias D. Krafft, on which we worked together closely and made significant contributions that bridge our respective research fields. Our contributions to the paper share many similarities in terms of their impact and importance. The paper has already been accepted but is still under review.

Hauer, M. P., Müller-Kress, L., Leimüller, G., & Zweig, K. (2023). Using Assurance Cases to assure the fulfillment of non-functional requirements of AI-based systems - Lessons learned. *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 172–179.

Author's contribution: This publication expands on the ideas discussed in Hauer, Adler, and Zweig, 2021 and tests them in industrial practice. I had the scientific lead of the Assurance Case development process. The written elaboration of the paper predominantly draws from my work and contributions.

Hoffmann, H., Vogt, V., Hauer, M. P., & Zweig, K. (2022a). Fairness by awareness? on the inclusion of protected features in algorithmic decisions. *Computer Law & Security Review*, *44*, 105658.

Author's contribution: This is basically an English translation of Hoffmann, Vogt, Hauer, and Zweig, 2022b. I was responsible for providing the first translation.

Hoffmann, H., Vogt, V., Hauer, M. P., & Zweig, K. (2022b). Fairness by awareness? Zur Einbeziehung geschützter Merkmale in algorithmische Entscheidungen. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 191–220.

Author's contribution: Katharina Zweig and I were responsible for the technical elaborations, Hanna Hoffmann and Verena Vogt for the legal elaborations. We each ensured that the explanations of the other specialist group are comprehensible to experts of our own discipline. I was responsible for communication and organization.

Jöckel, L., Bauer, T., Kläs, M., Hauer, M. P., & Groß, J. (2021). Towards a common testing terminology for software engineering and data science experts. *International Conference on Product-Focused Software Process Improvement*, 281–289.

Author's contribution: All authors contributed approximately equally to this publication.

Kevekordes, J., Hauer, M. P., & Haeri, M. A. (2022). Rechtliche bewertung möglicher fairnessmaße. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 141–190.

Author's contribution: This is basically a German translation of Hauer, Kevekordes, and Haeri, 2021 with a stronger focus on the legal discussion. I was responsible for providing the first translation.

Krafft, T. D., Hauer, M. P., & Zweig, K. (2023). Black box testing and auditing of bias in ADM systems. *Minds and Machines*.

Author's contribution: The paper is the outcome of a close collaboration between Tobias Krafft and me. Throughout the process, we worked together and made substantial contributions that effectively connect our respective research fields. Our individual contributions to the paper exhibit significant similarities in terms of their impact and importance. The paper is still under review.

Krafft, T. D., Hauer, M. P., & Zweig, K. A. (2020). Why do we need to be bots? what prevents society from detecting biases in recommendation systems. *International Workshop on Algorithmic Bias in Search and Recommendation*, 27–34.

Author's contribution: This paper is based on a joint project for which Tobias Krafft was more responsible for the research and I was more responsible for the practical application. Around 40% of the publication are based on my written contribution.

Further Publications in the Context of this Thesis

Adler, R., Becker, N., Borges, G., Hauer, M. P., Heidrich, J., Hilpitsch, S., Hoffmann, R., Junginger, P., Jöckel, L., Kläs, M., Krupka, D., Martinez, L., Sesing, A., & Zweig, K. (2021). Abschlussbericht examai – KI testing und auditing. herausforderungen, lösungsansätze und handlungsempfehlungen für das testen, auditieren und zertifizieren von ki. *ExamAI – KI Testing & Auditing, Gesellschaft für Informatik e.V.*

Author's contribution: In the ExamAI project, I was mainly responsible for the application area of AI in HR processes and co-responsible for the technical aspects of testing, auditing, and certification. I co-authored all text sections that address these topics.

DIN/DKE. (2020). German standardization roadmap on artificial intelligence 1.0. *DIN/DKE, Berlin/Frankfurt*.

Author's contribution: For this version, I mainly participated in the ethics working group.

DIN/DKE. (2023). German standardization roadmap on artificial intelligence 2.0. *DIN/DKE, Berlin/Frankfurt*.

Author's contribution: For this version, I mainly participated in the working groups Ethics and Glossary, contributed to the topics Explainability/Transparency and Fairness across all working groups, and supported the introduction of the Assurance Case framework. Since the glossary had to be shortened considerably, the glossary team decided to publish the full version separately (see http://www.ai-glossary.org/index.php). Note that the German version was already published in 2022.

Hauer, M. P., & Zweig, K. (2021). Chancen und risiken algorithmischer entscheidungen. *Human Ressource Manager*, *01/2021*, 48–53.

Author's contribution: I developed a first draft and refined it together with Katharina Zweig. I was responsible for communication and organization.

Kunze, L., Leimüller, G., Müller-Kress, L., & Hauer, M. P. (2023). Method handbook: Assurance cases for fair ai systems. https://www.fairbydesign.eu/s/Handbook-for-Assurance-Cases-for-fair-AI-systems.pdf, last accessed on 20.04.2023.

Author's contribution: I provided most of the content in preparation but was barely involved in writing it down and visualizing it as a handbook. However, I was responsible for ensuring that the wording was precise and correct.

Runze, G., Haimerl, M., Hauer, M. P., Holoyad, T., Obert, O., Pöhls, H., Tagiew, R., & Ziehn, J. (2023). Das AI-Glossary als Weg aus Babylon - Ein Werkzeug für eine gemeinsame KI-Terminologie. *JavaSpektrum*,

03/2023, 42–46.
Author's contribution: All authors contributed equally to this publication.

Zweig, K., Hauer, M. P., & Raudonat, F. (2020). Anwendungsszenarien: KI-systeme im personal- und talentmanagement. *ExamAI – KI Testing & Auditing, Gesellschaft für Informatik e.V.*
Author's contribution: All authors contributed approximately equally to this publication. I was responsible for communication and organization.

Further Scientific Publications by the Author not Related to this Thesis

Bernstein, A., de Vreese, C., Helberger, N., Schulz, W., Zweig, K., Baden, C., Beam, M. A., Hauer, M. P., Heitz, L., Jürgens, P., et al. (2020). Diversity in news recommendations. *arXiv preprint arXiv:2005.09495*.
Author's contribution: All authors contributed approximately equally to the first draft of this publication. The five supervisors (and first authors) refined and published it.

Groen, E. C., Kopczyńska, S., Hauer, M. P., Krafft, T. D., & Doerr, J. (2017). Users—the hidden software product quality experts?: A study on how app users report quality aspects in online reviews. *2017 IEEE 25th international requirements engineering conference (RE)*, 80–89.
Author's contribution: Tobias Krafft and I were equally responsible for pre-processing further test data and improving the regular expressions. I found a way to improve the recognition of negations, which improved prediction precision by around 10% (see Table 5). We both contributed smaller sections of text to the paper.

Zweig, K. A., Krafft, T. D., & Hauer, M. P. (2017). *Dein Algorithmus - meine Meinung! Algorithmen und ihre Bedeutung f́ur Meinungsbildung und Demokratie*.
Author's contribution: The brochure was written together with Katharina Zweig and Tobias D. Krafft and has been printed and downloaded over 20,000 times.

Complete Bibliography

Abraham, A. (2005). Artificial Neural Networks. *Handbook of measuring system design*.

Abràmoff, M. D., Lavin, P. T., Birch, M., Shah, N., & Folk, J. C. (2018). Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. *NPJ digital medicine*, *1*(1), 39.

Adi, A., Crowther, D., & Grigore, G. (2015). *Corporate social responsibility in the digital age*. Emerald Group Publishing.

Adler, R., Becker, N., Borges, G., Hauer, M. P., Heidrich, J., Hilpitsch, S., Hoffmann, R., Junginger, P., Jöckel, L., Kläs, M., Krupka, D., Martinez, L., Sesing, A., & Zweig, K. (2021). Abschlussbericht examai – KI testing und auditing. herausforderungen, lösungsansätze und handlungsempfehlungen für das testen, auditieren und zertifizieren von ki. *ExamAI – KI Testing & Auditing, Gesellschaft für Informatik e.V.*

Author's contribution: In the ExamAI project, I was mainly responsible for the application area of AI in HR processes and co-responsible for the technical aspects of testing, auditing, and certification. I co-authored all text sections that address these topics.

Agichtein, E., Brill, E., & Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 19–26.

Allhutter, D., Cech, F., Fischer, F., Grill, G., & Mager, A. (2020). Algorithmic profiling of job seekers in austria: How austerity politics are made effective. *Frontiers in big Data*, *3*, 5.

Ammann, P., & Offutt, J. (2017). *Introduction to software testing* (Vol. 2). Cambridge University Press.

Ananny, M., & Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, *20*(3), 973–989. https://doi.org/10.1177/1461444816676645

Andersson, J., Bache, G., & Sutton, P. (2003). Xp with acceptance-test driven development: A rewrite project for a resource optimization system. *International Conference on Extreme Programming and Agile Processes in Software Engineering*, 180–188.

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine Bias - There's software used across the country to predict future criminals. And it's biased against blacks. *ProPublica*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.

(2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, *58*, 82–115.

ASTM. (2022). Form and Style for ASTM Standards. https://www.astm.org/media/pdf/bluebook_FormStyle.pdf%20(last%20accessed%20on%2017.02.2023)

Austin, J. L. (1962). *How to do things with words*. Oxford University Press.

Baer, T. (2019). *Understand, manage, and prevent algorithmic bias: A guide for business users and data scientists*. Apress.

Barbalau, A., Cosma, A., Ionescu, R. T., & Popescu, M. (2020). A generic and model-agnostic exemplar synthetization framework for explainable ai. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 190–205.

Barocas, S., Hardt, M., & Narayanan, A. (2017). *Fairness and machine learning* (Vol. 1).

Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., & Yoo, S. (2015). The oracle problem in software testing: A survey. *IEEE transactions on software engineering*, *41*(5), 507–525.

Baum, T., Liskin, O., Niklas, K., & Schneider, K. (2016). A faceted classification scheme for change-based industrial code review processes. *2016 IEEE International conference on software quality, reliability and security (QRS)*, 74–85.

Bechavod, Y., & Ligett, K. (2018). Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*.

Beck, K. (2003). *Test-driven development: By example*. Addison-Wesley Professional.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2021). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, *50*(1), 3–44.

Bernstein, A., de Vreese, C., Helberger, N., Schulz, W., Zweig, K., Baden, C., Beam, M. A., Hauer, M. P., Heitz, L., Jürgens, P., et al. (2020). Diversity in news recommendations. *arXiv preprint arXiv:2005.09495*. **Author's contribution:** All authors contributed approximately equally to the first draft of this publication. The five supervisors (and first authors) refined and published it.

Bertrand, M., & Duflo, E. (2017). Chapter 8 - Field Experiments on Discrimination. In A. V. Banerjee & E. Duflo (Eds.), *Handbook of field experiments* (pp. 309–393, Vol. 1). North-Holland. https://doi.org/10.1016/bs.hefe.2016.08.004

Betka, M., & Wagner, S. (2021). Extreme mutation testing in practice: An industrial case study. *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*, 113–116.

Bharadiya, J. (2023). Artificial Intelligence in Transportation Systems A Critical Review. *American Journal of Computing and Engineering*, *6*(1), 34–45.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Bovens, M. (2007). Analysing and Assessing Accountability: A Conceptual Framework. *European law journal*, *13*(4), 447–468.

Braithwaite, V. (2020). Beyond the bubble that is Robodebt: How governments that lose integrity threaten democracy. *Australian Journal of Social Issues*, *55*(3), 242–259.

Brauneis, R., & Goodman, E. P. (2018). Algorithmic transparency for the smart city. *Yale JL & Tech.*, *20*, 103.

Brayne, S., & Christin, A. (2021). Technologies of crime prediction: The reception of algorithms in policing and criminal courts. *Social Problems*, *68*(3), 608–624.

Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. *2017 IEEE International Conference on Big Data (Big Data)*, 1123–1132.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical science*, *16*(3), 199–231.

Brendel, W., Rauber, J., & Bethge, M. (2018). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*.

Broniatowski, D. A., et al. (2021). Psychological foundations of explainability and interpretability in artificial intelligence. *NIST, Tech. Rep*.

Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., et al. (2020). Toward trustworthy AI development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. *Conference on fairness, accountability and transparency*, 77–91.

Busuioc, M. (2021). Accountable artificial intelligence: Holding algorithms to account. *Public Administration Review*, *81*(5), 825–836.

Calders, T., Kamiran, F., & Pechenizkiy, M. (2009). Building classifiers with independency constraints. *2009 IEEE international conference on data mining workshops*, 13–18.

Calders, T., & Verwer, S. (2010). Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, *21*(2), 277–292.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R. (2017). Optimized pre-processing for discrimination prevention. *Advances in Neural Information Processing Systems*, 3992–4001.

Calmon, F. d. P., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R. (2018). Data pre-processing for discrimination prevention: Information-theoretic optimization and analysis. *IEEE Journal of Selected Topics in Signal Processing*, *12*(5), 1106–1119.

Camp, L. J. (2006). Varieties of software and their implications for effective democratic government. *Proceedings of the British Academy*, *135*, 183–185.

Campbell, I. C. (2021). The Apple Card doesn't actually discriminate against women, investigators say. *The Verge (Online)*, *23*. https://www.theverge.com/2021/3/23/22347127/goldman-sachs-apple-card-no-gender-discrimination, last accessed on 27.07.2023.

Carl, K. V. (2021). Corporate digital responsibility: Evaluating privacy and data security activities on company-level. *INFORMATIK 2021*.

Carl, K. V., Mihale-Wilson, C., Zibuschka, J., & Hinz, O. (2023). A consumer perspective on corporate digital responsibility: An empirical evaluation of consumer preferences. *Journal of Business Economics*, 1–46.

Chang, J. C., Amershi, S., & Kamar, E. (2017). Revolt: Collaborative crowdsourcing for labeling machine learning datasets. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2334–2346.

Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *Ieee Access*, *8*, 75264–75278.

Chen, T. Y., Tse, T. H., & Zhou, Z. Q. (2003). Fault-based testing without the need of oracles. *Information and Software Technology*, *45*(1), 1–9.

Chen, T., Cheung, S., & Yiu, S. (1998). Metamorphic testing: A new approach for generating next test cases. technical report hkust-cs98-01. *Hong Kong Univ. of Science and Technology*.

Cheng, C.-H., Huang, C.-H., Ruess, H., Yasuoka, H., et al. (2018). Towards dependability metrics for neural networks. *2018 16th ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, 1–4.

Chettiar, I. M., & Gupta, V. (2011). Smart reform is possible: States reducing incarceration rates and costs while protecting communities. *Available at SSRN 1934415*.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*, 1–13.

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

Citron, D. K., & Pasquale, F. (2014). The scored society: Due process for automated predictions. *Wash. L. Rev.*, *89*, 1.

Coeckelbergh, M. (2020). Artificial intelligence, responsibility attribution, and a relational justification of explainability. *Science and engineering ethics*, *26*(4), 2051–2068.

Cofone, I. N., & Strandburg, K. J. (2019). Strategic Games and Algorithmic Secrecy. *McGill LJ*, *64*, 623–663.

Cohen, D. M., Dalal, S. R., Parelius, J., & Patton, G. C. (1996). The combinatorial design approach to automatic test generation. *IEEE software*, *13*(5), 83–88.

Cooper, T., Siu, J., & Wei, K. (2015). Corporate digital responsibility: Doing well by doing good. *Outlook (Accenture)*.

Cortes, C., Jackel, L. D., Chiang, W.-P., et al. (1995). Limits on learning machine accuracy imposed by data quality. *KDD*, *95*, 57–62.

Cortez, P., & Embrechts, M. J. (2011). Opening Black Box Data Mining Models Using Sensitivity Analysis. *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 341–348.

Cosier, C. (2017). How Centrelink's 'robodebt' ran off the rails. *Australian Broadcasting Corporation, Radio National*. https://www.abc.net.au/radionational/programs/backgroundbriefing/8319442, last accessed on 27.07.2023.

Craven, M., & Shavlik, J. (1995). Extracting Tree-Structured Representations of Trained Networks. *Advances in neural information processing systems*, *8*.

Crawford, K. (2016). Can an algorithm be agonistic? Ten scenes from life in calculated publics. *Science, Technology, & Human Values*, *41*(1), 77–92.

Cruz-Benito, J., Vázquez-Ingelmo, A., Sánchez-Prieto, J. C., Therón, R., García-Peñalvo, F. J., & Martín-González, M. (2017). Enabling adaptability in web forms based on user characteristics detection through a/b testing and machine learning. *IEEE Access*, *6*, 2251–2265.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. (2022). Underspecification presents challenges for credibility in modern machine learning. *The Journal of Machine Learning Research*, *23*(1), 10237–10297.

Danks, D., & London, A. J. (2017). Algorithmic Bias in Autonomous Systems. *IJCAI*, *17*, 4691–4697.

Dastin, J. (2018). 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-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G, last accessed on 27.07.2023.

Datenethikkommission. (2019). Gutachten der Datenethikkommission der Bundesregierung. https://www.bmi.bund.de/SharedDocs/downloads/DE/publikationen/themen/it-digitalpolitik/gutachten-datenethikkommission.pdf, last accessed on 27.07.2023.

Datta, A., Tschantz, M. C., & Datta, A. (2015). Automated Experiments on Ad Privacy Settings. *Proc. Priv. Enhancing Technol.*, *2015*(1), 92–112.

Dawson, D., Schleiger, E., Horton, J., McLaughlin, J., Robinson, C., Quezada, G., Scowcroft, J., & Hajkowicz, S. (2019). Artificial intelligence: Australia's ethics framework-a discussion paper.

De Laat, P. B. (2018). Algorithmic decision-making based on machine learning from Big Data: Can transparency restore accountability? *Philosophy & technology*, *31*(4), 525–541.

Delgado-Rodriguez, M., & Llorca, J. (2004). Bias. *Journal of Epidemiology & Community Health*, *58*(8), 635–641.

Demir, U., Kozan, A., & Özer, S. (2022). Experimental investigation of the effect of urea addition to fuel on engine performance and emissions in diesel engines. *Fuel*, *311*.

Deutscher Bundestag. (2020). Mehrheit der Fraktionen gegen den Begriff "Rasse" im Grundgesetz. https://www.bundestag.de/dokumente/textarchiv/2020/kw48-de-rassismus-807790, last accessed on 27.07.2023.

Di Rattalma, M. F. (2017). *The dieselgate: A legal perspective*. Springer.

Diakopoulos, N. (2014). Algorithmic accountability reporting: On the investigation of black boxes. *Tow Center for Digital Journalism*. https://doi.org/10.7916/D8ZK5TW2

Diakopoulos, N. (2020). Transparency. *The Oxford handbook of ethics of AI*, *17*(4), 197–213.

Dieterich, W., Mendoza, C., & Brennan, T. (2016). Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpointe Inc*, *7*(4), 1–36.

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, *27*(3), 326–327.

DIN/DKE. (2020). German standardization roadmap on artificial intelligence 1.0. *DIN/DKE, Berlin/Frankfurt*.
Author's contribution: For this version, I mainly participated in the ethics working group.

DIN/DKE. (2023). German standardization roadmap on artificial intelligence 2.0. *DIN/DKE, Berlin/Frankfurt*.
Author's contribution: For this version, I mainly participated in the working groups Ethics and Glossary, contributed to the topics Explainability/Transparency and Fairness across all working groups, and supported the introduction of the Assurance Case framework. Since the glossary had to be shortened considerably, the glossary team decided to publish the full version separately (see http://www.ai-glossary.org/index.php). Note that the German version was already published in 2022.

Donoho, D. (2017). 50 years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

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

Dwork, C., Immorlica, N., Kalai, A. T., & Leiserson, M. (2018). Decoupled classifiers for group-fair and efficient machine learning. *Conference on Fairness, Accountability and Transparency*, 119–133.

Elliott, K., Price, R., Shaw, P., Spiliotopoulos, T., Ng, M., Coopamootoo, K., & van Moorsel, A. (2021). Towards an equitable digital society: Artificial intelligence (ai) and corporate digital responsibility (cdr). *Society*, *58*(3), 179–188.

Embrechts, M. J., Arciniegas, F. A., Ozdemir, M., & Kewley, R. H. (2003). Data mining for molecules with 2-d neural network sensitivity analysis. *International Journal of smart engineering system design*, *5*(4), 225–239.

EPIC. (2020). Liberty at Risk: Pre-Trial Risk Assessment Tools in the U.S. *Electronic Privacy Information Center*. https://archive.epic.org/LibertyAtRiskReport.pdf, last accessed on 04.04.2023.

Eren, O., & Mocan, N. (2018). Emotional judges and unlucky juveniles. *American Economic Journal: Applied Economics*, *10*(3), 171–205.

European Commission. (2021). *Proposal for a regulation of the European Parliament and the Council laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union legislative acts* (Report). Brussels. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELLAR:e0649735-a372-11eb-9585-01aa75ed71a1

Evans, R. B., & Savoia, A. (2007). Differential testing: A new approach to change detection. *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering: Companion Papers*, 549–552.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268.

Felzmann, H., Fosch-Villaronga, E., Lutz, C., & Tamò-Larrieux, A. (2020). Towards Transparency by Design for Artificial Intelligence. *Science and Engineering Ethics*, *26*(6), 3333–3361.

Fish, B., Kun, J., & Lelkes, Á. D. (2016). A confidence-based approach for balancing fairness and accuracy. *Proceedings of the 2016 SIAM International Conference on Data Mining*, 144–152.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, *20*(177), 1–81.

Fracarolli Nunes, M., & Lee Park, C. (2016). Caught red-handed: The cost of the volkswagen dieselgate. *Journal of Global Responsibility*, *7*(2), 288–302.

Freitas, A. A. (2014). Comprehensible classification models: A position paper. *ACM SIGKDD explorations newsletter*, *15*(1), 1–10.

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE transactions on neural networks and learning systems*, *25*(5), 845–869.

Frouillou, L. (2016). Post-bac admission: an algorithmically constrained "free choice". *Justice spatiale - Spatial justice*, *10*. http://www.jssj.org/article/admission-post-bac-un-libre-choix-sous-contrainte-algorithmique/, last accessed on 03.05.2023.

Gaddis, S. M. (2018). An Introduction to Audit Studies in the Social Sciences. In S. M. Gaddis (Ed.), *Audit Studies: Behind the Scenes with Theory, Method, and Nuance* (pp. 3–44). Springer International Publishing. https://doi.org/10.1007/978-3-319-71153-9_1

Gao, H., & Ding, X. (2022). The research landscape on the artificial intelligence: A bibliometric analysis of recent 20 years. *Multimedia Tools and Applications*, *81*(9), 12973–13001.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: Methods and prospects. *Big Data Analytics*, *1*(1), 1–22.

Gärtner, M. (2012). *Atdd by example: A practical guide to acceptance test-driven development*. Addison-Wesley.

Garvie, C., Bedoya, A., & Frankle, J. (2016). The perpetual line-up: Unregulated police face recognition in america. *Georgetown Law, Center on Privacy & Technology*. https://www.perpetuallineup.org/, last accessed on 03.08.20223.

Gates, S. W., Perry, V. G., & Zorn, P. M. (2002). Automated underwriting in mortgage lending: Good news for the underserved? *Housing Policy Debate*, *13*(2), 369–391.

Gauerhof, L., Munk, P., & Burton, S. (2018). Structuring validation targets of a machine learning function applied to automated driving. *Computer Safety, Reliability, and Security: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings 37*, 45–58.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, *64*(12), 86–92.

Gilotte, A., Calauzènes, C., Nedelec, T., Abraham, A., & Dollé, S. (2018). Offline A/B Testing for Recommender Systems. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 198–206.

Giuntella, O., Hyde, K., Saccardo, S., & Sadoff, S. (2021). Lifestyle and mental health disruptions during covid-19. *Proceedings of the National Academy of Sciences*, *118*(9).

Glinz, M. (2014). A glossary of requirements engineering terminology. *Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version*, *1*, 56.

Goel, N., Yaghini, M., & Faltings, B. (2018). Non-discriminatory machine learning through convex fairness criteria. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 116–116.

Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, *32*, 100577.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint, arXiv:1412.6572*.

Grari, V., Ruf, B., Lamprier, S., & Detyniecki, M. (2020). Achieving fairness with decision trees: An adversarial approach. *Data Science and Engineering*, *5*(2), 99–110.

Grgić-Hlača, N., Zafar, M. B., Gummadi, K. P., & Weller, A. (2016). The case for process fairness in learning: Feature selection for fair decision making. *NIPS symposium on machine learning and the law*, *1*(2), 11.

Grgić-Hlača, N., Zafar, M. B., Gummadi, K. P., & Weller, A. (2018). Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. *Thirty-Second AAAI Conference on Artificial Intelligence*, 51–60.

Grimmelmann, J. (2008). The google dilemma. *New York Law School Law Review; NYLS Legal Studies Research Paper*, *53*(08/09-2), 939. https://ssrn.com/abstract=1160320

Groce, A., Holzmann, G., & Joshi, R. (2007). Randomized Differential Testing as a Prelude to Formal Verification. *29th International Conference on Software Engineering (ICSE'07)*, 621–631.

Groen, E. C., Kopczyńska, S., Hauer, M. P., Krafft, T. D., & Doerr, J. (2017). Users—the hidden software product quality experts?: A study on how app users report quality aspects in online reviews. *2017 IEEE 25th international requirements engineering conference (RE)*, 80–89.

Author's contribution: Tobias Krafft and I were equally responsible for pre-processing further test data and improving the regular expressions. I found a way to improve the recognition of negations, which improved prediction precision by around 10% (see Table 5). We both contributed smaller sections of text to the paper.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42.

Haag, M. (2022). Algorithmendiskriminierung unter dem AGG und den Gleichbehandlungsrichtlinien - ausgewählte Problemfelder und Reformvorschläge. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 119–139.

Haeri Amir, M., Hauer, M. P., & Zweig, K. (2023). Equality of quality: The relation between quality measures and fairness measures for evaluating machine learning models.
Author's contribution: All authors contributed approximately equally to this publication. The paper is still under review.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239.

Hall, M. A. (1999, April). *Correlation-based feature selection for machine learning* [Doctoral dissertation, University of Waikato Hamilton].

Hall, P. (2018). On the art and science of machine learning explanations. *arXiv preprint arXiv:1810.02909*.

Hallensleben, S., Hustedt, C., Fetic, L., Fleischer, T., Grünke, P., Hagendorff, T., Hauer, M. P., Hauschke, A., Heesen, J., Herrmann, M., Hillerbrand, R., Hubig, C., Kaminski, A., Krafft, T. D., Loh, W., Otto, P., & Puntschuh, M. (2020). From principles to practice - an interdisciplinary framework to operationalise ai ethics. *iRights. Lab, Tech. Rep.* https://www.bertelsmann-stiftung.de/fileadmin/files/BSt/Publikationen/GrauePublikationen/WKIO_2020_final.pdf.
Author's contribution: Tobias Krafft and I wrote Chapter 3 with equal participation. In terms of content, my own novel contribution primarily involved dividing the dimensions of the risk matrix into individual sub-aspects. My collection of AI-based systems that had negative effects on individuals, groups of people, or society as a whole due to errors or unintended side effects served as the basis for the considerations.

Hardt, M., Price, E., Srebro, N., et al. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 3315–3323.

Harel-Canada, F., Wang, L., Gulzar, M. A., Gu, Q., & Kim, M. (2020). Is neuron coverage a meaningful measure for testing deep neural networks? *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 851–862.

Hauer, M. P., Adler, R., & Zweig, K. (2021). Assuring fairness of algorithmic decision making. *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 110–113.
Author's contribution: Rasmus Adler and I developed the idea to extend the Assurance Case framework from safety engineering with Acceptance Test-Driven Development from agile software development methodologies to make it suitable for arguing extra-functional properties of AI-based systems together. I was responsible for the first draft, communication and organization.

Hauer, M. P., Hofmann, X. C., Krafft, T. D., Zweig, K. A., et al. (2020). Quantitative analysis of automatic performance evaluation systems based on the h-index. *Scientometrics*, *123*(2), 735–751.
Author's contribution: This paper is based on my Master's thesis and Xavier Hofmann's Bachelor's thesis. I made the main contribution to the writing and composition of the paper. I was responsible for communication and organization.

Hauer, M. P., Kevekordes, J., & Haeri, M. A. (2021). Legal perspective on possible fairness measures–A legal discussion using the example of hiring decisions. *Computer Law & Security Review*, *42*, 105583.
Author's contribution: I had the idea for this paper and moderated the interdisciplinary discussions between the legal perspective (Johannes Kevekordes) and the computer science perspective (Maryam Amir Haeri). I was responsible for providing a first draft, communication, organization, and quality assurance.

Hauer, M. P., Krafft, T. D., Sesing-Wagenpfeil, D., Zweig, P., et al. (2023). Quantitative study about the estimated impact of the ai act. *arXiv preprint arXiv:2304.06503*.
Author's contribution: Around half of the publication is based on my written contribution. However, the main effort clearly was the study itself, to which I contributed around 30%. A journal submission is currently being processed.

Hauer, M. P., Krafft, T. D., & Zweig, K. (2023). Overview of transparency and inspectability mechanisms to achieve accountability of ai systems. *Data and Policy*.
Author's contribution: The paper is a product of close collaboration with Tobias D. Krafft, on which we worked together closely and made significant contributions that bridge our respective research fields. Our contributions to the paper share many similarities in terms of their impact and importance. The paper has already been accepted but is still under review.

Hauer, M. P., Müller-Kress, L., Leimüller, G., & Zweig, K. (2023). Using Assurance Cases to assure the fulfillment of non-functional requirements of AI-based systems - Lessons learned. *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 172–179.
Author's contribution: This publication expands on the ideas discussed in Hauer, Adler, and Zweig, 2021 and tests them in industrial practice. I had the scientific lead of the Assurance Case development process. The written elaboration of the paper predominantly draws from my work and contributions.

Hauer, M. P., & Zweig, K. (2021). Chancen und risiken algorithmischer entscheidungen. *Human Ressource Manager*, *01/2021*, 48–53.
Author's contribution: I developed a first draft and refined it together with Katharina Zweig. I was responsible for communication and organization.

Heaton, J. (2016). An empirical analysis of feature engineering for predictive modeling. *SoutheastCon 2016*, 1–6.

Hedayat, A. S., Sloane, N. J. A., & Stufken, J. (2012). *Orthogonal arrays: Theory and applications*. Springer Science & Business Media.

Heesen, J., Müller-Quade, J., Wrobel, S., & et al. (2020). *Zertifizierung von KI-Systemen – Kompass für die Entwicklung und Anwendung vertrauenswürdiger KI-Systeme* (tech. rep.) (https://www.plattform-lernende-systeme.de/files/Downloads/Publikationen/AG1_3_Whitepaper_Zertifizierung_KI_Systemen.pdf). Whitepaper aus der Plattform Lernende Systeme.

Herden, C. J., Alliu, E., Cakici, A., Cormier, T., Deguelle, C., Gambhir, S., Griffiths, C., Gupta, S., Kamani, S. R., Kiratli, Y.-S., et al. (2021). "Corporate Digital Responsibility" New corporate responsibilities in the digital age. *Sustainability Management Forum| Nachhaltigkeits-ManagementForum*, *29*(1), 13–29.

High-Level Expert Group on AI. (2019a). *Ethics guidelines for trustworthy ai* (Report). Brussels. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

High-Level Expert Group on AI. (2019b). *Policy and Investment Recommendations for Trustworthy AI* (Report). Brussels. https://digital-strategy.ec.europa.eu/en/library/policy-and-investment-recommendations-trustworthy-artificial-intelligence

High-Level Expert Group on AI. (2020a). *Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment* (Report). Brussels. https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

High-Level Expert Group on AI. (2020b). *Sectoral Considerations on Policy and Investment Recommendations for Trustworthy AI* (Report). Brussels. https://futurium.ec.europa.eu/en/european-ai-alliance/document/ai-hleg-sectoral-considerations-policy-and-investment-recommendations-trustworthy-ai

Hilbert, M. (2012). Toward a synthesis of cognitive biases: How noisy information processing can bias human decision making. *Psychological bulletin*, *138*(2), 211.

Hoffmann, H., Vogt, V., Hauer, M. P., & Zweig, K. (2022a). Fairness by awareness? on the inclusion of protected features in algorithmic decisions. *Computer Law & Security Review*, *44*, 105658.
Author's contribution: This is basically an English translation of Hoffmann, Vogt, Hauer, and Zweig, 2022b. I was responsible for providing the first translation.

Hoffmann, H., Vogt, V., Hauer, M. P., & Zweig, K. (2022b). Fairness by awareness? Zur Einbeziehung geschützter Merkmale in algorithmische Entscheidungen. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 191–220.
Author's contribution: Katharina Zweig and I were responsible for the technical elaborations, Hanna Hoffmann and Verena Vogt for the legal elaborations. We each ensured that the explanations of the other specialist group are comprehensible to experts of our own discipline. I was responsible for communication and organization.

Holl, J., Kernbeiß, G., & Wagner-Pinter, M. (2019). *Personenbezogene Wahrscheinlichkeitsaussagen ("Algorithmen") - Stichworte zur Sozialverträglichkeit* (tech. rep.). Technical report, Synthesis Forschung Gesellschaft mbH.

Holl, J., Kernbeiß, G., & Wagner-Pinter, M. (2018). Das AMS-Arbeitsmarktchancen-Modell. *Arbeitsmarktservice Österreich, Wien*.

Howden, W. E. (1978). Theoretical and empirical studies of program testing. *IEEE Transactions on Software Engineering*, *SE-4*(4), 293–298.

Hynes, N., Sculley, D., & Terry, M. (2017). The Data Linter: Lightweight, Automated Sanity Checking for ML Data Sets. *NIPS MLSys Workshop*, *1*.

Ibrahim, A., Kyriakopoulos, S., & Pretschner, A. (2021). Causality-based accountability mechanisms for socio-technical systems. *Journal of Responsible Technology*, *7*, 100016.

IEEE Std 610.12-1990. (1990). IEEE Standard Glossary of Software Engineering Terminology, 1–84. https://doi.org/10.1109/IEEESTD.1990.101064

IEEE Std 829-2008. (2008). IEEE Standard for Software and System Test Documentation, 1–150. https://doi.org/10.1109/IEEESTD.2008.4578383

Iosifidis, V., & Ntoutsi, E. (2019). Adafair: Cumulative fairness adaptive boosting. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 781–790.

Itkonen, J., & Rautiainen, K. (2005). Exploratory testing: A multiple case study. *2005 International Symposium on Empirical Software Engineering, 2005.*, 10–pp.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*, *29*(3), 31–44.

Janich, P. (2015). *Handwerk und mundwerk: Über das herstellen von wissen*. CH Beck.

Jiang, Y., Li, X., Luo, H., Yin, S., & Kaynak, O. (2022). Quo vadis artificial intelligence? *Discover Artificial Intelligence*, *2*(1), 4.

Jöckel, L., Bauer, T., Kläs, M., Hauer, M. P., & Groß, J. (2021). Towards a common testing terminology for software engineering and data science experts. *International Conference on Product-Focused Software Process Improvement*, 281–289.
Author's contribution: All authors contributed approximately equally to this publication.

Jorgensen, P. C. (2014). *Software testing: A craftsman's approach* (4th ed.). Auerbach Publications.

Kacianka, S., Beckers, K., Kelbert, F., & Kumari, P. (2017). How accountability is implemented and understood in research tools. *International Conference on Product-Focused Software Process Improvement*, 199–218.

Kaminski, A. (2017). Hat Vertrauen Gründe oder ist Vertrauen ein Grund?–Eine (dialektische) Tugendtheorie von Vertrauen und Vertrauenswürdigkeit. In *Praxis und> zweite natur<* (pp. 167–188). Brill mentis.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, *33*(1), 1–33.

Kamiran, F., Calders, T., & Pechenizkiy, M. (2010). Discrimination aware decision tree learning. *2010 IEEE International Conference on Data Mining*, 869–874.

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision theory for discrimination-aware classification. *2012 IEEE 12th International Conference on Data Mining*, 924–929.

Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 35–50.

Kamishima, T., Akaho, S., & Sakuma, J. (2011). Fairness-aware learning through regularization approach. *2011 IEEE 11th International Conference on Data Mining Workshops*, 643–650.

Kanewala, U., & Bieman, J. M. (2013). Using machine learning techniques to detect metamorphic relations for programs without test oracles. *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, 1–10.

Karp, C. (2020). Grieving mother whose son, 28, killed himself after he was incorrectly billed $28k by Centrelink breaks her silence on PM's move to repay 'unlawful debts'. https://www.dailymail.co.uk/news/article-8383213/Jennifer-Millers-son-killed-received-Centrelink-debt-28-00.html, last accessed on 01.07.2023.

Kelly, T. P., et al. (1999). *Arguing safety: A systematic approach to managing safety cases* [Doctoral dissertation, University of York York, UK].

Kevekordes, J., Hauer, M. P., & Haeri, M. A. (2022). Rechtliche bewertung möglicher fairnessmaße. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 141–190.
Author's contribution: This is basically a German translation of Hauer, Kevekordes, and Haeri, 2021 with a stronger focus on the legal discussion. I was responsible for providing the first translation.

Kim, M. P., Ghorbani, A., & Zou, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 247–254.

Kim, Y. W. (2003). Efficient use of code coverage in large-scale software development. *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative Research*, 145–155.

Klare, B. F., Burge, M. J., Klontz, J. C., Bruegge, R. W. V., & Jain, A. K. (2012). Face recognition performance: Role of demographic information. *IEEE Transactions on information forensics and security*, *7*(6), 1789–1801.

Klees, G., Ruef, A., Cooper, B., Wei, S., & Hicks, M. (2018). Evaluating fuzz testing. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2123–2138.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., & Mullainathan, S. (2018). Human decisions and machine predictions. *The quarterly journal of economics*, *133*(1), 237–293.

Knight, J. C., & Leveson, N. G. (1986). An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Transactions on software engineering*, *SE-12*(1), 96–109.

Kohavi, R., & Longbotham, R. (2017). Online controlled experiments and a/b testing. *Encyclopedia of machine learning and data mining*, *7*(8), 922–929.

Kohli, N., Barreto, R., & Kroll, J. A. (2018). Translation tutorial: A shared lexicon for research and practice in human-centered software systems. *1st Conference on Fairness, Accountability, and Transparency. New York, NY, USA*, *7*.

Kok, J. N., Boers, E. J., Kosters, W. A., Van der Putten, P., & Poel, M. (2009). Artificial Intelligence: Definition, Trends, Techniques, and Cases. *Artificial intelligence*, *1*, 270–299.

Krafft, T. D., Gamer, M., & Zweig, K. A. (2019). What did you see? a study to measure personalization in google's search engine. *EPJ Data Science*, *8*(1), 38.

Krafft, T. D., Hauer, M. P., & Zweig, K. (2023). Black box testing and auditing of bias in ADM systems. *Minds and Machines*.
Author's contribution: The paper is the outcome of a close collaboration between Tobias Krafft and me. Throughout the process, we worked together and made substantial contributions that effectively connect our respective research fields. Our individual contributions to the paper exhibit significant similarities in terms of their impact and importance. The paper is still under review.

Krafft, T. D., Hauer, M. P., & Zweig, K. A. (2020). Why do we need to be bots? what prevents society from detecting biases in recommendation systems. *International Workshop on Algorithmic Bias in Search and Recommendation*, 27–34.
Author's contribution: This paper is based on a joint project for which Tobias Krafft was more responsible for the research and I was more responsible for the practical application. Around 40% of the publication are based on my written contribution.

Krafft, T. D., Reber, M., Krafft, R., Coutrier, A., & Zweig, K. A. (2021). Crucial challenges in large-scale black box analyses. *International Workshop on Algorithmic Bias in Search and Recommendation*, 143–155.

Krafft, T. D., & Zweig, K. (2019). Transparenz und Nachvollziehbarkeit algorithmenbasierter Entscheidungsprozesse. Ein Regulierungsvorschlag aus sozioinformatischer Perspektive. *Verbraucherzentrale Bundesverband e. V. Berlin*.

Krafft, T. D., & Zweig, K. A. (2019). Transparenz und Nachvollziehbarkeit algorithmenbasierter Entscheidungsprozesse | Ein Regulierungsvorschlag [Verbraucherzentrale Bundesverband. Online verfügbar unter https://www.vzbv.de/sites/default/files/downloads/2019/05/02/19-01-22_zweig_krafft_transparenz_adm-neu.pdf; aufgerufen am 31.12.2022].

Krafft, T. D., Zweig, K. A., & König, P. D. (2022). How to regulate algorithmic decision-making: A framework of regulatory requirements for different applications. *Regulation & Governance*, *16*(1), 119–136.

Kraus, T., Ganschow, L., Eisenträger, M., & Wischmann, S. (2022). Explainable ai: Requirements, use cases and solutions. *Study commissioned by the Federal Ministry for Economic Affairs and Climate Action (BMWK) within the framework of the mandated accompanying research for the technology program "Artificial Intelligence as a Driver for Economically Relevant Ecosystems" (AI innovation competition)*. https://www.digitale-technologien.de/DT/Redaktion/EN/Downloads/Publikation/KI_Inno_Xai_Studie.html

Kroll, J. A., Barocas, S., Felten, E. W., Reidenberg, J. R., Robinson, D. G., & Yu, H. (2016). Accountable algorithms. *U. Pa. L. Rev.*, *165*, 633.

Kroll, J. A., Huey, J., Barocas, S., Felten, E. W., Reidenberg, J. R., Robinson, D. G., & Yu, H. (2017). Accountable algorithms. *University of Pennsylvania Law Review*, *165*, 633.

Kuhn, D. R., Kacker, R. N., & Lei, Y. (2013). *Introduction to combinatorial testing*. CRC press.

Kuhn, M., Johnson, K., et al. (2013). *Applied predictive modeling* (Vol. 26). Springer.

Kuhn, T. (2014). A survey and classification of controlled natural languages. *Computational linguistics*, *40*(1), 121–170.

Kunze, L., Leimüller, G., Müller-Kress, L., & Hauer, M. P. (2023). Method handbook: Assurance cases for fair ai systems. https://www.fairbydesign.eu/s/Handbook-for-Assurance-Cases-for-fair-AI-systems.pdf, last accessed on 20.04.2023.
Author's contribution: I provided most of the content in preparation but was barely involved in writing it down and visualizing it as a handbook. However, I was responsible for ensuring that the wording was precise and correct.

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. *Advances in Neural Information Processing Systems*, 4066–4076.

Lahno, B. (2002). Der Begriff des Vertrauens. In *Der begriff des vertrauens*. Brill mentis.

Lahoti, P., Gummadi, K. P., & Weikum, G. (2019). Ifair: Learning individually fair data representations for algorithmic decision making. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1334–1345.

Larus, J., Hankin, C., Carson, S. G., Christen, M., Crafa, S., Grau, O., Kirchner, C., Knowles, B., McGettrick, A., Tamburri, D. A., et al. (2018). When computers decide: European recommendations on machine-learned automated decision making.

Latonero, M. (2018). Governing artificial intelligence: Upholding human rights & dignity.

Leadership Conference on Civil and Human Rights. (2018). More than 100 civil rights, digital justice, and community-based organizations raise concerns about pretrial risk assessment. https://civilrights.org/2018/07/30/more-than-100-civil-rights-digital-justice-and-community-based-organizations-raise-concerns-about-pretrial-risk-assessment/, last accessed on 31.03.2023.

Lepri, B., Oliver, N., Letouzé, E., Pentland, A., & Vinck, P. (2018). Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology*, *31*(4), 611–627.

Lerner, J. S., & Tetlock, P. E. (1999). Accounting for the effects of accountability. *Psychological bulletin*, *125*(2), 255.

Leung, H. K., & White, L. (1989). Insights into regression testing (software testing). *Proceedings. Conference on Software Maintenance-1989*, 60–69.

Li, O., Liu, H., Chen, C., & Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1).

Liberati, E. G., Ruggiero, F., Galuppo, L., Gorli, M., González-Lorenzo, M., Maraldi, M., Ruggieri, P., Polo Friz, H., Scaratti, G., Kwag, K. H., et al. (2017). What hinders the uptake of computerized decision support

systems in hospitals? A qualitative study and framework for implementation. *Implementation Science*, *12*(1), 1–13.

Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

Lobschat, L., Mueller, B., Eggers, F., Brandimarte, L., Diefenbach, S., Kroschke, M., & Wirtz, J. (2021). Corporate digital responsibility. *Journal of Business Research*, *122*, 875–888.

Lohia, P. K., Ramamurthy, K. N., Bhide, M., Saha, D., Varshney, K. R., & Puri, R. (2019). Bias mitigation post-processing for individual and group fairness. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2847–2851.

Lopes, A. T., De Aguiar, E., De Souza, A. F., & Oliveira-Santos, T. (2017). Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order. *Pattern recognition*, *61*, 610–628.

Lovelace, A., & DataKind, U. (2020). *Examining the black box: Tools for assessing algorithmic systems* (tech. rep.). Technical report, Ada Lovelace Institute, https://ico. org. uk/media/about . . .

Loyola-Gonzalez, O. (2019). Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access*, *7*, 154096–154113.

Lu, J., Lee, D., Kim, T. W., & Danks, D. (2019). Good explanation for algorithmic transparency. *Available at SSRN 3503603*.

Luengo, J., García, S., & Herrera, F. (2012). On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and information systems*, *32*(1), 77–108.

Maksimov, M., Fung, N. L., Kokaly, S., & Chechik, M. (2018). Two decades of assurance case tools: A survey. *International Conference on Computer Safety, Reliability, and Security*, 49–59.

Mamone, S. (2000). Documentation testing. *ACM SIGSOFT Software Engineering Notes*, *25*(2), 26–29.

Mannion, M., & Keepence, B. (1995). Smart requirements. *ACM SIGSOFT Software Engineering Notes*, *20*(2), 42–47.

Marick, B., Smith, J., & Jones, M. (1999). How to misuse code coverage. *Proceedings of the 16th International Conference on Testing Computer Software*, 16–18.

Marijan, D., Gotlieb, A., & Ahuja, M. K. (2019). Challenges of testing machine learning based systems. *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, 101–102.

McCarthy, J. (2007). What is artificial intelligence. http://jmc.stanford.edu/articles/whatisai/whatisai.pdf, last accessed on 11.08.2023.

McGregor, S. (2021). Preventing repeated real world ai failures by cataloging incidents: The ai incident database. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(17), 15458–15463.

McKeeman, W. M. (1998). Differential testing for software. *Digital Technical Journal*, *10*(1), 100–107.

Michener, G., & Bersch, K. (2013). Identifying transparency. *Information Polity*, *18*(3), 233–242.

Mihale-Wilson, C., Hinz, O., van der Aalst, W., & Weinhardt, C. (2022). Corporate digital responsibility: Relevance and opportunities for business and information systems engineering. *Business & Information Systems Engineering*, *64*(2), 127–132.

Mihale-Wilson, C. A., Zibuschka, J., Carl, K. V., & Hinz, O. (2021). Corporate digital responsibility-extended conceptualization and empirical assessment. *ECIS*.

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

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the conference on fairness, accountability, and transparency*, 220–229.

Mittelstadt, B., Wachter, S., & Russell, C. (2023). The Unfairness of Fair Machine Learning: Levelling down and strict egalitarianism by default. *arXiv preprint arXiv:2302.02404*.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, *65*, 211–222.

Moran, R. (2006). Getting told and being believed. *The epistemology of testimony*, 272–306.

Moser, C. A. (1952). Quota sampling. *Journal of the Royal Statistical Society. Series A (General)*, *115*(3), 411–423.

Mueller, B. (2022). Corporate digital responsibility. *Business & Information Systems Engineering*, *64*(5), 689–700.

Mulgan, R. (2000). 'accountability': An ever-expanding concept? *Public administration*, *78*(3), 555–573.

Myers, G. J., Thomas, T. M., Badgett, T., & Sandler, C. (2004). *The art of software testing* (Vol. 2). John Wiley & Sons.

Nellis, A. (2021). The color of justice: Racial and ethnic disparity in state prisons.

Nemitz, P. (2018). Constitutional democracy and technology in the age of artificial intelligence. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *376*(2133), 20180089.

Nguyen, A., Yosinski, J., & Clune, J. (2016). Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.

Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, *2*(2), 29–50.

Nie, C., & Leung, H. (2011). A survey of combinatorial testing. *ACM Computing Surveys (CSUR)*, *43*(2), 1–29.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565–1567.

Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M.-E., Ruggieri, S., Turini, F., Papadopoulos, S., Krasanakis, E., et al. (2020). Bias in data-driven artificial intelligence systems—An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(3), e1356.

Ohm, P. (2009). Broken promises of privacy: Responding to the surprising failure of anonymization. *Ucla L. Rev.*, *57*, 1701.

O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown.

Ontañón, S. (2020). An overview of distance and similarity functions for structured data. *Artificial Intelligence Review*, *53*(7), 5309–5351.

Orabi, M., Mouheb, D., Al Aghbari, Z., & Kamel, I. (2020). Detection of bots in social media: A systematic review. *Information Processing & Management*, *57*(4), 102250.

Orwat, C. (2019). *Diskriminierungsrisiken durch verwendung von algorithmen*. Antidiskriminierungsstelle des Bundes.

Orwat, C., Bareis, J., Folberth, A., Jahnel, J., & Wadephul, C. (2022). Risikoregulierung von künstlicher intelligenz und automatisierten entscheidungen. *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, 255–287.

Paeffgen, N., & Perdrisat, S. (2021). Labels and certifications for the digital world - mapping the international landscape. *Swiss Digital Initiative*.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* (tech. rep.). Stanford InfoLab.

Palos-Sánchez, P. R., Baena-Luna, P., Badicu, A., & Infante-Moro, J. (2022). Artificial intelligence and human resources management: A bibliometric analysis. *Applied Artificial Intelligence*, *36*(1), 2145631.

Panteras, G., Lu, X., Croitoru, A., Crooks, A., & Stefanidis, A. (2016). Accuracy of user-contributed image tagging in flickr: A natural disaster case study. *Proceedings of the 7th 2016 International Conference on Social Media & Society*, 1–6.

Panunzio, M., & Vardanega, T. (2014). An architectural approach with separation of concerns to address extra-functional requirements in the development of embedded real-time software systems. *Journal of Systems Architecture*, *60*(9), 770–781.

Papadakis, M., Kintis, M., Zhang, J., Jia, Y., Le Traon, Y., & Harman, M. (2019). Mutation testing advances: An analysis and survey. In *Advances in computers* (pp. 275–378, Vol. 112). Elsevier.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Pasquale, F. (2015). *The black box society*. Harvard University Press.

Pearl, J., et al. (2000). Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, *19*, 2.

Pearl, J., et al. (2009). Causal inference in statistics: An overview. *Statistics surveys*, *3*, 96–146.

Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). Deepxplore: Automated whitebox testing of deep learning systems. *proceedings of the 26th Symposium on Operating Systems Principles*, 1–18.

Petsios, T., Tang, A., Stolfo, S., Keromytis, A. D., & Jana, S. (2017a). Nezha: Efficient domain-independent differential testing. *2017 IEEE Symposium on security and privacy (SP)*, 615–632.

Petsios, T., Tang, A., Stolfo, S., Keromytis, A. D., & Jana, S. (2017b). Nezha: Efficient domain-independent differential testing. *2017 IEEE Symposium on security and privacy (SP)*, 615–632.

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017). Data management challenges in production machine learning. *Proceedings of the 2017 ACM International Conference on Management of Data*, 1723–1726.

Porter, Z., Habli, I., & McDermid, J. (2022). A principle-based ethical assurance argument for AI and autonomous systems. *arXiv preprint arXiv:2203.15370*.

Potter, B., & McGraw, G. (2004). Software security testing. *IEEE Security & Privacy*, *2*(5), 81–85.

Powers, D. M. (2020). Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Prechelt, L. (1998). Automatic early stopping using cross validation: Quantifying the criteria. *Neural networks*, *11*(4), 761–767.

Pugh, K. (2010). *Lean-agile acceptance test-driven development: Better software through collaboration*. Pearson Education.

Pugh, K. (2015). Acceptance test-driven development: Better software through collaboration. Excerpt from PNSQC 2015 Proceedings: http://uploads.pnsqc.org/2015/papers/Pugh-Acceptance-Test-Driven-Development.pdf.

Ramakrishnan, R., Gehrke, J., & Gehrke, J. (2003). *Database management systems* (Vol. 3). McGraw-Hill New York.

Reber, M., Krafft, T. D., Krafft, R., Zweig, K. A., & Couturier, A. (2020). Data donations for mapping risk in google search of health queries: A case study of unproven stem cell treatments in sem. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2985–2992.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*.

Ricci, P., & Sheng, H. (2013). Benefits and limitations of the precautionary principle.

Rinehart, D. J., Knight, J. C., & Rowanhill, J. (2017). *Understanding what it means for assurance cases to "work", NASA/CR–2017-219582* (tech. rep.). NASA.

Ripley, B. D. (1996). *Pattern recognition and neural networks* (Vol. 8). Cambridge University Press.

Roh, Y., Heo, G., & Whang, S. E. (2019). A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*.

Romei, A., & Ruggieri, S. (2014). A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, *29*(5), 582–638.

Rosalie, W. (2022). Why ai ethics is a critical theory. *Philosophy & Technology*, *35*(1).

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Roskam, E. E. (1989). Operationalization, a superfluous concept. *Quality and Quantity*, *23*, 237–275.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rudin, C. (2019a). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Rudin, C. (2019b). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Runze, G., Haimerl, M., Hauer, M. P., Holoyad, T., Obert, O., Pöhls, H., Tagiew, R., & Ziehn, J. (2023). Das AI-Glossary als Weg aus Babylon - Ein Werkzeug für eine gemeinsame KI-Terminologie. *JavaSpektrum*, *03/2023*, 42–46.

Author's contribution: All authors contributed equally to this publication.

Sandvig, C., Hamilton, K., Karahalios, K., & Langbort, C. (2014). Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Data and discrimination: converting critical concerns into productive inquiry*, *22*, 4349–4357.

Sazli, M. H. (2006). A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, *50*(01).

Schmidt, P., Stummer, H., Mally, T., Lohninger, T., & Lohninger, D. (2022). *Abschlussergebnisse zum projekt: "Stoppt den AMS-Algorithmus"* (Report). epicenter.works. https://amsalgorithmus.at/static/AMS_Algorithmus_Report_epicenter_works_Petra_Schmidt.pdf

Schwab, K. (2017). *The fourth industrial revolution*. Currency.

Schwartz, R., Vassilev, A., Greene, K., Perine, L., Burt, A., Hall, P., et al. (2022). Towards a standard for identifying and managing bias in artificial intelligence. *NIST Special Publication*, *1270*, 1–77.

Segura, S., Fraser, G., Sanchez, A. B., & Ruiz-Cortés, A. (2016). A survey on metamorphic testing. *IEEE Transactions on software engineering*, *42*(9), 805–824.

Sentilles, S., Štěpán, P., Carlson, J., & Crnković, I. (2009). Integration of extra-functional properties in component models. *International Symposium on Component-Based Software Engineering*, 173–190.

Sevinchan, Y., Herdeanu, B., Mack, H., Riedel, L., & Roth, K. (2020). Boosting group-level synergies by using a shared modeling framework. *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part VII 20*, 442–456.

Shamshiri, S., Rojas, J. M., Galeotti, J. P., Walkinshaw, N., & Fraser, G. (2018). How do automatically generated unit tests influence software maintenance? *2018 IEEE 11th international conference on software testing, verification and validation (ICST)*, 250–261.

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310–316.

Shore, J., & Warden, S. (2021). *The art of agile development*. " O'Reilly Media, Inc."

Simpson, E. H. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *13*(2), 238–241.

Siroker, D., & Koomen, P. (2013). *A/B testing: The most powerful way to turn clicks into customers*. John Wiley & Sons.

SMART, G. T. D. T. (1981). Way to write management's goals and objectives/george t. doran. *Management Review*, *70*(11), 35–36.

Snow, R., O'connor, B., Jurafsky, D., & Ng, A. Y. (2008). Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. *Proceedings of the 2008 conference on empirical methods in natural language processing*, 254–263.

Sokol, K., & Flach, P. (2020). Explainability fact sheets: a framework for systematic assessment of explainable approaches. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 56–67.

Sommerville, I., & Sawyer, P. (1997). *Requirements engineering: A good practice guide*. John Wiley & Sons, Inc.

Speith, T. (2022). A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods. *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2239–2250.

Stark, L., & Hutson, J. (2021). Physiognomic artificial intelligence. *Fordham Intell. Prop. Media & Ent. LJ*, *32*, 922.

Steineck, G., & Ahlbom, A. (1992). A definition of bias founded on the concept of the study base. *Epidemiology*, 477–482.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tan, C.-H., Teo, H.-H., & Benbasat, I. (2010). Assessing screening and evaluation decision support systems: A resource-matching approach. *Information Systems Research*, *21*(2), 305–326.

Tang, X., Li, X., Ding, Y., Song, M., & Bu, Y. (2020). The pace of artificial intelligence innovations: Speed, talent, and trial-and-error. *Journal of Informetrics*, *14*(4).

Taylor, A., Marcus, M., & Santorini, B. (2003). The penn treebank: An overview. *Treebanks*, 5–22.

Taylor, R. D. (2020). Quantum artificial intelligence: A "precautionary" U.S. approach? *Telecommunications Policy*, *44*(6), 101909.

Templeton, A. R. (1998). Human races: A genetic and evolutionary perspective. *American Anthropologist*, *100*(3), 632–650.

Tharwat, A. (2020). Classification assessment methods. *Applied computing and informatics*, *17*(1), 168–192.

The Alan Turing Institute. (2021). Data science and AI in the age of covid-19 - reflections on the response of the UK's data science and AI community to the COVID-19 pandemic. https://www.turing.ac.uk/sites/default/files/2021-06/data-science-and-ai-in-the-age-of-covid_full-report_2.pdf, last accessed on 20.03.20223.

Thorun, C. (2018). Corporate digital responsibility: Unternehmerische verantwortung in der digitalen welt. *Fallstudien zur Digitalen Transformation: Case Studies für die Lehre und praktische Anwendung*, 173–191.

Thorun, C., Vetter, M., Reisch, L. A., & Zimmer, A. K. (2017). Indicators of consumer protection and empowerment in the digital world: Results and recommendations of a feasibility study.

UN Environment Programme – Finance Initiative. (2004). *Who cares wins - connecting financial markets to a changing world* (Report). The Global Compact. https://www.unepfi.org/fileadmin/events/2004/stocks/who_cares_wins_global_compact_2004.pdf

Ustun, B., Liu, Y., & Parkes, D. (2019). Fairness without harm: Decoupled classifiers with preference guarantees. *International Conference on Machine Learning*, 6373–6382.

Utting, M., & Legeard, B. (2010). *Practical model-based testing: A tools approach*. Elsevier.

Utting, M., Pretschner, A., & Legeard, B. (2012). A taxonomy of model-based testing approaches. *Software testing, verification and reliability*, *22*(5), 297–312.

van der Heijden, J. (2019). Risk governance and risk-based regulation: A review of the international academic literature. *State of the Art in Regulatory Governance Research Paper Series*.

Van Vliet, H., Van Vliet, H., & Van Vliet, J. (2008). *Software engineering: Principles and practice* (Vol. 13). John Wiley & Sons Hoboken, NJ.

Vartak, M., Rahman, S., Madden, S., Parameswaran, A., & Polyzotis, N. (2015). Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, *8*(13), 2182.

Vecchione, B., Levy, K., & Barocas, S. (2021). Algorithmic Auditing and Social Justice: Lessons from the History of Audit Studies. https://doi.org/10.1145/3465416.3483294

Velasquez, M., Andre, C., Shanks, T., & Meyer, M. J. (1990). Justice and fairness. *Issues in Ethics*, *3*(2), 1–3.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. *2018 ieee/acm international workshop on software fairness (fairware)*, 1–7.

Vigdor, N. (2019). Apple card investigated after gender discrimination complaints. *The New York Times*, *10*. https://www.nytimes.com/2019/11/10/business/Apple-credit-card-investigation.html, last accessed on 20.06.2023.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). Recaptcha: Human-based character recognition via web security measures. *Science*, *321*(5895), 1465–1468.

Vouk, M. A. (1988). On back-to-back testing. *Computer Assurance, 1988. COMPASS'88*, 84–91.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, *7*(2), 76–99.

Wachter, S., Mittelstadt, B., & Russell, C. (2021). Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. *Computer Law & Security Review*, *41*, 105567.

Wagner, B. (2018). Ethics as an escape from regulation. from "ethics-washing" to ethics-shopping?

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, *30*(4), 964–994.

Weller, A. (2019). Transparency: Motivations and challenges. In *Explainable ai: Interpreting, explaining and visualizing deep learning* (pp. 23–40). Springer.

Whately, R. (1834). *Elements of logic: Comprising the substance of the article in the encyclopaedia metropolitana, with additions &c.* (5th ed.). B. Fellowes. https://archive.org/details/bub_gb_5mgAAAAAMAAJ/page/n453/mode/2up

Whittaker, J. A. (2009). *Exploratory software testing: Tips, tricks, tours, and techniques to guide test design*. Pearson Education.

Wickens, M. R. (1972). A note on the use of proxy variables. *Econometrica: Journal of the Econometric Society*, 759–761.

Wieringa, M. (2020). What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 1–18.

Wikle, C. K., Cressie, N., Zammit-Mangion, A., & Shumack, C. (2017). A common task framework (ctf) for objective comparison of spatial prediction methodologies. *Statistics Views*.

Wilhelm, A., & Zweig, K. (n.d.). Hacking a surrogate model approach to XAI [By the time this thesis was submitted, the paper was still in progress and had not been submitted yet.].

Wilson, S. P., McDermid, J. A., Pygott, C. H., & Tombs, D. J. (1996). Assessing complex computer based systems using the goal structuring notation. *Proceedings of ICECCS'96: 2nd IEEE International Conference on Engineering of Complex Computer Systems (held jointly with 6th CSESAW and 4th IEEE RTAW)*, 498–505.

Woodworth, B., Gunasekar, S., Ohannessian, M. I., & Srebro, N. (2017). Learning non-discriminatory predictors. *Conference on Learning Theory*, 1920–1953.

Xu, Y., Chen, N., Fernandez, A., Sinno, O., & Bhasin, A. (2015). From infrastructure to culture: A/b testing challenges in large scale social networks. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2227–2236.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316.

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., & Dai, Y. (2014). Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *8*(1), 1–29.

Yang, Z., Shi, J., Asyrofi, M. H., & Lo, D. (2022). Revisiting neuron coverage metrics and quality of deep neural networks. *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 408–419.

Young, S. W. (2014). Improving library user experience with a/b testing: Principles and process. *Weave: Journal of Library User Experience*, *1*(1).

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. *International Conference on Machine Learning*, 325–333.

Zhang, M., Atwal, G., & Kaiser, M. (2021). Corporate social irresponsibility and stakeholder ecosystems: The case of volkswagen dieselgate scandal. *Strategic Change*, *30*(1), 79–85.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Zhu, H. (2015). Jfuzz: A tool for automated java unit testing based on data mutation and metamorphic testing methods. *2015 Second International Conference on Trustworthy Systems and Their Applications*, 8–15.

Žliobaitė, I., & Custers, B. (2016). Using sensitive personal data may be necessary for avoiding discrimination in data-driven decision models. *Artificial Intelligence and Law*, *24*(2), 183–201.

Zweig, K. (2023). *Die KI war's!: Von absurd bis tödlich: Die Tücken der künstlichen Intelligenz* (Vol. 1). Heyne Verlag.

Zweig, K., Hauer, M. P., & Raudonat, F. (2020). Anwendungsszenarien: KI-systeme im personal- und talentmanagement. *ExamAI – KI Testing & Auditing, Gesellschaft für Informatik e.V.*

Author's contribution: All authors contributed approximately equally to this publication. I was responsible for communication and organization.

Zweig, K. A., Krafft, T. D., & Hauer, M. P. (2017). *Dein Algorithmus - meine Meinung! Algorithmen und ihre Bedeutung f'ur Meinungsbildung und Demokratie*.

Author's contribution: The brochure was written together with Katharina Zweig and Tobias D. Krafft and has been printed and downloaded over 20,000 times.

Zweig, K. A., Krafft, T. D., Klingel, A., & Park, E. (2021). *Sozioinformatik: Ein neuer blick auf informatik und gesellschaft*. Carl Hanser Verlag GmbH Co KG.

Zweig, K. A., Wenzelburger, G., & Krafft, T. D. (2018). On chances and risks of security related algorithmic decision making systems. *European Journal for Security Research*, *3*(2), 181–203.

Marc Hauer

marc@tuev-lab.ai
0000-0002-1598-1812
Marc P. Hauer
marc-hauer

Education and career path

since Nov. 2023 **Senior AI Solution Architect**, TÜV AI Lab

since Aug. 2018 **Consultant/Freelance speaker**, consulting and science communication on the topics of media literacy, algorithms and AI, regularly on behalf of the TrustedAI GmbH and the Konrad Adenauer Stiftung, among others.
A full list of all workshops and talks is attached.

Aug. 2018 - Dec. 2023 **Media education trainer**, science communication on the topics of media literacy, algorithms and AI, on behalf of the LMZ Baden-Württemberg.
A full list of all workshops and talks is attached.

Apr. 2019 - Oct 2023 **PhD Student**, RPTU Kaiserslautern Landau (formerly TU Kaiserslautern), Algorithm Accountability Lab, research on the accountability of software development processes and (primarily AI-based) software systems.
A complete list of publications, projects and conference participations is attached.

Nov. 2016 - Sep. 2023 **Teaching Assistant**, courses of the Algorithm Accountability Lab
Tutorial conception and supervision, script drafting, supervision and assessment of seminar papers and term papers, support in conception and supervision of courses and student projects, supervision and correction of 3 Bachelor's theses and 1 Master's thesis.

Jan. 2018 - Mar. 2019 **Software Developer**, Insiders Technologies GmbH, Kaiserslautern, 3rd level support and quality assurance.

Oct. 2010 - Oct. 2017 **Applied Computer Science Studies (BA and MA)**, TU Kaiserslautern, final grade 1.5.

Certifications and Scholarship

Sep. 2020 - Sep. 2023 **Certificate in Higher Education Didactics**

Oct. 2022 **Quality Officer (TÜV)**

Apr. 2019 - Jan. 2020 **Doctoral Scholarship from the Department of Computer Science**, TU Kaiserslautern

Standardization

since Feb. 2025 **Link expert between NQSZ and JWG 6**, development of ISO 42007, conformity assessment schemes for AI systems under ISO/CASCO and ISO/IEC JTC1/SC42, DIN NA NQSZ

since Feb. 2025 **Member of the DIN Technical Commitee 147-00-03**, Basics of Conformity Assessment, DIN NA NQSZ

since Nov. 2024 **Member of the DIN Technical Commitee NA 043-01-42 WG 2**, Basic Standards and Conformity, DIN

since Mar. 2024 **Member of the Expert Testing Task Force (TTF) T038**, Towards a Harmonized Documentation Scheme for Trustworthy AI, ETSI

since Mar. 2024 **Member of the VDE SPEC 92006**, Artificial Intelligence - Requirements for AI testing tools, VDE

since Oct. 2022 **Founding member of the independent AI glossary working group of** `www.ai-glossary.org`
Development and expansion of a comprehensive AI glossary

Feb. 2024 - Feb. 2025 **Member of DIN SPEC 91512 *Fairness of AI in financial services***, DIN

Apr. 2024 - Nov. 2024 **Member of the DIN Technical Committee NA 043-01-42 GA**, Artificial Intelligence, DIN

Feb. 2024 **Initiator of DIN SPEC 91512 *Fairness of AI in financial services***, DIN

May. 2023 - Feb. 2024 **Co-Chair of the DIN SPEC Working Group *Fairness of AI in financial services***, DIN

Jan. 2022 - Sep. 2022 **Standardization roadmap of artificial intelligence 2.0 [22]**, on behalf of the *German Federal Ministry of Economics and Climate Protection*, DIN
Member of several working groups as an expert on the vertical topics of ethics, explainability, transparency and fairness

Jan. 2020 - Dec. 2020 **Standardization roadmap of artificial intelligence 1.0 [21]**, on behalf of the *German Federal Ministry of Economics and Climate Protection*, DIN
Member of the Ethics Working Group

Projects

since Jan. 2024 **Mission AI**

Mission AI is a joint project of acatech - National Academy of Science and Engineering and the Federal Ministry for Digital and Transport (BMDV). The research and development project is one of the leverage projects of the German government's digital strategy and aims to strengthen the digital competitiveness of the German economy. The initiative addresses the challenges of artificial intelligence on a broad scale - by expanding the database for AI innovations and promoting the development and growth of trustworthy, marketable AI applications. The project focuses on networking and harmonising the numerous promising approaches and foundations in Germany and Europe. I am responsible for WP1, which deals with the collection, preparation and selection of use cases for the further course of the project.

Dec. 2022 - Sep. 2023 **Industry project on AI in HR processes**

On behalf of and in cooperation with an internationally operating company, this project is about identifying ethical and legal requirements for an AI-based talent management system and developing ways to ensure or measure their fulfilment. In this project, I have taken over the scientific leadership and am responsible for the conception of the content and the scientific communication.

Jun. 2022 - Apr. 2023 **fAIr by design**

fAIr by design is a research project involving eight partners from Austria (five companies, two universities and one NGO), focusing on the practical implementation of fairness requirements into AI systems. It is a three-year project, running from 2021 until 2024, funded by the National Foundation for Research, Technology and Development Austria. I participated as an external expert to build an Assurance Case together with the companies *winnovation consulting GmbH*, *Rania Wazir e.U.*, and *rotable* to assure that an actual software product developed by rotable can be considered fair. In this project, I co-authored the publications [7] and [24].

Mar. 2021 - Okt. 2022 **Ethical and Trustworthy Artificial and Machine Intelligence (ETAMI)**

etami is a non-profit organisation that works on making ethical AI principles actionable. By translating European and global principles for ethical AI into actionable and measurable guidelines, tools and methods, etami supports and promotes trustworthy and ethical design, development, and deployment of AI systems. The consortium started working in 2019 on quality standards and conformity assessment for AI software. As an initiative of the Machine Learning Research Lab of Volkswagen Group in Munich, it gathered 17 multinationals and universities to jointly develop excellence in AI methods. In 2022, etami joined the *Big Data Value Assosiation* (BDVA) and is hosted as a Task Force since. I have been involved in the regular interdisciplinary discussions with other academics, practitioners and industry representatives and have contributed to etami's *Open Guidebook*, which aims to support the research, development and application of AI methods (`https://guidebook.etami.org`).

Jan. 2020 - Nov. 2021 **Testing, Auditing and Certification of AI (ExamAI)**

The ExamAI project, founded by the *German Federal Ministry of Labour and Social Affairs*, was led by the *Gesellschaft für Informatik e.V.* and consisted of an interdisciplinary team of (socio-) computer scientists, software engineers, legal scientists, and political scientists. Based on eleven use cases in the application areas „Human-Machine Cooperation in Industrial Production" and „AI Systems in Human Resources and Talent Management as well as in Recruiting" that were identified at the beginning of the project, the team explored the question of how appropriate control and test procedures for AI systems could look like. In (and about) this project, I co-authored the publications [3], [9], [11], [**?**], [20], [23] and [26].

Sep. 2019 - Dec. 2021 **Governance of and by Algorithms (GOAL)**

The research subject of this project was the governance of and by algorithmic decision-making systems based on machine learning methods. It was funded by the *German Federal Ministry of Science and Education*. The first primary focus of the project was on the need for governance instruments, such as value-oriented technology design, self-regulation, and technical standards, on what possibilities these offer and where there are gaps that still need to be closed. The second focus was a discussion of the possible extent to which algorithms themselves can exercise governance functions in order to reduce risks or even avoid them altogether. Based on these discussions, the needs for action were identified and addressed. In this project, I co-authored the publications [5], [8], [6], [16], [17], [18] and [19].

List of Publications

Peer-Reviewed Journal Articles and Conference Proceedings

[1] K Valerie Carl, Marc P Hauer, and Thomas Arnold. Are we still on track with our responsibility strategy? introducing an internal assessment of corporate digital responsibility engagement. In *INFORMATIK 2024*, pages 1573–1585. Gesellschaft für Informatik eV, 2024.

[2] Eduard C Groen, Sylwia Kopczyńska, Marc P. Hauer, Tobias D. Krafft, and Joerg Doerr. Users—the hidden software product quality experts?: A study on how app users report quality aspects in online reviews. In *2017 IEEE 25th international requirements engineering conference (RE)*, pages 80–89. IEEE, 2017.

[3] Marc P. Hauer, Rasmus Adler, and Katharina Zweig. Assuring Fairness of Algorithmic Decision Making. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 110–113. IEEE, 2021.

[4] Marc P. Hauer, Xavier C.R. Hofmann, Tobias D. Krafft, Katharina A. Zweig, et al. Quantitative analysis of automatic performance evaluation systems based on the h-index. *Scientometrics*, 123(2):735–751, 2020.

[5] Marc P. Hauer, Johannes Kevekordes, and Maryam Amir Haeri. Legal perspective on possible fairness measures–A legal discussion using the example of hiring decisions. *Computer Law & Security Review*, 42:105583, 2021.

[6] Marc P. Hauer, Tobias D. Krafft, and Katharina Zweig. Overview of transparency and inspectability mechanisms to achieve accountability of AI systems. *Data and Policy*.

[7] Marc P. Hauer, Lena Müller-Kress, Gertraud Leimüller, and Katharina Zweig. Using Assurance Cases to assure the fulfilment of non-functional requirements of AI-based systems - Lessons learned. In *2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, page pages are still pending. IEEE, 2023.

[8] Hanna Hoffmann, Verena Vogt, Marc P. Hauer, and Katharina Zweig. Fairness by awareness? On the inclusion of protected features in algorithmic decisions. *Computer Law & Security Review*, 44:105658, 2022.

[9] Lisa Jöckel, Thomas Bauer, Michael Kläs, Marc P. Hauer, and Janek Groß. Towards a Common Testing Terminology for Software Engineering and Data Science Experts. In *International Conference on Product-Focused Software Process Improvement*, pages 281–289. Springer, 2021.

[10] Tobias D Krafft, Marc P Hauer, and Katharina Zweig. Black-box testing and auditing of bias in adm systems. *Minds and Machines*, 34(2):15, 2024.

[11] Tobias D. Krafft, Marc P. Hauer, and Katharina A. Zweig. Why do we need to be bots? what prevents society from detecting biases in recommendation systems. In *International Workshop on Algorithmic Bias in Search and Recommendation*, pages 27–34. Springer, 2020.

Not Yet Published Peer-Reviewed Journal Articles and Conference Proceedings

[12] Maryam Haeri Amir, Marc P. Hauer, and Katharina Zweig. Equality of Quality: The Relation between Quality Measures and Fairness Measures for evaluating Machine Learning Models.

Non-Peer-Reviewed Scientific Contributions

[13] Djalel Benbouzid, Christiane Plociennik, Laura Lucaj, Mihai Maftei, Iris Merget, Aljoscha Burchardt, Marc P Hauer, Abdeldjallil Naceri, and Patrick van der Smagt. Pragmatic auditing: a pilot-driven approach for auditing machine learning systems. *arXiv preprint arXiv:2405.13191*, 2024.

[14] Abraham Bernstein, Claes de Vreese, Natali Helberger, Wolfgang Schulz, Katharina Zweig, Christian Baden, Michael A Beam, Marc P. Hauer, Lucien Heitz, Pascal Jürgens, et al. Diversity in news recommendations. *arXiv preprint arXiv:2005.09495*, 2020.

[15] Jan Fiete-Schütte, Susanna Wolf, Marc P. Hauer, and Christopher Koska. Vertrauen im Kontext - Messung und Operationalisierung.

[16] S Hallensleben, C Hustedt, L Fetic, T Fleischer, P Grünke, T Hagendorff, M P Hauer, A Hauschke, J Heesen, M Herrmann, R Hillerbrand, C Hubig, A Kaminski, T D Krafft, W Loh, P Otto, and M Puntschuh. From Principles to Practice - An interdisciplinary framework to operationalise AI ethics. *iRights. Lab, Tech. Rep.*, 2020.

[17] Marc P. Hauer, Tobias D. Krafft, Andreas Sesing-Wagenpfeil, and Katharina Zweig. Quantitative study about the estimated impact of the AI Act. *arXiv preprint arXiv:2304.06503*, 2023.

Book Contributions

[18] Hanna Hoffmann, Verena Vogt, Marc P. Hauer, and Katharina Zweig. Fairness by awareness? Zur Einbeziehung geschützter Merkmale in algorithmische Entscheidungen. In *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, pages 191–220. C.H.Beck, 2022.

[19] Johannes Kevekordes, Marc P. Hauer, and Maryam Amir Haeri. Rechtliche Bewertung möglicher Fairnessmaße. In *Information und Recht 87 (Künstliche Intelligenz - Ethik und Recht)*, pages 141–190. C.H.Beck, 2022.

Non-Scientific Contributions

[20] Rasmus Adler, Nikolas Becker, Georg Borges, Marc P. Hauer, Jens Heidrich, Sven Hilpitsch, Robert Hoffmann, Pauline Junginger, Lisa Jöckel, Michael Kläs, Daniel Krupka, Lukas Martinez, Andreas Sesing, and Katharina Zweig. Abschlussbericht ExamAI – KI Testing und Auditing. Herausforderungen, Lösungsansätze und Handlungsempfehlungen für das Testen, Auditieren und Zertifizieren von KI, 2021.

[21] DIN/DKE. Standardization roadmap of artificial intelligence 1.0. *DIN/DKE, Berlin/Frankfurt*, 2020. Contribution to the ethics working group.

[22] DIN/DKE. Standardization roadmap of artificial intelligence 2.0. *DIN/DKE, Berlin/Frankfurt*, 2022. Contribution to the glossary working group, Vertical topics: Explainability, transparency and fairness.

[23] Marc P. Hauer and Katharina Zweig. Chancen und Risiken algorithmischer Entscheidungen. *Human Ressource Manager*, 01/2021:48–53, 2021.

[24] Lene Kunze, Gertraud Leimüller, Lena Müller-Kress, and Marc P. Hauer. Method handbook: Assurance Cases for fair AI systems. 2023.

[25] Gerhard Runze, Martin Haimerl, Marc P. Hauer, Taras Holoyad, Otto Obert, Henrich Pöhls, Rustam Tagiew, and Jens Ziehn. Das AI-Glossary als Weg aus Babylon - Ein Werkzeug für eine gemeinsame KI-Terminologie. *JavaSpektrum*, 03/2023:42–46, 2023.

[26] Katharina Zweig, Marc P. Hauer, and Franziska Raudonat. Anwendungsszenarien: KI-Systeme im Personal- und Talentmanagement, 2020.

[27] Katharina Zweig, Tobias D. Krafft, and Marc P. Hauer. Dein Algorithmus - Meine Meinung. Algorithmen und ihre Bedeutung für Meinungsbildung und Demokratie. *Landeszentrale für neue Medien*, 2017.

Conference Participations

Aug. 2024 - Okt. 2024 **Symposium on Scaling AI Assessments**, Project Certified AI, Cologne
Member of the program committee, chair of the sessions *Ethics* and *Standards*

Sep. 2024 **Informatik Festial 2024**, Gesellschaft für Informatik, Wiesbaden, Are We Still on Track with Our Responsibility Strategy? - Introducing an Internal Assessment of Corporate Digital Responsibility Engagement
Presentation of our publication

Nov. 2023 **Fueling European Innovation with AI**, Federal Ministry for Digital and Transport, Mainz, Transparent AI - testing standards and education
Panel discussion

Nov. 2023 **TÜV AI Forum 2023**, Digital Technologies Forum, Berlin, Using Assurance Cases to assure the fulfillment of extra-functional requirements of AI-based systems - Lessons learned
Presentation of our publication and progress since then

Oct. 2023 **AISoLa 2023**, Bridging the Gap Between AI and Reality, Crete, A quantitative study about the estimated impact of the AI Act
Presentation of our publication

Oct. 2023 **Software QS Tag**, Sustainable Quality - Software and Systems Quality all Stakeholders can depend on, Frankfurt, Alle wollen {faire, gute, ...} KI, aber wie geht das? Assurance Cases als Methode zur Gewährleistung komplexer Anforderungen von KI-Systemen
Expert talk

Jun. 2023 **re:publica**, Thema: CASH, Berlin, Hacking Explainable AI: Wie KI-basierte Systeme diskriminierendes Verhalten hinter der richtigen Erklärung verstecken können
Expert talks and exchange

Apr. 2023 **7th International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems (ITEQS 2023)**, Co-located with the IEEE International Conference on Software Testing (ICST 2023), Dublin, Using Assurance Cases to assure the fulfilment of non-functional requirements of AI-based systems - Lessons learned
Presentation of our publication

Feb. 2023 **Politische Kommunikation und KI**, Chancen und Herausforderungen für die Regierungskommunikation, Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg
Invited participation to the exploratory workshop

Apr. 2021 **5th International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems (ITEQS 2021)**, Co-located with the IEEE International Conference on Software Testing (ICST 2021), Digitalveranstaltung, Assuring Fairness of Algorithmic Decision Making
Support for the presentation of our publication

Apr. 2021 **GOAL International AI Conference**, Governance of and by Algorithms, Digitalveranstaltung, AI Ethics: From Principles to Practice – An interdisciplinary framework to operationalise AI ethics
Expert talk

Nov. 2019 **Diversity in (News) Recommender System**, Dagstuhl-Perspektiven-Workshop, Schloss Dagstuhl, Algorithmic Accountability and Fairness – A computer scientist's perspective
Expert talk and participation in the interdisciplinary workshop