Data & Policy (2023), 5: e36
doi:10.1017/dap.2023.30

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Overview of transparency and inspectability mechanisms to achieve accountability of artificial intelligence systems

Marc P. Hauer, Tobias D. Krafft and Katharina Zweig

Algorithm Accountability Lab, RPTU Kaiserslautern Landau, Kaiserslautern, Germany
Corresponding author: Marc P. Hauer; Email: hauer@cs.uni-kl.de

Received: 14 December 2021; **Revised:** 14 September 2023; **Accepted:** 22 September 2023

Keywords: AI policy; algorithmic accountability; inspectability; transparency; overview

Abstract

Several governmental organizations all over the world aim for algorithmic accountability of artificial intelligence systems. However, there are few specific proposals on how exactly to achieve it. This article provides an extensive overview of possible transparency and inspectability mechanisms that contribute to accountability for the technical components of an algorithmic decision-making system. Following the different phases of a generic software development process, we identify and discuss several such mechanisms. For each of them, we give an estimate of the cost with respect to time and money that might be associated with that measure.

Policy Significance Statement

Our article supports policymakers, artificial intelligence (AI) system engineers, and operators of AI systems to choose a suitable set of transparency and inspectability mechanisms to render the use of AI systems accountable.

1. Introduction

The European Commission and many non-EU countries around the world are currently working on frameworks to ensure the accountable use of artificial intelligence (AI) systems. This goal of an "accountable use" is often discussed under the term *algorithmic accountability.* Algorithmic accountability concerns all "obligations to justify the use, design, and/or decisions of/concerning an algorithmic system and the subsequent effects of that conduct" (Wieringa, 2020).

In this article, we focus on so-called *algorithmic decision-making* systems (ADM systems). A software is called an ADM system if it scores or categorizes input data, that is, if it assigns a number to the input data and performs a task based on that number (Dubber et al., 2020). While the systems are called "decision-making" systems, most of them are currently used as decision support systems, that is, the numbers computed by the system are then interpreted by a human to make the final decision. But even today, some of the systems make automated decisions, for example, the MIDAS systems was used to automatically detect potential social system fraud and to initiate letters to the potential infringers (Alvesalo-Kuusi et al., 2022). When these systems use behavioral data or other personal data of individuals, their results could infringe fundamental rights and, thus, their use needs to be accountable whenever there is a risk of serious

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

harm (Saurwein et al., 2015; Zweig, 2019, p. 75). Next to those systems that might directly harm individuals, other ADM systems can indirectly induce harm, be illegal or have other, unintended consequences, for example, a drone that inspects the stability of a bridge and judges it incorrectly, might thereby endanger the safety of car drivers without using their personal data, and an incorrect risk evaluation for floodings or other natural catastrophes of an area might devalue the houses there without any substantial reason. Smith (2017) provides a comprehensive list of various kinds of harms of ADM systems. With release of the White Paper on Artificial Intelligence (European Commission, 2020, p. 17) and the so-called AI Act (European Commission, 2021) of the European Commission proposed regulations of AI systems based on the harm that their usage may cause.

The possible consequences of using an ADM system are complex to assess, because harm can originate from multiple sources:

1. Harm can be inflicted by wrong decisions of the system. Wrong decisions can be caused by a variety of problems in the training phase, for example, the choice of the training data and the machine learning procedure.
2. Harm can also be caused by how the ADM system is used (e.g., the decision is not discussed with the evaluated person, it cannot be overruled by a human decider, the human decider is not educated to interpret the decision).
3. Harm can also result from a complex combination of both, the system and its actual use.

Accountability can thus refer to problems with the design of the system, the underlying data, and/or the concrete situation in which the system is used and how its result is used. As long as ADM systems are black box systems and detailed information about their usage is lacking, it is therefore difficult to identify unwanted results and their exact cause in order to make someone accountable for them (Neyland, 2016). To improve this process of identifying problems and finding the person accountable for them, there are two main mechanisms: *Transparency* about both, the ADM system and the details of its usage, and *inspectability mechanisms* that allow third parties to analyze the system's decisions in an experimental fashion (Lepri et al., 2018).

The implementation of all of these mechanisms would ensure maximal transparency and inspectability. However, the principle of proportionality requires a careful consideration of the costs involved in disclosing information (transparency mechanisms) and providing access to the system (inspectability mechanisms). In this sense, this article gives a descriptive overview of possible transparency and inspectability mechanisms for the technical components of an ADM system. To support those that decide about the necessary requirements of transparency and inspectability mechanisms, we also classify their respective implementation into two broad groups: "low cost" or "high cost". We discuss our estimation and contrast it with the general insight the mechanisms allow. This builds the first foundation for legal scholars, politicians, and regulators to make decisions on the proportionality of technical requirements to render the use of AI systems accountable. Based on this broad estimate, the real costs of the requested mechanisms will then have to be refined either within sectors or maybe even for single AI systems before normative requirements are set.

2. Accountable software development process of an AI system

In this article, we focus on accountability in the subfield of AI called ADM-systems containing a learned or learning component (see Figure 1), such as all sorts of recommendation systems (e.g., products, job applicants, medical treatments), credit scoring system and risk assessment system. As a *learned or learning component* we define any statistical model that was built by or is consistently updated by a *machine learning* (ML) procedure—this is the part of the system that is denoted as *AI* in the broad sense. Generative AI systems like DeepFake systems (Westerlund, 2019) for images, and videos or GPT3 (Floridi and Chiriatti, 2020) for texts, are not in the focus of this article, neither are the safety or security

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

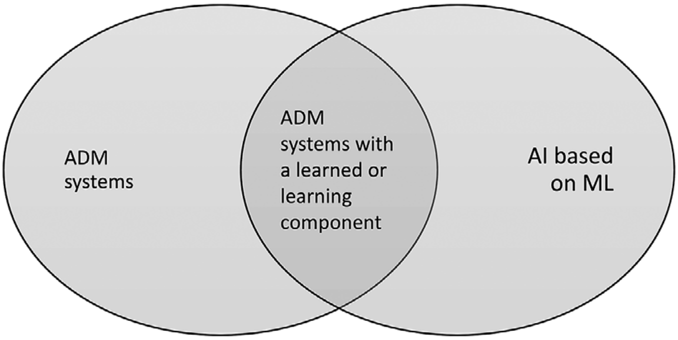

Figure 1. *ADM systems constitute a large group of software systems, including expert systems with man-made rules. When such a system contains a learned or learning component, it is a member of those artificial intelligence systems that are based on machine learning. The article focuses on algorithmic decision-making systems with a learned or learning component. Figure by Algorithm Accountability Lab (Prof. Dr. K. A. Zweig)/CC BY.*

aspects of complex AI systems like autonomous cars, which normally consist of multiple ADM systems whose complex interaction might create emergent behavior (Zweig et al., 2021, p. 63).

The question of how an operator or developer of an AI system can be held accountable is much debated. Various authors propose algorithm accountability frameworks that greatly differ in aims and scopes. Raji et al. (2020) present internal algorithm audits, performed by external experts, as an approach to check whether ethical expectations are met. Felzmann et al. (2020) propose a transparency by design approach as a software development concept that aims to help engineers in developing accountable systems from the start. In this article, we follow a similar idea as Felzmann et al. (2020), but build our elaborations along a generic software development process. Naja et al. (2021) also follow such a process, but elaborate on an ontology-based approach by building a knowledge graph based on numerous relevant questions for supporting accountability of AI systems and procedures of auditing them. They clearly focus on the methodology itself and less on the questions to be answered. To derive the *questions to be answered*, that is, the relevant information and accesses for achieving accountability, we follow the accountability theory of Mark Bovens. Bovens defines accountability in the public sector as a relationship between *an actor* and *a forum* in which the actor has to *explain and justify* its posture (Bovens, 2007, p. 450; see Figure 2). Then, the forum should have the right to *challenge the operator's statements*, for example, by posing questions (for clarifications, and additional explanations) and *passing judgment.* Finally, "the actor may face consequences." Wieringa transferred Bovens' accountability theory to the use of AI. Especially, she states that now there might be multiple actors, among which the operator and/or the developer of the AI system are the most prominent. Similarly, the forum can also be instantiated by

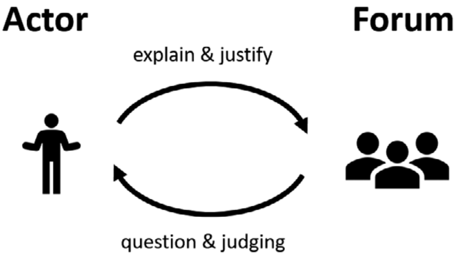

Figure 2. *Visualization of the accountability process according to Bovens (2007, p. 450). Figure by Algorithm Accountability Lab (Prof. Dr. K. A. Zweig)/CC BY.*

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

different groups, for example, a controlling authority, NGOs, the society at large, and/or consumer protection agencies (Wieringa, 2020).

While personal discussions between actor(s) and forum(s) are theoretically possible, they will be impractical in most cases, especially in an agile development process with fast updates. One way to deal with that problem is to take an asynchronous approach in which the actor(s) explain their design choices and the way the results are obtained and used. All kinds of static information are referred to as *transparency mechanisms* in this article. Such mechanisms can generally be realized by publishing the respective information in form of textual or tabular documents, but for some information there also exist more sophisticated methods that are mentioned later in this article. In general, transparency allows the actors to explain and justify certain decisions of the past. It also allows the user of an ADM system to explain and justify its usage in a given application scenario.

Definition 1 (Transparency mechanism). A transparency mechanism is the disclosure of any information regarding the development or usage of an ADM system, such that the forum is then enabled to control whether, for example, the design decisions are state-of-the-art and suitable, and whether the system is used in practice as described in theory. They can also contain information that can only be verified by an *inspectability mechanism* (see Definition 2).

By implementing transparency mechanisms, the forum is then enabled to check whether, for example, the aimed and claimed properties of the system are actually implemented and whether the system is used as described. However, the possibility to pass appropriate judgment based on transparency alone is very limited (Citron and Pasquale, 2014; Kroll et al., 2016; Ananny and Crawford, 2018): The complex working of an AI system cannot be understood just by disclosing information about it. An important misunderstanding is that access to the source code of the system would be necessary or sufficient to understand its behavior. First, it is important to understand that there are two types of source code (see Figure 3):

1. Source code I: The code of the machine learning procedure that gets data as an input and computes some statistical model based on that information. This includes the untrained model structure and the learning method (e.g., supervised, unsupervised, or semi-supervised approach);
2. Source code II: The statistical model which is then embedded into code that allows to run new input data and compute the most likely classification, score, or rank.

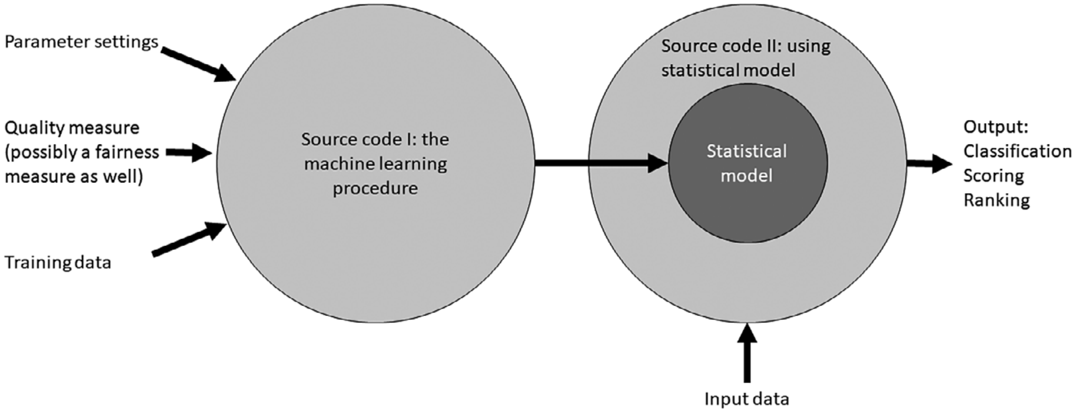
Parameter settings
Quality measure (possibly a fairness measure as well)
Training data
Source code I: the machine learning procedure
Source code II: using statistical model
Statistical model
Output: Classification Scoring Ranking
Input data

Figure 3. *ADM systems are based on two different source codes: The first one computes a statistical model from input data and the second one uses this statistical model to compute a classification/score/ranking for new data. Figure by Algorithm Accountability Lab (Prof. Dr. K. A. Zweig)/CC BY.*

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

The first code normally contains a set of parameters that determine, how exactly the statistical model is computed based on the training data. It is furthermore guided by the evaluation of a quality measure and sometimes its “fairness” is additionally assessed by a so-called fairness measure. *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 they should be or compare the systems output distribution with regard of subgroups (Verma and Rubin, 2018). In most cases, parameter settings and quality/fairness measures are not contained in the code, but are external information. This is one reason why the source code is not sufficient. The statistical model, contained in the second source code, is then the result of a complex interaction between the first code, the parameters, and the input data. For example, for some machine learning procedure, the order in which the training data is fed to the system will change the resulting statistical model (Lopes et al., 2017, p. 621). Similarly, the choice of a specific exit criterion for the training process or combination of multiple exit criteria, like quality and fairness measures, heavily influence the resulting statistical model. This is a second reason, why neither the first nor the second source code are sufficient on their own. Last but not least, in most cases, it is not necessary to inspect either of the codes. Instead, the resulting statistical model can be seen as a black box and experimentally examined by creating artificial input data sets and observing the resulting decisions (Diakopoulos, 2016; Kemper and Kolkman, 2019).

Definition 2 (Inspectability mechanism). Any mechanism that allows a third party to explore the behavior and properties of an ADM system, including its constituent parts (the training data, test data, the machine learning method, and the actual input data), is called an *inspectability mechanism* in this article. It is only called an inspectability mechanism if it allows the forum to “ask questions” and “judge” properties of the ADM system and its constituent parts. For example, by accessing the system with an artificial input data set with known classifications via API, the claimed quality of a classification system can be tested. Thereby, the forum can asynchronously assess the claims made by the actor.

In summary, transparency and inspectability mechanisms can help to establish a large part of the accountability process in the design and usage of an ADM system in an asynchronous, scalable fashion (see Figure 4).

However, the development and the usage of an ADM systems follows different phases with probably different actors in each phase. In general, decisions made by actors in a previous phase may be of interest to actors of all subsequent phases and finally, to society. Thus, transparency and inspectability mechanisms can often be assigned to the different phases of this process, with possibly different actors and forums for each of them. This article follows the development process outlined by Zweig et al. (2018) (see Figure 5) and is compatible with other software development process models like CRISP-DM (Azevedo and Santos, 2008) but also with Wieringa’s explanations regarding ex ante, in medias res and ex post considerations (Wieringa, 2020, p. 7).

In the model we differentiate the following eight steps:

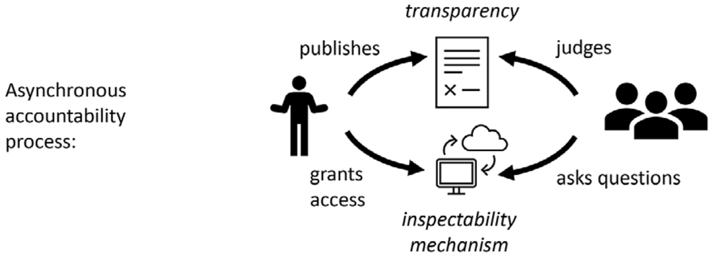

Figure 4. *Transparency about past decisions and actions plus access to inspectability mechanisms help to establish an asynchronous accountability process between different actors and forums. Figure by Algorithm Accountability Lab (Prof. Dr. K. A. Zweig)/CC BY.*

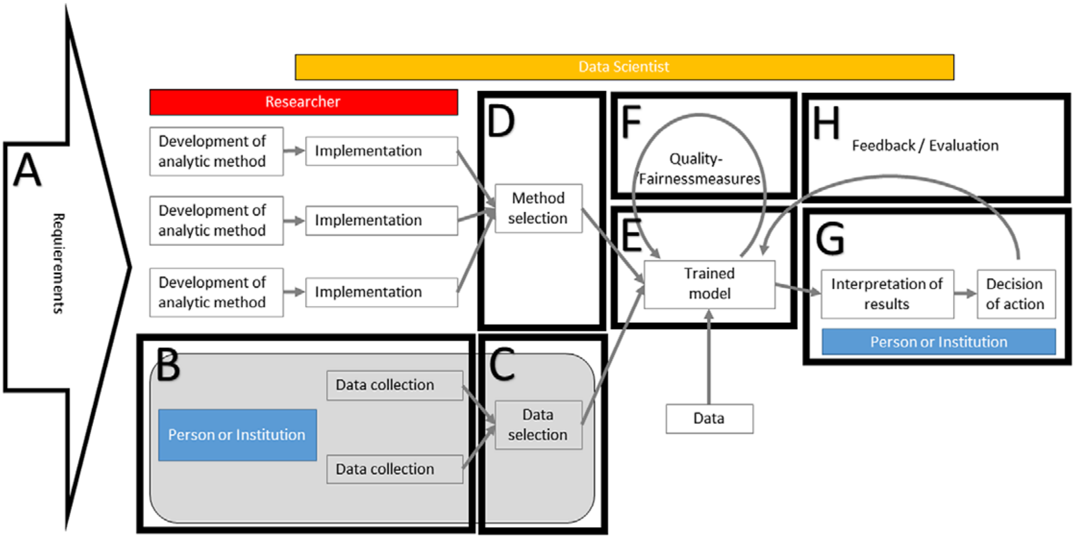

Figure 5. *The long chain of responsibilities according to Zweig et al. (2018). Figure by Algorithm Accountability Lab (Prof. Dr. K. A. Zweig)/CC BY.*

1. Phase A addresses the *requirements engineering*. It contains information on what exactly is to be achieved with the ADM system, what the (informal) target criteria are, a benefit and risk assessment and many more information regarding the requirements.
2. Phase B addresses the *data collection* procedure, which already poses a huge risk of introducing errors and bias if not done properly.
3. Phase C addresses the *training data set construction* step, which aims for preparation and cleaning.
4. Phase D addresses the *choice of machine learning procedure and parameters*. This also includes the used tool(s). The choice of machine learning procedure and the creation of the training dataset are interdependent. One can either choose a method for which the data is suitable or which requires little preprocessing, or design the details of the preprocessing with the chosen procedure in mind.
5. Phase E addresses the details of the *learning procedure*, including the quality and/or fairness measures used for training.
6. Phase F addresses the installed *quality assessment*. This includes the data used for testing, the overall performance of the trained model, and the conditions under which tests are performed.
7. Phase G addresses *the system usage in an application scenario*. This is about application-specific requirements that were not available when the system was developed.
8. Phase H addresses the *evaluation of the ADM system in an application scenario* which may specify an environment and quality requirements, which were potentially not yet (precisely) known in phase F.

In the following, we present an extensive overview of all possible mechanisms of transparency and all possible mechanisms that provide inspectability of an ADM system and its usage and assign them to the different phases of this simple development process. This list can be used as a basis for selecting the most useful mechanisms to enhance accountability for any individual ADM system development.

3. Possible transparency and inspectability mechanisms in the development of an ADM system

The following sections trace the different phases in the development process of ADM systems as defined above and describe the possible mechanisms for transparency and inspectability in detail, that is, for each phase,

1. we describe a set of information that could help forums to identify accountable actors and possible problems (*transparency mechanisms*) and
2. we describe possible mechanisms (e.g., processes and interfaces) that allow a third party to explore, change, interact with the ADM system or its usage to better understand the consequences of using its decisions (*inspectability mechanisms*).

Each section starts with a short description of the corresponding phase. Then, the respective mechanisms are discussed with regard to the accountability they establish, that is, we name the mechanism, the actors, and what kind of insight the mechanism might yield for a forum. In the following, transparency mechanisms are labeled by Arabic numerals (A1, A2, A3…) and inspectability mechanisms by Roman numerals (I, II). We do not yet discuss the forums: To whom an actor is accountable is very much situation-specific, might be given by law, and depends on the rights of actors to keep certain decisions and systems private. In general, any person or organization affected by or charged with the oversight of a system is a potential forum. It is thus the prime focus of legal institutions to assess which information and mechanism must be accessible by which forum to allow for accountability. We start with phase A, the *requirements engineering.*

A. Requirements engineering

Before any software system is built, it is important to understand what functions it should offer to its users and what other requirements need to be met (e.g., energy consumption, legal requirements)—the latter are often (unintuitively) called *nonfunctional requirements.*[1] Requirements engineering is a systematic approach to identify, specify, and manage those requirements; it aims to understand the customers' and future users' needs to provide a satisfying product (Glinz, 2011). Its results are requirement documents that describe the specific properties of all functional and nonfunctional requirements. The requirements may also include an application scenario or a set of application scenarios; they determine the environment in which the system is to be embedded. For ADM systems, the exact circumstances in which they are used can be especially important as they might influence the required quality of the system as illustrated in the following examples:

1. A face recognition system which is used in a company to grant access to 50 people, based on video material in a very well lit space, can in most cases not be used to search for criminals based on video material from public spaces with very different lighting.
2. If the system is used as a *decision support system*, that is, a human is supported by the machine, but makes the decision himself, the quality acceptance threshold might be lower than if the system automatically triggers an action without human supervision (these systems are called *automated decision making systems*) (Wagner, 2019).
3. A generalized recommendation system might also be subdued to very different fairness requirements based on the products or services it recommends: Recommendation of people on a career platform to employers might legally require that people of the same education are recommended regardless of their gender, religion, or ethnicity, while recommendation of household items is much less regulated.

The application scenario can also determine other aspects of the software which need to be detailed as software requirements:

1. If the application context demands explainable decisions of the ADM system, such requirements need to be specified. Some machine learning models suffer from a lack of explainability (e.g.,

[1] See ISO 25010—Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE) —System and software quality models.

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

artificial neural networks (Rudin, 2019)). There are attempts to achieve explainability post hoc (e.g., with techniques like LIME (Ribeiro et al., 2016), or Anchors (Ribeiro et al., 2018)), though this is an ongoing research field currently posing several limitations (Rudin, 2019). If such explainability is necessary, models that are not considered inherently explainable should not be used (Rudin, 2019). Note that the term explainability is controversial, as it is often unclear what exactly is meant by it. In many cases, it concerns the question of which input values were decisive for an output

2. There is also the question of whether the final product should be able to continue learning in use. If this was not planned by the developers, but the system is used that way in practice, problems are likely to arise (Kroll et al., 2016, p. 660).

In the following, we discuss three possible transparency mechanisms, concerning the results of the requirements engineering process.

A.1 Disclosure of the application scenarios

1. *What is made transparent?* The set of application scenarios for which the system is suitable together with the assumptions they are based on.
2. *Actor:* There are two kinds of software: customized and generalized software. In a customized software, the customer sets the application scenario and helps to identify all subsequent requirements together with the software provider. For a more generally applicable software, the software provider identifies the requirements.
3. *Explanation/Justification:* Transparency about application scenarios enables third parties to evaluate whether a given ADM system is used within them or for other application scenarios that might not fit the identified requirements based on the ones used in the design process (Raji et al., 2020).
4. *Cost*: The are no considerable costs for publishing information about those application scenarios for any given ADM system that were taken into account.

Next to the application scenarios and their basic assumptions, the full requirements documents could be published as discussed in the following.

A.2 Disclosure of requirements documents

1. *What is made transparent?* Publishing of all or parts of the requirements documents and the actors that identified and detailed the requirements.
2. *Actor*: The people involved in the requirements engineering process.
3. *Explanation/Justification*: Transparency of the requirements documents allows for the identification of the accountable actors and multiple types of controls:
   a) Are the requirements complete or are important requirements missing, for example, was a quality measure identified?
   b) Are the requirements adequate, for example, is the threshold for the quality measure sufficient for the given application scenario(s)?
   c) Are there requirements addressing a quick reaction to errors and problems or possibilities for people affected to report such (sometimes referred to as *risk mitigation plan* (Raji et al., 2020))?

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

   d) Together with the appropriate inspectability mechanism, it can be controlled whether the requirement was actually satisfied, for example, is the quality of the system at least as good as the predefined quality threshold?

4. *Cost*: The costs can range from almost zero for the publication itself to considerable costs, depending on whether the developer already has a well-established requirements engineering process that results in easily understandable documents. The publication of these documents can result in indirect costs: often, they allow insights into the system that might, for example, make it amenable for hacking or manipulation. It can give insights into the inner workings of the company. It can reveal trade secrets. Thus, in most cases, any reasonable legal requirement will only identify a fraction of the documents to be published—this induces costs for selecting those parts that need to be published. In the same vein, the access to this information might need to be secured which incurs further costs.

For any given usage scenario, there is often an overarching goal of using the ADM system. In many cases, the ADM system is believed to be faster, more efficient, less expensive, and/or better in decision-making. For example, in Austria, a system was trialed to predict the risk of future unemployment for currently unemployed persons. Here, according to the software company who built the system, the overarching goal is to minimize the cost for further education and to maximize the number of people in the job market (Holl et al., 2018), while the ADM system itself is evaluated by the number of correct predictions. It can be assumed that the operator of an ADM system has an evaluation process together with criteria for success to quantify or qualify whether this goal is reached. To make the usage of an ADM system accountable, the overarching goal and its evaluation process can also be made transparent:

A.3 Disclosure of the goal of using an ADM system

1. *What is made transparent*? Publishing the overarching goal of using an ADM system, the evaluation process and success criteria by which the operator of an ADM system judges whether the goal of using the ADM system is reached.
2. *Actor*: The operator of the system.
3. *Explanation/Justification*: With this information, forums can judge whether the overarching goal is consistent with, for example, societal goals or the requirement documents. It can be judged whether the evaluation process is able to identify a successful use of the system and whether additional evaluation processes are necessary to identify unwanted consequences of the usage of an ADM system. Together with inspectability mechanisms, third parties can also check whether the usage actually is successful.
4. *Cost*: Publishing of the overarching goal is on the one hand cheap, however, on the other hand, the reaction of a forum (e.g., the general public) can be costly. The operator of the Austrian unemployment risk assessment software (AMAS) had multiple public discussions after activist groups[2] and scientists (Allhutter et al., 2020, p. 7) deemed the software to be discriminating.

There is no inspectability mechanism associated to this phase that we are aware of. The next phase concerns the data collection from which the training data is later built.

[2] For example, the 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/ (accessed 8 October 2021).

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

B. Data collection

ADM systems with a learned component are only able to derive good decision rules based on a high quantity of high-quality data (Cortes et al., 1995; Roh et al., 2019). Their collection is the first step that influences all further steps in the process. Since it is often done by different actors than those that select and preprocessed it into the training data, we decided that it justifies its own phase. Data collection underlies several legal constraints; additionally, the question of whether they can be used in a given ADM system is strongly dependent on the application scenario. For example, the face recognition software in a company that grants access to restricted areas needs to learn from multiple images of all employees. The data set does not need to be balanced to represent all ethnicities in a fair manner; the face recognition of criminals that someone wants to use in a public space does require a diverse and balanced data set to ensure comparable false-positive rates over all relevant groups. Furthermore, different application scenarios might require different levels of acceptable error rates in the data. These are just some examples that highlight the importance of this phase. Therefore, accountability of this process is necessary for all subsequent steps in the development and usage of an ADM system. The *data science and AI in the age of COVID-19* report, published by The Alan Turing Institute, even suggests that multiple scientific challenges that dealt with the pandemic could have been reduced by such mechanisms.[3]

The first transparency mechanism concerns the way the data was collected.

B.1 Disclosure of how, when, and what data was collected

1. *What is made transparent*? Information about how, when and what data was collected.
2. *Actor*: The collectors of the data set. The buyers of the data set.
3. *Explanation/Justification*: With explanations about how, when and what data was collected, the forum can judge whether the data is in principle valid as a basis for the ADM system. For example, the forum can judge whether the software development team fell for the *availability bias* (data has not been collected because it seems to be most appropriate for building the ADM system, but because it is available) (Baer, 2019, pp. 19–23) or whether there might be a *conceptual bias* in the data (the method of data collection has a direct influence on the data, for example, if the data is collected under model assumptions) (Baer, 2019, p. 74). The forum can also judge whether the data is too old; especially, when short-term societal changes occur (such as the corona crisis), it is possible that the data collected beforehand cannot be used to build a useful statistical model for future decisions. This perspective is being researched under the term concept drift (Webb et al., 2016).
4. *Cost*: If data is specifically collected for the task at hand, documenting the collection process does not involve any significant costs. Many data sets are little documented and nonetheless used in multiple contexts. A post-documentation is very often not possible and any requirement on the publication of the data collection, like data protection regulation, might essentially render the data set useless. Other data sets are commercially available, but the producers might not be willing to share the specifics about their collection for various reasons. It can be assumed that over time, if publication of the data collection process is required for many ADM systems, a market will emerge in which data sets become available together with a reasonable description of their collection process. However, until then, it might be very costly to fulfill such a requirement.

[3] https://www.turing.ac.uk/sites/default/files/2021-06/data-science-and-ai-in-the-age-of-covid_full-report_2.pdf (accessed 28 September 2021).

While it is already useful to understand how and when the data was collected, some of the measures might be proxy measures for something that is not easy to quantify. For example, when is an unemployed person successfully integrated into the job market? In which time frame does she need to find a job? For how many days does it need to last? This type of quantification of complex terms is called *operationalization*, that is, the process of making properties and concepts measurable (Burnette, 2007). In essence, all quality and fairness measures are themselves the result of an operationalization. For both, more than two dozens measures are known. 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 the data collection, transparency about them is helpful:

B.2 Disclosure of the operationalizations

1. *What is made transparent?* The way in which complex terms are measured, based on an operationalization and why this specific operationalization was chosen.
2. *Actor*: The person(s) which conducted the operationalizations or chose one of the existing ones and the person(s) that decided that the operationalizations are reasonable.
3. *Explanation/Justification*: The forum can judge whether the chosen operationalization captures the most important aspects of the term to be quantified.
4. *Cost*: The cost of the publication is low. Formulating description of operationalizations that are comprehensible to external parties can be difficult, depending on the complexity.

In some cases, a forum might actually want to look into the process of data collection, that is, look into the code of digital data collection or inspect manual data collection processes. If the data is collected digitally, the code that collects, operationalizes, and labels the data, including all related information (e.g., documentation), could be published. Major challenges of making code available to external entities are the protection of trade secrets and the question for how often the accessible code needs to be updated. Similar problems have already been solved for customers of software that need access to code.

In general, there are multiple ways of granting access to code. Either, someone physically visits the operator to view the code on the spot, or the code is made available remotely. The latter can be done by handing out the code files or by granting access to the development system or the version control system. If code is handed out physically, this can also be mediated, for example, by a lawyer. To do so, the company uploads the current code version regularly (e.g., with every major update) to a server to which only a lawyer has access. An analogous concept can be applied to provide a controlling entity with the code. This ensures that business secrecy and data protection are maintained and that a sufficiently up-to-date version is available without direct access to the interim variants of the code. Obviously, the effort to grant access to code is in this case directly proportional to the number of necessary code updates. If direct access to a versioning system is granted, there is no cost per update, but other costs need to be taken into account, for example, security of access to the system for external users.

For all manually performed activities, inspectability about the process (B.1) can only be achieved by some kind of process audit, performed by a third party. This requires a human auditor to observe the manually performed data collection process and to publish the documentation. Process auditing is a complex field on its own, which is why we omit this discussion here and refer to further literature (Hoyle, 2009; Russell, 2010).

C. Constructing the training data set

To turn the collected data set(s) into a training data set, different steps have to be conducted:

1. Identifying the output variable in the data set. The output variable is the number to be predicted by the ADM system. 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. This process is called *labeling*, that is, each input data is assigned to its class or is assigned a score/rank by humans from which the machine then learns the relevant patterns.
2. *Feature engineering*, that is, computing 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. *Selection* of those variables in the data set that are most likely related to the output variable (García et al., 2016).
4. Further *data preprocessing*, that is, dealing with input data that contains missing data (missing values imputation) or wrong data (noise treatment) (García et al., 2016).

Sometimes, the output variable to be predicted by an ADM system is part of the collected data (or can be computed ad hoc), sometimes it is not. In the latter case, the data needs to be *labeled.* There are various labeling processes that range from manual labeling (Taylor et al., 2003) over crowd-sourced techniques (Snow et al., 2008; Chang et al., 2017) to fully automated approaches (Agichtein et al., 2006). For example, one data set uses the tags that users gave to photos on flickr (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, use user behavior to classify data, 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 assumptions and qualities. When the label is part of the data collection itself, how and when it was observed is already part of B.1. If not, one further transparency mechanism would be to publish how exactly the label was assigned to the data: 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, 2020, p. 99):

C.1 Disclosure of the labeling process

1. *What is made transparent?* The labeling process, that is: Who labeled the data? How were these persons trained? Were the persons representative of the population? Were there test data to be labeled by the persons such that errors in labeling could be identified or did multiple persons label the same item? What happened with items with multiple labels? This is just a subset of the most important questions around the labeling process that could be published.
2. *Actor*: The persons organizing the labeling process.
3. *Explanation/Justification*: The forum can judge to what extend the labels of the input data are truthful based on the description of the labeling process.
4. *Cost*: The costs are comparably low but publication of the details might result in social pushback; for example, the label assigned to unemployed persons in the AMAS algorithm were socially strongly discussed and partly condemned (Allhutter et al., 2020).

The selection of the most important variables from one or more data sets is the next step in the data preprocessing. It can be made transparent to enable the forum to judge whether all variables are likely to be causally related to the output variable and whether the most important likely causes are included in the data set.

Sometimes, there is also a change of the selected variables. For example, the values can be normalized between 0 and 1, and a classification label might be assigned a number. More complex constructions of new variables, for example, by multiplying one or more variables or by clustering them, are termed *feature*

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

engineering (Heaton, 2016, p. 1). The newly created variables (or *features*) are then added to the training data set. Once all variables in the data set are determined, transparency about the included variables can help the forum to judge the overall suitability of the training data:

C.2 Disclosure of why which variables were included in the training data set

1. *What is made transparent?* Information about how variables were engineered from the "raw" data set plus which and why variables were selected for the training data set.
2. *Actor*: The persons constructing the training data set.
3. *Explanation/Justification*: The actor(s) justify why they think which variable could influence the output variable. The forums can evaluate the appropriateness of the selection.
4. *Cost*: As a short explanation for each variable should be sufficient, we estimate the cost of this mechanism as very low. Optimally, these explanations are documented anyway during the development process and only need to be translated into a format comprehensible for the forum.

Data preparation addresses techniques to deal with imperfect data. This includes, for example, the imputation ("guessing") of missing values and noise identification. Since the quality of the results of ADM systems strongly depends on the data quality, any impairment of this, for example, by noise,[4] must be taken into account (Frénay and Verleysen, 2013). Thus, many approaches have been developed to tackle the problem (Luengo et al., 2012) which all have their pros and cons, like artificially creating values based on statistical evaluation of other values, or completely removing instances with missing values (Little and Rubin, 2019). Data reduction addresses techniques to increase the information density inherent to the training data, like for example a targeted feature selection to identify and remove redundant information (Hall, 1999), which would only introduce a source of error, for example, by inducing spurious correlations. Since the methods decide about the quality of the training data set, transparency about their exact usage (C.3) and the statistics of the data before and after the preprocessing (C.4) helps to judge the quality of the resulting data set (DIN, 2020, p. 85):

C.3 Disclosure of the preprocessing techniques

1. *What is made transparent?* All decisions around missing and wrong values, that is, the number and type of input data deleted from the training data set plus all methods in detail that input data where it was missing.
2. *Actor*: The curator of the data set.
3. *Explanation/Justification*: The forum can judge how many input data were deleted and whether the deletion renders the data set imbalanced and thus invalid for the task. Similarly, the forum can judge whether the imputation method is suitable for the ADM system's application scenario(s).
4. *Cost*: The costs are comparably low. In many cases, all cleaning steps are contained in one rather small script which is easy to understand and follow. Thus, it might actually be the easiest and simplest way to publish the cleaning code.

[4] According to Ray J. Hickey, noise is defined as anything that obscures the relationship between the features of a data point and its label (Hickey, 1996).

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

Properties of the training data do not need to be limited to the selection of variables and the preprocessing, but may also imply aggregated information like distribution:

C.4 Disclosure of the training data properties

1. *What is made transparent?* Information about the resulting data, for example, number of data points, number of data points for various groups of interest (e.g., based on gender, ethnicity, age), statistics of the distributions of values for each variable like the mean, the standard deviation, and others. This information can be provided in the form of a spreadsheet or a database (Diakopoulos, 2019). 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 form of a so-called datasheet for datasets. This allows for disclosure of information without actually publishing potentially protected data. Next, to aggregate statistics, the rates of missing and possibly wrong data for each variable before preprocessing 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 input data.
2. *Actor*: The persons involved in data collection and preprocessing.
3. *Explanation/Justification*: With such information about the collected data, the forum can judge whether the quantity and quality of the training data is high enough for an ADM system later used in a specific application scenario.
4. *Cost*: The cost to publish the statistics, for example, per datasheet, is low. There already various paper that can be used for reference what a datasheet for datasets may look like, for example, Song et al. (2022) provide a datasheet in their supplementary files.

However, the pure information about the data might not be enough to make the actors accountable. In some cases, it might be necessary to give third parties access to the data so that they can answer question about the data's quality themselves as, for example, demanded by Algorithm Watch[5] or the German standardization roadmap on AI (DIN, 2020, p. 83).

I Full access to the data

1. *What is made inspectable?* The data, either fully or in parts, including all information necessary to understand it. It might be that the data is not directly assessable, but aggregate questions about groups of data can be asked; this might be a measure to reduce the risk of deanonymization (Ohm, 2009, p. 1751), for example, it is possible to ask the mean value of a variable (e.g., the town, instead of the specific address).
2. *Actor*: The persons involved in data collection and preprocessing.
3. *Explanation/Justification*: Partial or full access to the data can strengthen the judgment of the forum on the suitability of the collected data for learning the wanted information from it. Especially, 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 by

[5] 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(accessed 16 September 2021).

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

this access. Additionally, this 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.[6]

4. *Cost*: The costs are considerable. Data is often a valuable asset for a company that could bear other economic opportunities. Furthermore, many of the data are restricted by law in their use. Identifying which of the data can be published or made accessible in which way is likely to be costly, as well as the implementation of an access system. If a direct publication of the data is not advisable, a system can be built, that allows to ask for aggregate statistics of groups of input data, for example, the mean salary of women versus men in a financial data set (Michener and Bersch, 2013, p. 37). 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. To check for this possibility and to produce a system that prohibits deanonymization is likely to increase the costs.

With regard to accountability, this mechanism allows the forum to asynchronously ask questions regarding all information made transparent in phases B.2 and C.

D. Choice of machine learning procedure and parameters

In the development process of an ADM-system, multiple machine learning procedures might be tried out until one settles for the one with the best quality measure value. Most of these methods come with a handful of parameters, for which several values are usually tried before settling with the best combination, as again, evaluated by the quality measure. For reproduction of the result, one important information is which implementation of the machine learning procedure was used—in most cases, one of the many, freely available software packages will have been used. Common tools are, for example, Keras,[7] KNIME,[8] and RapidMiner.[9] Very rarely, software development teams will implement any of these methods from scratch. All of this information: The method, its implementation and the settings for all parameters are relevant, if the training process needs to be assessed and/or recreated (DIN, 2020, p. 85). Mitchell et al. introduce a concept called “model cards” to disclose such information (Mitchell et al., 2019).

D.1 Disclosure of the procedure, its implementation, and the parameter settings used

1. *What is made transparent?* The exact choice of the machine learning procedure including the origin of the code (e.g., which software package was used in which version) plus all settings for parameters and the interpretation of outputs.
2. *Actor*: The data science team training the model.
3. *Explanation/Justification*: Determining which method yields the most accurate predictions is often a matter of trial and error, but for some situations there are plain wrong choices. For example, a very simple regression model might make assumptions about the form of the data that are not met. On the other hand, a very complex machine learning procedure might require more data than are available in the training data. Another example is given by unbalanced data,

[7] https://keras.io/ (accessed 19 October 2021).
[8] https://www.knime.com/ (accessed 19 October 2021).
[9] https://rapidminer.com/ (accessed 19 October 2021).

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

that is, data that contains more information about one subgroup of, for example, people than other subgroups. Here, certain methods perform clearly better than others, as thoroughly investigated by Haixiang et al. (2017, p. 225). Combined with the information from B.2 (collected raw data) or better C.2 (preprocessed data) it can be evaluated whether the properties of the available data might be sufficient for a chosen procedure or not (Cheng et al., 2018). The machine learning procedure also reveals how easy it is for humans to understand how the decision is derived by the machine, based on the training data; that is, the explainability of the model can be judged and compared with the requirements stemming from the application scenarios.

4. *Cost*: The costs for publication of these information are very low, though providing a comprehensible description might be challenging.

E. Training the system

When the training data has been constructed and the machine learning procedure has been decided upon, there are still some decisions left to the data science team: 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. How the data set is divided into these parts is an important decision. Furthermore, some methods are order-dependent, that is, 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). All of these decision can be made transparent in order to allow for a recreation of the machine learning procedure.

E.1 Disclosure of training details

1. *What is made transparent?* The details of the learning procedure, for example, ratio of training to test data or order of the data.
2. *Actor*: The trainers of the system.
3. *Explanation/Justification*: This information is important if the forum wants to reproduce the training on the training data. To be useful, they also need access to the data.
4. *Cost*: While the cost of publication seems to be low, the documentation of the exact learning procedure might be considered part of the trade secret of a data science company and thus be very valuable.

F. Quality assessment

The quality assessment phase is here restricted to the assessment of the system itself, that is, to the question of how many errors it makes. Ethical reviews and social impact assessments, for example, in the form of audits, are also part of a thorough quality assessment (Raji et al., 2020). The phase H describes the quality assessment of the usage of the system, that is, whether the overarching goal is actually met.

In the development process of an ADM system, there are multiple cycles of training the system, evaluating it, and changing parameters or switching the machine learning procedure, when the result is not satisfying. The evaluation optimally takes place with the test data the system has not yet seen before.

While the information about the evaluation metrics has already been discussed in A.2, now it is time to reveal their results (DIN, 2020, p. 85):

F.1 Disclosure of the results of all evaluation metrics

1. *What is made transparent?* The results of all evaluation metrics mentioned in the requirements document and any further evaluation metrics deemed necessary later in the process. All information relevant to assess the meaning of the metrics, like details about the test data and test conditions.
2. *Actor*: It depends on toward whom accountability is to be established. On the one hand, the software team justifies the quality of their system to the customer. On the other hand, in our view, the operator of the system is the person responsible for deciding whether the system's quality is good enough for a given application scenario. Thus, both are actors in different accountability processes, based on the same kind of information.
3. *Explanation/Justification*: In the first scenario, the customer (who might also be the operator) can judge whether the system meets the quality requirements. In the second scenario, third parties can judge whether the system's qualities in a given application scenario are sufficient.
4. *Cost*: The costs of publishing metrics are very low. However, the effort to decide which metrics are most valuable, calculate them, and set acceptable thresholds can be very high (Hauer et al., 2021).

At this crucial phase, the ADM system is handed over to the customer/operator and the system starts to be applied. While information about its quality is helpful, the forum(s) might want to actually question that information and control for themselves (Citron and Pasquale, 2014). This could be achieved by giving access to the ADM system such that it can be tested with a test data set for which the true labeling is known:

II Full access to the output of the ADM system for any specific input data

1. *What is made inspectable?* The forum needs unrestricted access to the system to feed it with input data and access to the resulting output of the input.
2. *Actor*: The operator of the ADM system.
3. *Explanation/Justification*: Claims about the quality of the system can thus be validated. With carefully crafted input data, new questions can also be answered, for example, counterfactual questions like: If this input data said it came from a 23 year old versus a 65 year old, is the decision different? If the operator did not provide fairness measure results, they can be produced by this access for any wanted subgroups of input data.
4. *Cost*: The costs for this mechanism can vary strongly, but can be estimated to be rather high. In cases, where such an access is justifiably required by legal authorities, it is likely that the ADM system is used in a sensitive application scenario. Thus, unrestricted access might not be advisable. Setting up a protected access for accountability forums is likely to drive the costs. Furthermore, extensive inspection of the ADM system might reveal its inner workings so much, that one of the persons with access could rebuild a similar system which could infringe trade secrets. Others could learn how to game the system, that is, how to change input data to achieve a specific decision from the system. All in all, the implementation and maintenance of this mechanism are likely to be very costly for the operator.

Without question, this access for (selected) third parties is the most valuable in assessing the suitability of an ADM system, however, it is also the one whose requirement needs to be carefully justified. With regard to accountability, this mechanism allows the forum to asynchronously ask questions regarding all information made transparent in phases D–F.

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

G. Using the system in an application scenario

The usage of an ADM system in a specific scenario might also be bound to some requirements, for example, those that the designers of the software deem important. For example, the developers of the AMAS required a set of rules addressing the social compatibility (orig. *Sozialverträglichkeit;* Holl et al., 2019), for example, that any algorithmic output is only meant to support the decision of a human; that the person judged by the system can object to its decision; that the person can also see his or her input data and correct them, if necessary. Additionally, the operator might have some procedural requirements, as the operator will usually not apply the ADM system him- or herself. For example, ADM systems operated by the state will in general be used by civil servants. In such cases, the operator might set some internal rules about the usage. For third parties, it is valuable to know more about such requirements to check whether they are met in practice:

G.1 Disclosure of the procedural requirements in the usage of an ADM system

1. *What is made transparent?* All rules or requirements around the use of an ADM system, for example, those concerning how the actual data is collected, whether there are humans in the loop and how they can overrule a decision by the system, how persons can object a decision.
2. *Actor*: The operator of the ADM system.
3. *Explanation/Justification*: The forum can judge whether the rules and requirements are suitable for the application. The forum can use the disclosed information to check whether the rules and requirements are actually followed in practice.
4. *Cost*: The direct costs related to this mechanism are rather low, but might also be rather sensible in most cases.

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

H. Evaluating the ADM system in an application scenario

In most cases, the operator of the ADM system will have an overarching goal to use it. It can be assumed that this is also associated with an evaluation process of whether the ADM system is actually able to satisfy this goal. In some circumstances, for example, if the operator is the state (Krafft et al., 2020, p. 11) 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 procedure(s) (H.1) and/or their results (H.2) (Crawford, 2016).

H.1 Disclosure of the evaluation process of the ADM system in operation

1. *What is made transparent?* The details of the evaluation process of the ADM system in practice. Additional information might be the frequency of evaluation and the consequences of unsatisfying results.
2. *Actor*: The operator of the system.
3. *Explanation/Justification*: The forums can judge whether and how the anticipated benefit of the system's usage is quantified. It can also judge the suitability of the evaluation process.
4. *Cost*: The costs for publication might be low if the process is well documented by the operator.

H.2 Disclosure of evaluation results of the ADM system in operation

1. *What is made transparent?* How much the whole process, based on the usage of the ADM system, is able to satisfy the overarching goals, based on the evaluation process.
2. *Actor*: The operator of the ADM system.
3. *Explanation/Justification*: The forums can check whether the system's usage achieves the overarching goal.
4. *Cost*: We estimate the costs to be low.

III Full access to the output of the ADM system in operation

1. *What is made inspectable?* The forum needs an access that allows assessment of the functionality of a system in its given application scenario. This access can, for example, be created by surveying evaluated persons if they had the chance to see their in- and output data, using informed test candidates to probe the process or providing actual results of the system in its application scenario.
2. *Actor*: The operator of the ADM system.
3. *Explanation/Justification*: 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. Therefore, this mechanism can provide the most reliable information about how the system actually behaves in operation. As requirements can change over time (see, e.g., concept drift; Webb et al., 2016), this mechanism can also be used to monitor the system on a regular basis.
4. *Cost*: Since this mechanism allows experimentation with the system in operation, the costs are high. On the one hand, the effort to protect the persons affected by the system must be considered. Their data must not 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 risks that the forum will gain insights into the system behavior that jeopardize the trade secret, or allow the system to be gamed.

Table 1 summarizes the different transparency and inspectability mechanisms.

4. Discussion

In this article, we have provided an overview of the most important mechanisms to make operators and developers of AI systems accountable in an asynchronous manner. Due to the modular character of these mechanisms and a discussion of their respective costs, the list of mechanisms allows an individual selection based on a legal cost–benefit analysis: By our definition, transparency is identified with the disclosure of information which already exists or can relatively easily be prepared. Thus, the cost to apply this kind of transparency mechanisms is generally rather small (see Table 1). However, there is not yet a central infrastructure or any other defined way for this kind of information to be published and made accessible. For some specific transparency mechanisms, there are suggestions on how to best communicate them, as for example, the datasheets by Gebru et al. (2021) with regard to disclosing information about the data or the "model cards" by Mitchel et al. (2019) for disclosing information about an ADM system. Also, performing the tasks to produce artifacts that can be disclosed, such as testing and auditing procedures, usually involve a great deal of effort. However, these costs are not considered here, since the focus is on the costs, including the consequences, of transparency and inspectability mechanisms per se.

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

Table 1. *Summary of all transparency and inspectability mechanisms and their estimated costs. The costs for 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 A.*

Characteristic	Transparency	Cost of disclosing information	Inspectability	Cost of granting access
A. Requirements engineering	1. Disclosure of the application scenarios	Low	—	—
	2. Disclosure of requirement 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 the operationalizations	Low	I. Full access to the data	Medium
C. Training data set construction	1. Disclosure of the labeling process	Low		
	2. Disclosure of why which variables were included in the training data set	Low		
	3. Disclosure of the preprocessing techniques	Low		
	4. Disclosure of the training data properties	Low		
D. Method selection	1. Disclosure of the method, its implementation, and the parameters setting 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 evaluation results of the ADM system in operation	Low		

Note. The costs for A.2. and A.3. might be higher, depending on the circumstances (see explanations in Section 3).

By our definition, inspectability mechanisms allow a forum to directly validate information about an ADM system. Regarding their costs, such mechanisms take more time and effort to implement (see Table 1). Additionally, they may require confidentiality toward the forum, as reverse engineering based on the actually used training data (I), as well as the unrestricted access to the output of the ADM system for certain data (II/III) is possible.

Next to these problems, further problems are discussed. Paul B. de Laat, for example, lists four major concerns: privacy challenges, making a system gameable, publishing trade secrets, and the limited use of transparency mechanisms (De Laat, 2018, p. 534). Thus, requiring these mechanisms to be implemented and made accessible bears hidden costs that are very difficult to estimate up front. This is why many providers and users of ADM systems try to avoid these mechanisms (Ananny and Crawford, 2018). To mitigate the risks research is geared toward new solutions, for example, by introducing methods that prevent some of the risks, for example, by disclosing only aggregated information (Ohm, 2009, p. 1751), or by letting qualified organizations preprocess the information that is made transparent as outlined by Frank Pasquale (2015, 142). Others state that some risks, like the possibility to game systems, are exaggerated (Cofone and Strandburg, 2018).

As only few mechanisms described in the article have established terms, a systematic literature review was not feasible. In order to provide an overview as complete as possible, we have based our research on a generic development process and took the terminology of other processes into account as well (as described at the end of Section 2). Still, there might some mechanisms or sophisticated methods of how to realize them be missing.

The discussion shows that the actual costs, risks, and benefits are tied to the individual application context of an ADM system; thus, the appropriate selection of transparency and inspectability mechanisms can only be conclusively evaluated for a given application.

Finally, it is important to note that according to Bovens, ADM systems can only be run accountably if all four of the following steps are accomplished:

1. The selection of appropriate transparency and inspectability mechanisms;
2. The identification of the forums for each step in the long chain of responsibilities. This decision needs to be made early in the process, as the information and accesses should be prepared depending on the forum addressed: Domain experts may be able to handle raw information or specific technologies, the general public might need additionally prepared information. Depending on the social process, governmental organizations, legal bodies, NGOs and the general society are forums that are likely to be part of most accountability processes in one way or the other;
3. The process in which the decisions of the actor(s) are questioned and judged;
4. A process in which there are possible consequences for the actor that are actually enforced.

In this article, we have provided a modular tool set for the very first step of this process. We hope that this will support the implementation of the full procedure in the near future.

Funding statement. The research was performed within the project GOAL "Governance of and by algorithms" (Funding code 01IS19020; https://goal-projekt.de/en/) which is funded by the German Federal Ministry of Education and Research and the project "Deciding about, by, and together with algorithmic decision making systems," funded by the Volkswagen Foundation.

Competing interest. The authors declare that they have no conflict of interest.

Author contribution. All authors significantly contributed to the work at hand. M.P.H. and T.D.K. were responsible for the first draft. K.Z. had a supervisory role.

Data availability statement. This work is not based on any data that could be made available.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Agichtein E, **Brill E and Dumais S** (2006) Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM, pp. 19–26.

Allhutter D, **Cech F**, **Fischer F**, **Grill G and Mager A** (2020) Algorithmic profiling of job seekers in Austria: How austerity politics are made effective. *Frontiers in Big Data 3*, 5.

Alvesalo-Kuusi A, **Malik HM**, **Viljanen M and Lepinkainen N** (2022) Dynamics of social harms in an algorithmic context. *International Journal for Crime, Justice and Social Democracy 11*(1), 182–195.

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

Ananny M and Crawford K (2018) Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society 20*(3), 973–989.

Azevedo AIRL and Santos MF (2008) *KDD, semma and CRISP-DM: A parallel overview. IADS-DM.*

Baer T (2019) *Understand, Manage, and Prevent Algorithmic Bias: A Guide for Business Users and Data Scientists*. Berkeley, CA: Apress.

Bovens M (2007) Analysing and assessing accountability: A conceptual framework 1. *European Law Journal 13*(4), 447–468.

Burnette J (2007) Operationalization. In *Encyclopedia of Social Psychology.* Thousand Oaks, CA: Sage, pp. 636–637

Chang JC, **Amershi S and Kamar E** (2017) Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. New York: ACM, pp. 2334–2346.

Cheng C-H, **Huang C-H**, **Ruess H and Yasuoka H** (2018) Towards dependability metrics for neural networks. In *2018 16th ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE).* Beijing: IEEE, pp. 1–4.

Citron DK and Pasquale F (2014) The scored society: Due process for automated predictions. *Washington Law Review 89*, 1.

Cofone IN and Strandburg KJ (2018) Strategic games and algorithmic secrecy. *McGill Law Journal 64*, 623.

Cortes C, **Jackel LD and Chiang W-P** (1995) Limits on learning machine accuracy imposed by data quality. *KDD 95*, 57–62.

Crawford K (2016) Can an algorithm be agonistic? Ten scenes from life in calculated publics. *Science, Technology, & Human Values 41*(1), 77–92.

De Laat PB (2018) Algorithmic decision-making based on machine learning from big data: Can transparency restore accountability? *Philosophy & Technology 31*(4), 525–541.

Diakopoulos N (2016) Accountability in algorithmic decision making. *Communications of the ACM 59*(2), 56–62.

Diakopoulos N (2019) 6. Algorithmic accountability reporting. In *Automating the News*. Cambridge, MA: Harvard University Press, pp. 204–239.

DIN (2020) *German Standardization Roadmap on Artificial Intelligence*. Berlin/Frankfurt: DIN/DKE.

Dubber MD, **Pasquale F and Das S** (2020) *The Oxford Handbook of Ethics of AI*. Oxford: Oxford Handbooks.

European Commission (2020) White paper on artificial intelligence. A European approach to excellence and trust. https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf (accessed 26 April 2020).

European Commission (2021) Proposal for a regulation of the European parliament and of the council laying down harmonized rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206 (accessed 21 April 2021).

Felzmann H, **Fosch-Villaronga E**, **Lutz C and and Tamò-Larrieux A** (2020) Towards transparency by design for artificial intelligence. *Science and Engineering Ethics 26*(6), 3333–3361.

Floridi L and Chiriatti M (2020) Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines 30*(4), 681–694.

Frénay B and Verleysen M (2013) Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems 25*(5), 845–869.

García S, **Ramírez-Gallego S**, **Luengo J**, **Benítez JM and Herrera F** (2016) Big data preprocessing: Methods and prospects. *Big Data Analytics 1*(1), 1–22.

Gebru T, **Morgenstern J**, **Vecchione B**, **Vaughan JW**, **Wallach H**, **Iii HD and Crawford K** (2021) Datasheets for datasets. *Communications of the ACM 64*(12), 86–92.

Glinz M (2011) A glossary of requirements engineering terminology. *Standard glossary of the certified professional for requirements engineering (CPRE). Studies and Exam, Version 1*, 56.

Haixiang G, **Yijing L**, **Shang J**, **Mingyun G**, **Yuanyue H and Bing G** (2017) Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications 73*, 220–239.

Hall MA (1999) *Correlation-Based Feature Selection for Machine Learning.* PhD Thesis, University of Waikato Hamilton, Hamilton, New Zealand.

Hauer MP, **Kevekordes J and Haeri MA** (2021) Legal perspective on possible fairness measures – A legal discussion using the example of hiring decisions. *Computer Law & Security Review 42*, 105583.

Heaton J (2016) An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016*. Norfolk, VA: IEEE, pp. 1–6.

Hickey RJ (1996) Noise modelling and evaluating learning from examples. *Artificial Intelligence 82*(1–2), 157–179.

Holl J, **Kernbeiß G and Wagner-Pinter M** (2018) *Das ams-arbeitsmarktchancen-modell*. Wien: Arbeitsmarktservice Österreich.

Holl J, **Kernbeiß G and Wagner-Pinter M** (2019) Personenbezogene wahrscheinlichkeitsaussagen ("algorithmen"). Technical Report, Synthesis Forschung Gesellschaft MbH.

Hoyle D (2009) *ISO 9000 Quality Systems Handbook: Using the Standards as a Framework for Business Improvement*. New York: Routledge.

Kemper J and Kolkman D (2019) Transparent to whom? No algorithmic accountability without a critical audience. *Information, Communication & Society 22*(14), 2081–2096.

Krafft TD, **Zweig KA and König PD** (2020) *How to regulate algorithmic decision-making: A framework of regulatory requirements for different applications. Regulation & Governance.*

Kroll JA, **Barocas S**, **Felten EW**, **Reidenberg JR**, **Robinson DG and Yu H** (2016) Accountable algorithms. *University of Pennsylvania Law Review 165*, 633.

Lepri B, **Oliver N**, **Letouzé E**, **Pentland A and Vinck P** (2018) Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology 31*(4), 611–627.

https://doi.org/10.1017/dap.2023.30 Published online by Cambridge University Press

Little RJ and Rubin DB (2019) *Statistical Analysis with Missing Data*, Vol. *793*. Hoboken, NJ: John Wiley & Sons.

Lopes AT, **De Aguiar E**, **De Souza AF and 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.

Luengo J, **Garca S and 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.

Michener G and Bersch K (2013) Identifying transparency. *Information Polity 18*(3), 233–242.

Mitchell M, **Wu S**, **Zaldivar A**, **Barnes P**, **Vasserman L**, **Hutchinson B**, **Spitzer E**, **Raji ID and Gebru T** (2019) Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. New York: ACM, pp. 220–229.

Naja I, **Markovic M**, **Edwards P and Cottrill C** (2021) A semantic framework to support ai system accountability and audit. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*. Berlin: Springer, pp. 160–176.

Neyland D (2016) Bearing account-able witness to the ethical algorithmic system. *Science, Technology, & Human Values 41*(1), 50–76.

Ohm P (2009) Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review 57*, 1701.

Panteras G, **Lu X**, **Croitoru A**, **Crooks A and Stefanidis A** (2016) Accuracy of user-contributed image tagging in flickr: A natural disaster case study. In *Proceedings of the 7th 2016 International Conference on Social Media & Society*. New York: ACM, pp. 1–6.

Pasquale F (2015) *The Black Box Society*. Cambridge, MA: Harvard University Press.

Raji ID, **Smart A**, **White RN**, **Mitchell M**, **Gebru T**, **Hutchinson B**, **Smith-Loud J**, **Theron D and Barnes P** (2020) Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. New York: ACM, pp. 33–44.

Ribeiro MT, **Singh S and Guestrin C** (2016) "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, pp. 1135–1144.

Ribeiro MT, **Singh S and Guestrin C** (2018) Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. *32*. Wahington, DC: AAAI Press.

Roh Y, **Heo G and Whang SE** (2019) A survey on data collection for machine learning: A big data-AI integration perspective. *IEEE Transactions on Knowledge and Data Engineering 33*, 1328–1347.

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence 1*(5), 206–215.

Russell JP (2010) *The Process Auditing Techniques Guide*. Washington, DC: Quality Press.

Saurwein F, **Just N and Latzer M** (2015) *Governance of algorithms: Options and limitations. info.*

Smith L (2017) *Unfairness by algorithm: Distilling the harms of automated decision-making. Future of Privacy Forum.*

Snow R, **O'connor B**, **Jurafsky D and Ng AY** (2008) Cheap and fast–but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, HI: ACL, pp. 254–263.

Song C, **Granqvist F and Talwar K** (2022) Flair: Federated learning annotated image repository. *Advances in Neural Information Processing Systems 35*, 37792–37805.

Taylor A, **Marcus M and Santorini B** (2003) The penn treebank: An overview. In *Treebanks*. Dordrecht: Springer, pp. 5–22.

Verma S and Rubin J (2018) Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (Fairware)*. Sweden: IEEE, pp. 1–7.

Von Ahn L, **Maurer B**, **McMillen C**, **Abraham D and Blum M** (2008) Recaptcha: Human-based character recognition via web security measures. *Science 321*(5895), 1465–1468.

Wagner B (2019) Liable, but not in control? Ensuring meaningful human agency in automated decision-making systems. *Policy & Internet 11*(1), 104–122.

Webb GI, **Hyde R**, **Cao H**, **Nguyen HL and Petitjean F** (2016) Characterizing concept drift. *Data Mining and Knowledge Discovery 30*(4), 964–994.

Westerlund M (2019) The emergence of deepfake technology: A review. *Technology Innovation Management Review 9*(11), 39–52.

Wieringa M (2020) What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. New York: ACM, pp. 1–18.

Zweig K (2019) *Ein Algorithmus hat kein Taktgefühl: Wo künstliche Intelligenz sich irrt, warum uns das betrifft und was wir dagegen tun können*. Munich: Heyne Verlag.

Zweig KA, **Krafft TD**, **Klingel A and Park E** (2021) *Sozioinformatik – Ein neuer Blick auf Informatik und Gesellschaft*. München: Carl Hanser Verlag.

Zweig KA, **Wenzelburger G and Krafft TD** (2018) On chances and risks of security related algorithmic decision making systems. *European Journal for Security Research 3*(2), 181–203.

Cite this article: Hauer MP, Krafft TD and Zweig K (2023). Overview of transparency and inspectability mechanisms to achieve accountability of artificial intelligence systems. *Data & Policy*, 5: e36. doi:10.1017/dap.2023.30