Algorithm Assurance: Auditing Applications of Artificial Intelligence

Alexander Boer, Léon de Beer, and Frank van Praat

1 Introduction

Algorithm assurance is a specific form of IT assurance that supports risk management and control on applications of risky algorithms in products and in organizations. These algorithms will often be characterized in organizations as applications of Artificial Intelligence (AI), as advanced analytics, or—simply—as predictive models. The aim of this chapter is to introduce the concept of algorithm assurance, to give some background on the relevance and importance of algorithm assurance, and to prepare the auditor for the basic skills needed to organize and execute an algorithm audit.

An algorithm is essentially a recipe to solve a specific class of problems using a finite sequence of well-defined instructions. Starting in an initial state with input data that characterizes the problem, execution of the algorithm proceeds through a finite number of successor states, terminating in a final state with output data that solves the problem.

The concept of an algorithm is an important vehicle for communication of scientific results between computer scientists, and mathematically proving desirable properties of algorithms is an important part of those scientific results. Those desirable properties may for instance be related to the worst-case running time of the algorithm, the characterization of the specific class of problems it solves, or the qualities of the solutions it comes up with. In practice, this allows programmers to apply routine algorithms without further research if they can ascertain that (1) the problems they want to solve belong to the class of problems that can be solved by the algorithm, and (2) the desirable properties of the algorithm match the task at hand.

A. Boer · L. de Beer · F. van Praat (✉)
KPMG, Amstelveen, The Netherlands
e-mail: Boer.Alexander@kpmg.nl; deBeer.Leon@kpmg.nl; vanPraat.Frank@kpmg.nl

E. Berghout et al. (eds.), *Advanced Digital Auditing*, Progress in IS,
https://doi.org/10.1007/978-3-031-11089-4_7

Algorithms are in common parlance specifically associated with the field of AI (see Sect. 3 in chapter "Introduction to Advanced Information Technology" of this book), because that field aims to build computer programs that can perform tasks that would otherwise have to be performed by a skilled human being. The field of Artificial Intelligence pushes the envelope, looking to expand the class of problems to which algorithms can be applied. Sometimes with spectacular results, but also with considerable risk. AI uses of computer programs often introduce considerable risk, and this risk can be attributed to risky applications of algorithms to real-world problems that can have a profound impact for those involved. Applications of algorithms are, in essence, always fundamentally questionable given the nature of the problems to be solved. If the application of the algorithm to the class of real-world problems is sufficiently well-understood and becomes routine, it stops being of interest to Artificial Intelligence. Or the media, for that matter.

An important tool in the toolbox of AI is the machine learning algorithm, which is capable of adapting to the problems it is exposed to by learning. An ML algorithm only has a capability to learn to a certain extent, and that extent is often not well-understood. This type of algorithm is trained by exposure to data reflecting the class of problems it is supposed to solve. In chapter "Introduction to Advanced Information Technology," Sect. 3.1 of this book a distinction was made between three different modes of learning: supervised learning, unsupervised learning, and reinforcement learning. This distinction is going to be important for understanding this chapter.

Algorithm assurance is not about the properties of the algorithm itself, but about its implementation in a computer program and about its application to real-world problems. The object of assurance is never the algorithm itself. It is a computer program, or component of a computer program, containing implementations of a risky algorithm or algorithms, to be reviewed in the context of a task in which it is applied or a prospective class of tasks in which it may be applied (in case of for instance admissibility in a market).[1]

In this chapter we will introduce the algorithm assurance engagement as a specific type of IT audit. After a general discussion of the background of algorithm assurance and the type of IT applications we are concerned with in this type of engagement, we will extensively discuss the scope of an algorithm assurance engagement, how to approach the risk assessment that should take place initially, how to set up and audit plan, and the audit techniques and tools that play a role in an audit plan. In Sect. 7 we discuss some examples of development skills that may be called on by the audit team during an engagement to help it judge risk and find problems. Throughout the chapter we use a running example—introduced in Sect. 3—and discuss the various sections in context of that running example throughout the chapter.

[1] Because the term algorithm in this context has become equated to implementations and applications of algorithms, we will indiscriminately use the term algorithm wherever we mean implementation or application of the algorithm.

2 Background

We are increasingly surrounded by, and dependent on, applications of AI technology. And its potential dangers are increasingly worrying us. Dystopian perspectives of the future in literature, film, and games demonstrate the potential ramifications of decision-making computers using data about us. Basically, these dystopian perspectives have been introduced since the idea of general purpose computers started gaining traction.

Over the last decade these worries have led to terms like AI, algorithm risk, and algorithmic bias entering common parlance in the context of *burning platform*[2] situations and in broad and general discussions about the risks and ethics of application of AI. These discussions have led to new legislation focusing on the uses of data and the uses of algorithms. For instance, the General Data Protection Regulation, which limits the uses to which data about people can be put in automated decision-making. Another example is the Artificial Intelligence Act, which addresses various forms of manipulation and harm caused by AI. The Digital Services Act and Digital Markets Act address unfair competitive advantages caused by data collection and manipulation through recommendation algorithms. These discussions have also brought the topic of accountability for harms caused by algorithms to the attention of organizations.

The implementation and application of algorithms has therefore also become a Governance, Risk, and Compliance topic. As a consequence, there is a growing call for algorithm assurance services. But not every algorithm—in the computer science sense—is an object of concern. Only algorithms that create unchecked risks, and only if their implementation, or application to a problem, may cause harm. In general, these criteria touch upon the characteristics of AI applications. For effective Governance, Risk, and Compliance over algorithms, these risky algorithms need to be identified and tracked first.

2.1 Common Risk Factors

The identification of key risks the algorithm poses to the company is a critical step in effective risk management. This step needs to be comprehensive. If a potential risk is not identified at this stage, it may be omitted from further analysis. This may result in material risks being given insufficient attention at a later stage. In algorithm assurance, material risks are often hard to pinpoint, as these often originate from the *blackboxness* or lack of transparency of the technology itself, but materialize as risks

[2] A risk management term referring to the explosion of the Piper Alpha oil platform in 1988, due to a small risk ignored by the entire industry sector. The burning platform situation creates a sense of serious urgency absent before.

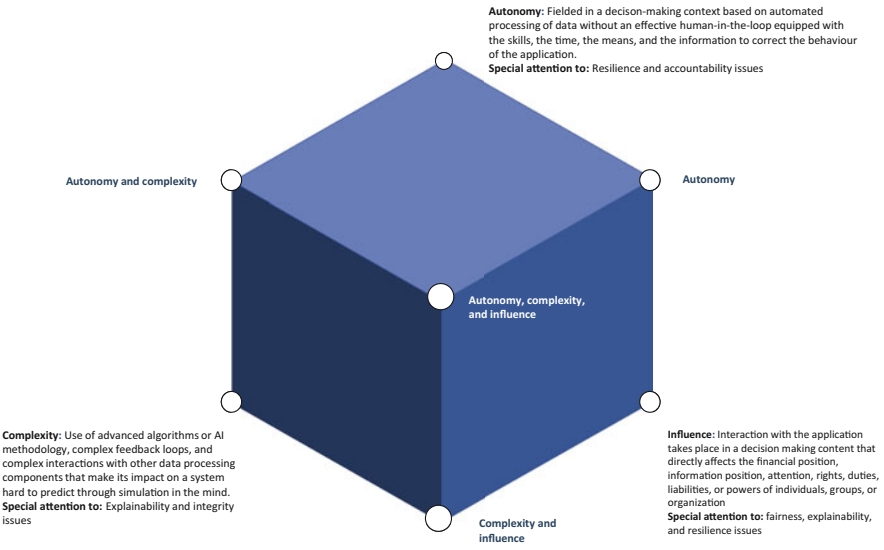

Fig. 1 Dimensions of risk and points of attention

in all kinds of other contexts. Common risk factors that relate to the deployment of algorithms may, roughly speaking, be grouped into three dimensions:

- Complexity
- Autonomy
- Impact

If the algorithm has a presence on all three dimensions, and on one of these dimensions can be considered high risk, it is likely to become a target for review or audit at some point for some reason. In Fig. 1, we show the three dimensions in the form of a cube. An easy way to convey risk profiles is scoring the application on each of the three dimensions and drawing a plane through the cube connecting the three selected points. At the axes we directly relate these risk dimensions to the five *control objectives* we use for our work: integrity, resilience, explainability, fairness, and accountability.

The first of these three dimensions is the *complexity* of the technology, of the task, and of the information ecosystem it operates in. In essence it relates to what is in the media often called *blackboxness*: the technology or information ecosystem is complex if it is hard to imagine simulating what it does in your mind and—importantly—if it is hard to recognize errors and hard to understand the cause of the errors it makes through simulation in the mind. Complexity can in this sense be seen as a dual of *explainability*, a concept that has been gaining in popularity in AI literature.

Complexity need not be directly related to the computational complexity class of the calculations made by the algorithm, or the complexity of the input data structure. These do definitely contribute to complexity: a deep learning-based algorithm will typically be considered more complex than a linear regression, and a linear

regression on many input parameters is more complex than a regression of a few parameters. But it is more often than not rather the complexity of the task to which they are set which is at issue. Facial recognition is for instance undoubtedly both computationally complex and based on a complex input data structure, but is often not seen as problematically complex. This is because the task—recognizing a face based on examples of that face—is not one we as humans usually consider complex. We appear to have an inborn talent for it, and we can often easily judge errors.

The algorithm may however still make errors that we would never make. Face recognition systems are for instance commonly fooled by holding a photo in front of your face, and that may be a fundamental flaw for the execution of the task to which they are set. For instance, if the face recognition unlocks a phone. The algorithm does what it was built to do: It recognizes the face. It is just not suitable for the complex task to which it was set. The task in this case turns out to be just a tiny bit more complex than the algorithm can reliably handle.

The second dimension is its *autonomy* in decision-making. The algorithm operates autonomously if it essentially functions without effective human oversight and its errors are likely to go undetected, unexplained, and unremedied. The face recognition phone lock scores high on these aspects of autonomy as well. Its user will be aware of false negative errors, when the phone does not unlock in the user's presence. The false positive error, unlocking without the user's presence, will go unnoticed. A last important aspect of autonomy is the algorithm's ability to autonomously adapt its behavior during its operational life by learning from its experiences without expert supervision. In general, this is a rare ability, but the face recognition phone lock has this ability as well. It learns to recognize its user without oversight by an expert, and without a formal validation process.

The third dimension is *impact*. Impact is determined by the characteristics of the task it performs. Impact is what is determined in an impact assessment, and is usually closely related to the motive for requesting an audit. It is material risk in the narrow sense: For instance, does the algorithm affect people's legal positions (it changes or establishes rights, duties, liabilities, etc.)? Does it handle money or valuable, private, or confidential information? Does it affect many people? Is it capable of abusing market power? The face recognition phone lock scores high on this dimension as well, because it may after all give access to all functions the user is authorized to access using that phone, including for instance banking and other functions based on authentication by phone.

Algorithm assurance differs from many other forms of assurance mainly on the impact dimension. A cybersecurity audit or an IT audit in the context of a financial statement audit is clearly scoped by a category of impacts on which the audit is focused. Algorithm assurance on the other hand focuses on the entity to be audited itself, and may cover a wide variety of impacts. Because algorithms may be set to any task, identifying its impacts requires some creativity from the auditor.

For governance functions scores on the three dimensions gain quick insight in the degree of attention an algorithm deserves, and what kind of risk mitigation needs extra attention. Complexity requires transparency and explainability, autonomy requires oversight, and impact requires explainability—because important decisions

must be justifiable—and impact-mitigating measures. As usual, everything starts and ends with the integrity of the implementation and application. If the algorithm doesn't effectively do what it is claimed to do, risk mitigation will not save us.

2.2 Algorithm Task Environments

Algorithms may be set to any task, and equally important, in any task environment. To get an overview of the field, we list some examples of categories of algorithms one may encounter in an algorithm audit.

A variety of algorithms are used for *financial prediction models*. These are commonly encountered in support of the financial statement audit, as they often have a direct effect on the financial statement. Technology used may vary from supervised machine learning to rule-based prediction models based on expert opinions, and hybrids of these. Typical issues are integrity and performance optimism, and less often gaming-the-system risks. The risk these algorithms pose mainly derives from complexity and impact on the financial statement. Compliance concerns relate to financial reporting regulations.

Supervised machine learning algorithms are typically used for *prognostic and diagnostic medical devices*. Applications range from prognosis of aggression by mental health patients based on non-invasive monitoring of vital signs to diagnosis of diseases of the retina using a high-quality camera. Typical issues are privacy and medical ethics concerns about data collection for training and testing the algorithm, equal performance on ethnic groups and genders, and presence of effective monitoring to check that actual use follows intended use. Compliance concerns relate to medical device regulation and regulation on medical ethics research involving human beings. Because decision-making is usually left to medical professionals, complexity of the algorithm is usually more of a concern than autonomy.

A variety of algorithms are used for *risk-based selection on applications* or claims to select suspicious applications for in-depth manual processing. Non-suspicious cases are then handled automatically. Technology used may vary from supervised machine learning, unsupervised machine learning (outlier detection or clustering when accurate training data for supervised learning is scarce), or rule-based prediction models based on expert opinions. Typical issues are differential treatment of groups based on static descriptors (profiling or discrimination), indirectly leaking sensitive data about individuals, and gaming-the-system risks because customers have reasons to game on ending up in the automatically processed or "happy" flow. Applications are for instance found in insurance, banking, policing, and taxation, and compliance concerns are often related to privacy and human rights. When operating on very large data streams, autonomy of the algorithm is a serious concern.

A variety of algorithms are used for *automated trading systems*, varying from basic robotic process automations for handling simple purchases or payments to high frequency, high volume flash trading of derivatives, to bidding agents for ad space. Technology used may vary from supervised machine learning to rule-based

prediction models based on expert opinions, and hybrids of these. Typical issues relate to intended use, oversight, and gaming-the-system risks. It is mainly the autonomy of the algorithm that is at stake. These systems may come into scope of the financial statement audit. More rarely compliance concerns related to for instance market manipulation (MIFID II) play an important role.

Unsupervised algorithms are often used for *clustering unstructured text into topics* to improve access to large corpuses of text. These texts are sometimes anonymized. A typical issue in this type of application is re-identification risk in anonymized corpuses based on the propensity of algorithms to cluster texts written by the same author together. Gaming-the-system issues may play a role as well. The leading compliance concern is generally privacy. The algorithms involved are usually just complex.

Recommendation algorithms for products, music, films, etc. usually involve a hybrid of reinforcement and unsupervised learning technology. Typical issues are differential treatment of groups based on static descriptors (profiling or discrimination) and gaming-the-system risks because suppliers of the products being recommended have reasons to game on ending up in recommendations. A less common compliance concern is self-preferencing by the organization running the algorithm if it acts as a supplier itself, which can be seen as an anti-competitive behavior by its business clients. Recommendation algorithms tend to be sensitive to *cold start* problems and *popularity bias.* Extra care needs to be taken when they are first deployed to mitigate these risks. These algorithms score high on autonomy.

A variety of algorithms are used for *profiling* and *ad targeting*. Hybrids of supervised, unsupervised, and reinforcement learning are used. Common issues in ad targeting is differential treatment of groups based on static descriptors (profiling) and indirectly leaking sensitive data about individuals. Compliance concerns are generally privacy and differential treatment of groups based on static descriptors (profiling or discrimination). Ad targeting business often also includes automated trading for advertising space.

The list of example task environments provides context to the rest of the chapter, but in the rest of the chapter we will limit ourselves to a single example task.

3 Running Example for This Chapter

As a detailed running example for this chapter to illustrate choices made in the audit, we introduce a public body that processes applications for child benefits. The public body does not have the manual processing capacity to investigate every application. Ninety-five percent of applications are processed automatically, following the claims made on the application form. In the vast majority of cases, this leads to an acceptance. In some cases, applications are directly rejected on formal grounds. Five percent are processed manually and claims are investigated in detail. Discretionary manual investigation can take anywhere from 5 min to many hours, often weeks in real time, leading to a final accept or reject decision. Manual investigation

can involve contacts with the applicant and third parties to collect additional information. If intentional noncompliance is suspected, the case may be handed over to a special investigation unit that will decide whether a report should be filed with the police.

The public body has a policy of picking applications for manual processing based in noncompliance risk. To help with this risk assessment it has introduced a supervised learning algorithm in the category of *risk-based selection on applications*, that selects risky applications based on historical information from applications manually processed in the past. The risky applications are automatically sidelined for manual processing. The algorithm will be retrained yearly, suing the new data generation by manual processing.

Processing takes place in the context of the GDPR. Based on specific administrative law about child benefits, the public body does however have special permission to process sensitive information about natural persons if this data is required for making decisions, and to collect additional information from third parties like banks, townships, or schools. The public body does however feel very vulnerable to scandals about unfair treatment based on sensitive attributes and has therefore decided to have the risk-based selection algorithm regularly audited so that it will be in control if a scandal would develop.

Because benefits will only be awarded if the parent takes care of children the majority of the time, child benefits usually go to the household where the mother is present (English, 2021). This leads to an increased likelihood that the historical data may be biased against single fathers and that this affects the algorithm. In addition, the rules about what is and what is not allowed have regularly changed over the last decade. Because it is clear that the historical data has been collected over a period in which the rules regularly changed, and presumably will keep changing, there is a risk that the algorithm is not as accurate and reliable as performance measures may suggest for the groups affected by the changes.

4 Scoping an Algorithm Assurance Engagement

In the previous section, we have introduced a model (see Fig. 1) with the three dimensions *complexity, autonomy, and impact* to determine if an algorithm is likely to become a target for review or audit. Especially if an algorithm is in its context perceived as impactful, the need to be assured of its reliability grows. In this section, we will discuss how to scope an algorithm assurance engagement by understanding the algorithm's context and the audit objectives, and how the context and audit objectives set the criteria that form the basis for the risk assessment.

4.1 The Importance of Understanding an Algorithm's Context

In any larger, more complex, social setting, algorithm assurance should not only focus on the (technical) properties of the algorithm itself, but also on its purpose as a problem solver in the real world. A standalone algorithm without task environment is not useful, but as soon as it is put into a complex task environment to perform highly impactful tasks, the things that can go wrong are countless. For the auditor, to comprehensively understand an algorithm in its context is crucial in order to start scoping an algorithm assurance engagement. The definition of an algorithm's success is in the end whether it is fit for purpose in the task environment in which it is embedded as a decision maker or decision support system. This purpose and the required skill level determine the technical requirements on the solution. In many cases, a traditional IT system will suffice, because most problems are relatively easy to solve. Only when the definition of success requires a more advanced type of solution due to the complexity of a real-world problem, the implementation of an AI algorithm should be considered. A computer program, or component of a computer program, that contains implementations of a risky algorithm or algorithms, is to be reviewed in the context of a task in which it is applied or a prospective class of tasks in which it may be applied. Figure 2 shows how traditional IT systems and advanced algorithms are often combined to work towards a single decision. In such situations, solely auditing an algorithm itself would make no sense.

Understanding the context of an algorithm requires an assessment and detailed understanding of a range of broader social and political facts about its stated definition of success. Typically, the context of an algorithm includes the process

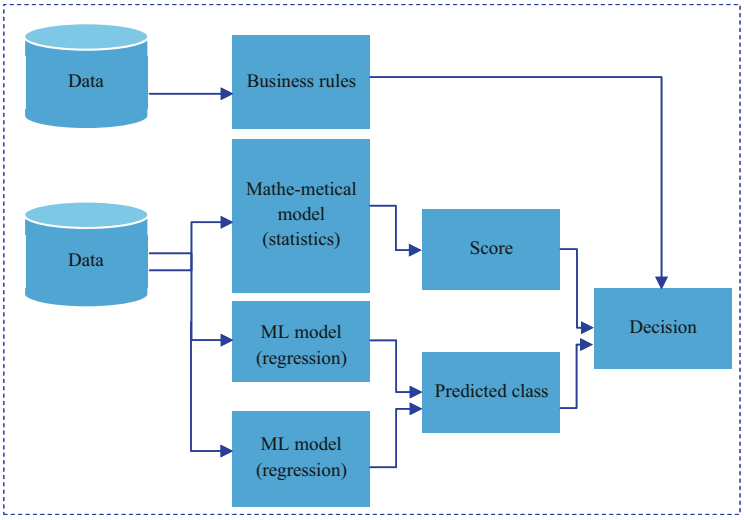

Fig. 2 Algorithm-based decision-making

of development of the algorithm, the process of preparing the data for training and testing the algorithm, the process of delivering an algorithm to its primary user, and often, most importantly, the setting within which it is used (Brown et al., 2021). To understand the algorithm's context and to take a first step in reviewing the algorithm itself, an important distinction needs to be made between a claimed skill and a claimed capability. Capability reflects the general problem-solving potential of the algorithm itself centered on accuracy and reliability claims, for a variety of tasks for which it could be fielded as a solution. Skill reflects the actual performance on a task in a specific task environment, including impact and autonomy aspects, and including risk-mitigating measures taken to control the task environment. An algorithm that works well in the Amsterdam office may not work in the Rotterdam office if the Rotterdam office lacks certain risk-mitigating mechanisms.

When we consider our running example again, the algorithms' definition of success is simple: detecting noncompliance. Incorrect applications are considered as a given, and the goal is to determine whether these applications are incorrect by accident or deliberate. The difference between accidentally or deliberately incorrect applications is of crucial importance in the context of this algorithm, because for mistakes made by accident the algorithm has no reason to create a signal. As a system for—essentially—fraud detection, compliance criteria and fairness criteria as typical issues for this type of fraud detection algorithm are differential treatment of groups based on static descriptors (profiling or discrimination). Consider how different it would be when a same type of algorithm is used with the purpose to identify incorrect applications to help citizens to better apply for subsidies? In that case, the definition of success would be entirely different and so are the relevant criteria to review.

4.2 Assurance Criteria

Over the past few years, many non-commercial and commercial organizations have issued principles for trustworthy AI. The EU High Level expert group for example, put forward a set of seven key principles that AI systems should follow in order to be deemed trustworthy (European Commission, 2019). Google as well introduced seven principles, and a complete audit framework for algorithms (Raji et al., 2020). Although these principles are to a certain extent similar, there are some notable differences. The EU stresses the importance of privacy and human oversight, while Google also finds it important to use AI only in alignment with scientific evidence.

If we consider how assurance engagements on other types of IT systems are currently carried out, the concept of overarching principles applies as well. The so-called *trust services criteria* (Ewals et al., 2019) are used as means to assess the extent to which an organization has controls in place to let IT systems operate in correspondence with the criteria.

Table 1 Overview of SOC2 trust principles, EU working groups, and coherent audit research questions

SOC2 trust principle	EU working group	Audit research question
Security	– Technical robustness and safety	– Can the data used by the algorithm be accessed by unauthorized individuals?
		– Are there risks of gaming the algorithm?
Availability	– Technical robustness and safety	– If the algorithm is business critical: how is its availability and contingency managed?
Confidentiality	– Privacy and data governance	– May the output of the algorithm lead to the identification of (protected) subgroups?
	Transparency	
Processing integrity	– Human agency and oversight	– Does the algorithm perform in line with its definition of success?
	– Accountability	– Is the algorithm fair and unbiased in its specific context?
	– Diversity, non-discrimination, and fairness	
	– Societal and environmental well-being	
Privacy	– Privacy and data governance	– Are there sufficient legal grounds to use the algorithm?
	– Diversity, non-discrimination, and fairness	

From an algorithm audit perspective, there are reasons to argue that such trustworthy AI principles are a good basis to scope an algorithm audit. This is because these principles provide a specific perspective, a set of control objectives appropriate for AI assurance, for an auditor to focus on. There is also reason to argue that the already existing trust services criteria are insufficient, because algorithm assurance should not only focus on the algorithm itself but also on the context in which it is being used. If you try to map the SOC2 trust services criteria to the AI principles of the EU working group, no exceptional creativity is required to successfully make it fit.

In an algorithm assurance engagement, the auditor should combine the auditees requirements with the context of the algorithm to select the appropriate criteria. We also provide some example audit questions that should be answered satisfactorily depending on the selected criteria (Table 1).

The auditee, or the client authorized to request the audit, may have its set of control objectives to be audited. The audit report should be relevant to its audience, after all. Business sectors moreover usually operate within a framework furnishing relevant assurance criteria as well. Various high-risk sectors, ranging from the financial, automotive, and health sector to the trade in children's toys, have, or will develop, guidance for using AI for high-risk functionality. If you are auditing a

medical diagnostic or prognostic application, for instance, there will be guidance that can be followed interpreting Medical Device Regulation regulations (e.g., there is a guidance for medical diagnosis in the Netherlands (Van Smeden et al., 2021)). Besides that, there will usually be a number of ISO/IEC standards to take into account. Sector-specific jargon and perspectives cannot be avoided, and over time algorithm assurance will require the development of a certain amount of sector specialization guided by scientific contributions (e.g., Wirtz et al., 2022).

Coming back to our running example of our algorithm to select applications for child benefits for manual processing, we argue that *diversity, non-discrimination, and fairness* would be the most relevant audit criteria. In this case, it would mean that the audit team will for example need to determine that the algorithm is unbiased against all protected groups. In addition, fairness is also about weighing the legitimacy of the task the system executes, how well it does at performing that task, its use of personal and sensitive data, and the quality and representativeness of that data for the task it performs. Assurance on diversity, non-discrimination, and fairness is therefore based on presumptions about technical robustness and safety and accountability. These should also be part of the audit team's investigations. Moreover, the targeted readers of the audit report are clearly citizens, politicians, journalists, and potentially a court of law. Having a good explanation of what the algorithm does is essential to risk mitigation. Investigating transparency is therefore unavoidable as well, even if the reported findings are about diversity, non-discrimination, and fairness.

There are two key differences between SOC2 assurance and algorithm assurance. Firstly, SOC2 criteria are formulated in a very generic manner, while in algorithm audits specific controls aligned with the algorithm's context and associated risks are crucial. Secondly, SOC2 follows the COSO-framework, which is extensive but in practical terms leads to audits that are fully focused on control testing only. In an algorithm assurance engagement, we argue that control testing only would fall short to be able to provide enough comfort about the algorithm working in alignment with the selected criteria. A typical audit approach for control testing is required to be augmented with other types of audit approaches such as testing the model itself or a form of substantive procedures. In the last section of this chapter, we will propose four of such approaches.

4.3 What Do the Trust Services Criteria Apply to?

In regular IT audits, one or a combination of the following components are assessed against the Trust Service Criteria during a SOC2 examination: Infrastructure, Software, People, Procedures, Data. In algorithm assurance, we argue that the scoping exercise in terms of (technical) components is subordinate to the importance of how an algorithm has been implemented in its context. Typically, we believe that the audit or review of an algorithm would focus for a large part on the steps that were carried out by the team that builds the model, instead of all the individual

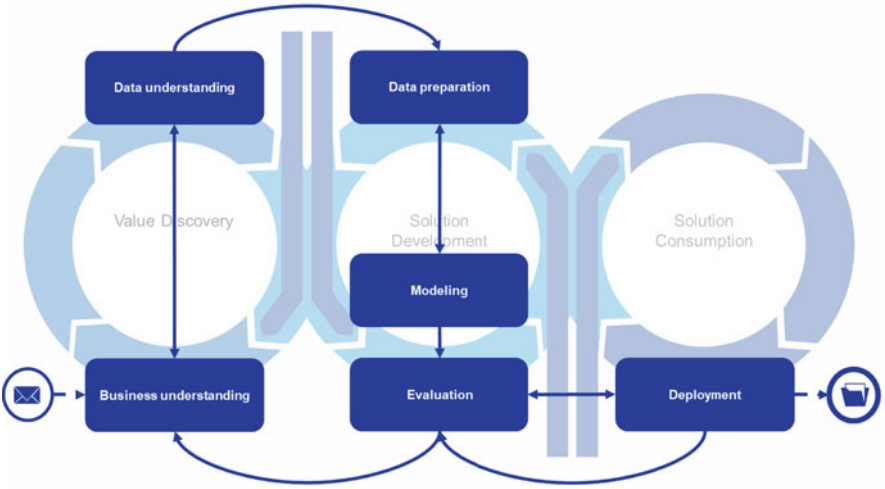

Fig. 3 Spheres of activity where risk and control play different roles

components of an algorithm and how they exactly operate. As described in Sect. 4.1, next to the setting in which an algorithm is used, it would also include the process of development of the algorithm, the process of preparing the data for training and for the process evaluating the algorithm, and the process of deploying an algorithm in its task environment. And finally, the central issue of developing a good problem conceptualization, which should be based on a realistic data understanding and business understanding. Generally speaking, we distinguish three different spheres of activity in the life of an algorithm (see Fig. 3). Each phase requires a different perspective on dealing with risk and control.

To further illustrate how the process of developing an AI algorithm is important, we return to our running example. When building a supervised learning algorithm that is aimed at identifying noncompliance, a common issue is the number of false negatives. As many noncompliant transactions will go unnoticed, the labeled data that is required to build a supervised learning algorithm is going to be extremely biased towards learning about true and false positives. It doesn't come as a surprise that in banks for example, unsupervised learning systems are favored for fraud detection over supervised learning algorithms to tackle this problem. Assuming that the developer in our example is aware of this general issue with fraud detection algorithms, there must be reasons why supervised learning was still preferred over other type of models. The relevant question to ask as an auditor is: How did the developer come to this decision, and what steps were taken in order to discover the false negatives for which no outcome of manual processing is available. How the developer has coded its model and what frameworks were used is considerably less important.

4.3.1 An AI Model's Technical Architecture

AI algorithms are often hidden behind user interfaces, web services or in software components. There is no one typical AI architecture that is common across all AI capabilities. If we browse online through the setups that are disclosed by companies or third-party vendors, we mostly come across an overview of relevant platforms, frameworks, and supporting tools during the development and deployment cycles of algorithms only. Each year Firstmark[3] publishes an overview of all relevant vendors in the ML and AI business in the so-called Machine Learning, Artificial Intelligence, and Data (MAD) Landscape. The overview distinguishes high-level categories to show what is available in the marketplace. The MAD Landscape shows a myriad of vendors arranged by type of services, ranging from infrastructure and data (re)-sources, to analytics and machine learning/AI platforms. For an auditor, it would never be possible to build the required expertise to appropriately assess all the hundreds of different products available on the marketplace.

The audit team should limit itself to the development process instead of the specific platforms, frameworks, and tools to perform AI and Machine Learning tasks. Uber, the taxi and food delivery company that is well-known for its advanced AI deployments, provides some guidance in this regard. The description of Michelangelo, their Machine Learning platform, is based on the steps taken in the machine learning lifecycle[4] instead of the technical architecture: manage data, train models, evaluate models, deploy models, make predictions, monitor predictions. Another common model that is used to lay out the AI development lifecycle is the CRoss Industry Standard Process for Data Mining (CRISP-DM),[5] which also forms the basis for our previously presented Fig. 3 on spheres of activity where risk and control plays different roles.

4.4 Stakeholders in the Audit and Accountability

As part of the criteria, we identified accountability as one of the key aspects to look into. The assurance engagement should be *scoped* towards the risks that matter to the client, depending on the purpose of the engagement and the algorithm's context.

An algorithm assurance engagement may be motivated by internal risk management needs of the engagement client, reporting obligations to supervisory authorities, the risk management needs of one or more third-party stakeholders in the decisions the algorithm takes or supports, or a general need for transparency towards society. The risks that need to be focused on are determined by the motive for the engagement.

[3] https://mattturck.com/data2021/

[4] https://eng.uber.com/michelangelo-machine-learning-platform/

[5] https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

An important aspect to scoping the problem is whether the assurance client is a provider of the algorithm, a user of the algorithm, or both—in case an algorithm developed in-house is used. This is an important question from an accountability point of view since the provider and user have different responsibilities. The provider needs to provide something that will work well *if* the manual is followed. Assurance is in this case mainly about consistency between claims about the algorithm and their substantiation by the algorithm *if* it is used correctly. The user needs to follow the manual: any deviation from intended usage is a relevant finding, and potentially a source of additional risk.

4.4.1 Accountability of Cloud Providers

Most companies these days use some sort of cloud computing to reap the benefits of AI. For many companies Uber's approach to set up an end-to-end platform from scratch is unrealistic, because of the required investments and the scarce knowledge that is required to set up such a platform. Therefore, most companies turn to the larger cloud vendors such as Microsoft's Azure, Amazon's Web Services, and Google's Google Cloud to work with off-the-shelf learning algorithms. For the auditor these larger vendors remain an almost insurmountable obstacle, as they typically try to avoid to contractually agree on a right to audits. In these situations, the process approach helps to limit the reliance on the work done by the cloud providers. It is increasingly common to depend on ISO/IEC 27001 and 27018 certifications from cloud service providers.

5 Risk Assessment

In Sect. 2.1 of this chapter, we introduced a simple three-dimensional risk model and classification method for determining whether an algorithm is a suitable candidate for algorithm assurance. In practice, the algorithm rarely scores as high risk on all three dimensions of the risk model, because the presence of clear risks on two of these dimensions typically leads to lower risk choices on the third dimension. The risk classification method does not replace a true risk assessment. It selects candidates for a risk assessment. In this section we introduce a risk assessment method based on identifying risk likelihood drivers and impact drivers in the task environment. We also discuss the need for a diverse audit team composition.

5.1 Drivers for Likelihood and Impact

Identifying the key risks an algorithm poses to the company is a critical step in effective risk management. This step needs to be comprehensive. If a potential risk is

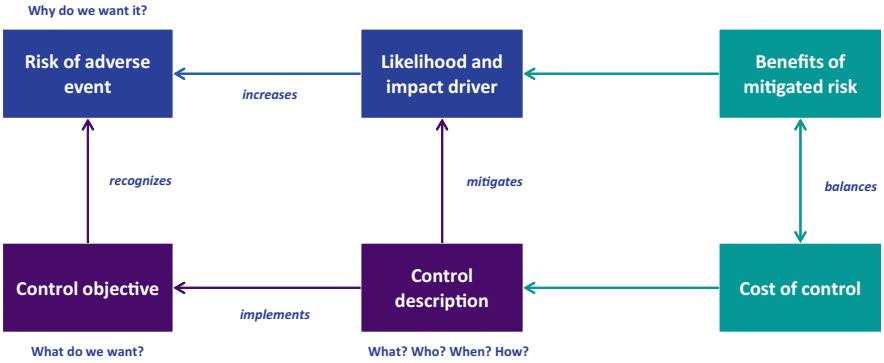

Fig. 4 How control objectives, risks, and likelihood and impact drivers relate to each other

not identified at this stage, it may be overlooked during further analysis. This may result in material risks being given insufficient attention at a later stage. In algorithm assurance, material risks are often hard to pinpoint, as these often originate from the *blackboxness* or lack of transparency of the technology itself, but materialize as risks in other places.

In Fig. 4, we relate the ingredients of our approach to AI Assurance to each other. The *risk* you take with an algorithm is your exposure to loss or damage caused by *adverse events* involving the algorithm. Which events you consider adverse events is determined by your *control objectives* (like the aforementioned seven AI Ethics principles). A *likelihood driver* is a circumstance (in the task environment, or during the conceptualization of development phases in Fig. 3) that increases the probability of adverse events happening to the algorithm. An *impact driver* is a circumstance that increases the impact of adverse events, usually by enabling additional adverse events to happen to people, processes, data, etc. *Controls* mitigate for the circumstance that increases the probability or impact of the adverse event happening to the algorithm. Generally, the point of risk mitigation processes is:

- To create awareness of likelihood and impact drivers present in the environment of the algorithm
- To select and implement controls that reduce the total amount of risk to an acceptable proportion
- To periodically check the continued presence and operation of the controls

For most auditors, likelihood drivers and impact drivers will sound new. Typically, a risk assessment is carried out in terms of likelihood and impact only. In algorithm auditing specifically, likelihood is often replaced by complexity, suggesting that if a model is more complex automatically its risk profile rises. We argue that this equivocation is far too broad and simple. An algorithm's context is much more decisive for its risk profile than its complexity, and combinations of factors constitute risk. A three- or five-point scale from low to high is used to build a risk profile. We believe a solid risk assessment should take it a step deeper

considering factors contributing to the likelihood or impact of adverse events. Since risks factors involving algorithmic bias often form mechanisms that can be expressed in the form of causal loops, we recommend to, where appropriate, assess the drivers in the form of causal loop diagrams or a similar diagramming technique.

The context of the algorithm, in combination with the control objectives you committed to, determines what the relevant adverse events are. When doing the risk assessment, the auditor should hypothesize what outcomes are to be considered as irregular in relation to the algorithm's normal performance and behavior. In general audit terms, these adverse events are often referred to as what-could-go-wrongs. These must be reduced to acceptable proportions using controls. Acceptable risk relates to the cost of control: Controls usually have a cost, and that cost has to be balanced in practice against the risk mitigation benefits of the control mechanism.

In our running example of the public service organization selecting applications for manual processing, we can also make a distinction between likelihood and impact drivers. As mentioned earlier in this chapter, supervised learning algorithms used in fraud detection are typically known to be very susceptible for their lack of ground truth. Because typically only the fraud that meets human expectations is discovered, other types of fraud are not identified and therefore the data only shows parts of the truth. This ground truth issue clearly classifies as a driver on likelihood: lack of representativeness of the available training and testing data for the data that the algorithm receives as input (including all the false negatives) directly contributes to the risk that the algorithm, and its evaluation, will be inaccurate. The organization *could* control for that risk through random sampling for manual processing, but searching manually for the false negatives is going to be very costly in man hours and this cost of control may be at odds with the business case for the algorithm.

The purpose of the algorithm is an impact driver: because the outcome of the process directly affects the legal and financial position of citizens, and citizens do not usually participate in that process for fun only. Even the delay caused by selection for manual processing may be considered unfair.

The possibility of bias against single father household applicants is a typical adverse outcome in the fairness category. Because benefits will only be awarded if the parent takes care of children the majority of the time, child benefits usually go to the household where the mother is present (English, 2021). There is a clear likelihood factor present: likelihood that the historical data may be biased against single fathers. This may affect the algorithm. From a risk assessment perspective, the auditor (and public body) should take into account that the impact of an accusation of algorithmic unfairness may be considerable. Single fathers may generate a lot of attention and sympathy in the media, and differential treatment without a good justification may be considered a human rights violation in court. Impact may be considerably reduced by having a good explanation ready at hand for the media for any apparent differential treatment.

5.2 A Standard Set of Likelihood and Impact Drivers

A comprehensive risk assessment of an algorithm highly depends on the context and the real-world problem. AI algorithms are associated with risks that capture the public imagination, and stir the interests of regulators: deanonymization, profiling, unfairness to protected groups (discrimination), surveillance, restriction of freedom of speech, gaming the system, hampering competition, disturbing public order, abuse of markets, and abuse of information position. Financial risks often relate to the costs of reparations: manually re-doing processed cases, litigation costs, fines, damage, loss of reputation.

In the overview below, we present some examples of likelihood and impact drivers including a short description from our own risk identification inventory. By no means this should be perceived as an extensive list of algorithm risks, but it helps the auditor in the line of thinking to objectify the likelihood and impact of algorithms not operating in line with their definition of success (Tables 2 and 3).

5.3 Who to Involve in the Risk Assessment?

There is increasing consensus (Shen et al., 2021) on the relevance of involving a heterogeneous group of people in terms of cultural background, technical expertise, and domain expertise in the development teams of AI algorithms. By making people with a pluriform background part of a development team, the integrated team will be better at conceptualizing a real-world problem from different perspectives. Consequently, pluriform teams develop better AI algorithms with and diminish the likelihood of undetected risks. In the same vein, we argue that this line of reasoning also holds for algorithm auditing, and carrying out the risk assessment as part of it (Shen et al., 2021). Making sure that an audit team that performs a risk assessment represents the cultural and gender demographics of the stakeholders in the algorithms that they are auditing, major blind spots on stakeholder impact with potentially critical issues surfacing only post-deployment can be already identified during a risk assessment. Composing a heterogenous team is not always achievable, but making sure the audit team has a certain level of heterogeneity will actually help to assess an algorithm in its broad context.

6 The Audit Plan

In this section, we will discuss how to formulate an audit plan, how traditional tools and techniques from the auditor can be leveraged during an algorithm audit, and how AI-related skills play a crucial role to perform successful algorithm audits.

Table 2 Overview of impact drivers and rationale thereof

Likelihood drivers	Explanation
The predictions of the AI application cannot be adequately or timely verified by observation to measure performance.	For an AI application, you would like to know whether your prediction also came true. In some cases, this is not possible. For example, when the AI application predicted when something would break, but it is repaired before that specific date. Or whether a mortgage loan will be paid off, which is known only after 30 years.
All training and evaluation data originates from one specific task environment.	In case an AI application is designed in a specific environment, but is executed in a different environment, the outcomes might not be correct. For example, predicting what EU citizens would like to pay for a hotel based on Europe, but erroneously assuming this model will predict correctly for South America. Since the EU cannot be compared to South America, the model will likely not be generalizable.
Experts making the same decision with the same information report complex and diverse reasoning patterns for different cases that are hard to capture by the machine learning technology applied from a learning capacity perspective.	The complexity of the task environment is beyond the learning capacity of the algorithm employed. For instance, if you train an application to predict whether someone is ill, completely ignoring the fact that doctors distinguish a lot of different diseases with different underlying mechanisms. Better to train an algorithm per disease category, and combine these in a hybrid system. This type of application will moreover create huge explainability problems.
The risks involved in wrong predictions made by the AI application for downstream tasks are not adequately distinguished from the accuracy of predictions in performance measurement, leading to a conflation of accuracy and utility of the AI application.	Any abductive argument is uncertain, in the sense that you jump to a conclusion knowing you may be wrong. How tolerant you are of making mistakes depends on the value of the conclusion in tasks that functionally depend on it. This risk tolerance needs to play a role in the measurement of performance, but should not be implicitly mixed in with accuracy. Conflation means treating two distinct concepts—in this case accuracy and utility—as if they were one, which produces errors or misunderstandings as a fusion of distinct subjects tends to obscure analysis of relationships which are emphasized by contrasts. Very common mistake, for instance if the *F*-value statistic is used for performance measurement without consideration of risk appetite for false positives and false negatives, which is important for determining the utility of an algorithm.
The AI application operates in a task environment that requires complex interactions with	Certain failure modes may be easy to prevent for an individual agent, but may arise for a

(continued)

Table 2 (continued)

Likelihood drivers	Explanation
other software agents, consists of a complex combination of AI techniques or models, or is input to, or dependent on the output of, other AI applications.	combination of agents. Typical examples are market abuse (MIFID II rules) or algorithmic price cartels. Even though each individual trading agent keeps to MIFID II rules, all agents in the organization taken together may violate them. Similarly, one agent may simply be following market prices, a cluster of agents may form a cartel setting prices.

6.1 Audit Approaches

The aim of the audit plan is to formulate the required steps to perform the audit based on the approach that is the most feasible. We present four high-level approaches an AI auditor could follow to structure the audit plan. These approaches have a different area of focus and in practice will often be combined into an audit plan tailored to the case at hand (Table 4).

6.1.1 Approach 1: Evaluation of Algorithm Entity Level Controls

As part of this approach, the auditor shall evaluate at enterprise level whether sufficient entity level controls are in place to ensure algorithms are built and managed in a controlled environment. Controls in the area of AI strategy and policies, data governance, technology and platforms, skills and awareness, and development methodology should be part of the review. When only assessing a company's entity level controls, no direct assurance regarding the outcomes of an individual algorithm would be possible, but in general it may help to identify and assess overarching risks.

Algorithm entity level controls generally reduce the *risk of failure* for the algorithm and its outcomes, allowing for reduction of depth of testing (model test) or sample size (substantive procedures). An advantage to this approach is its feasibility. Testing entity level controls would only require traditional control evaluation procedures such as inquiry, inspection, and reperformance.

6.1.2 Approach 2: Testing the Model

As part of this approach, the auditor shall perform an in-depth assessment to determine if the algorithm performs in line with relevant audit criteria and whether the identified risks are properly mitigated. The approach to test an algorithm itself is generally speaking not too different from testing an automated control, because the initial focus would also lie on design and implementation. Still, for machine learning and AI algorithms (i.e., not rule-based models), the test of design is fundamentally

Table 3 Overview of impact drivers and rationale thereof

Impact drivers	Explanation
The decision made by the AI application significantly or irreversibly affects the interests or legal position of people.	The decision the system takes can affect legal position, financial position, or emotional interests. For example, rejecting to pay out a claim or to give the person a mortgage, award custody over children, infringe on people's privacy, stigmatize them, etc. Basically, anything that may drive people to court, causing damage to the organization.
The AI application takes decisions fully autonomously, without or only with pro forma supervision by people.	The decision made by the AI application is final and in practice not reviewed by a human. Adverse events may go unnoticed for some time, causing damage. This is most common for system that takes decision with a high frequency, like trading and recommendation systems.
Unfairness extends specifically to a subpopulation defined by a legally protected attribute (like ethnicity, gender, religion, etc.) that is required to be protected in that task environment.	AI application outcomes could be unfair to a subpopulation defined by a legally protected attribute. For example, the outcome could be unfair to women, giving them a lower chance of getting invited for a job interview. Presence of this driver increases the chances of other damage, and the organization may violate its own ethical principles.
The adverse outcome causes significant reputation damage.	The use of the AI application can cause significant reputation damage when certain adverse events happen. This depends on the presence of other impact factors, but also significantly on how visible the functioning of the system is to the outside world. A system that is open to outsiders for probing may for instance easily be tested for manipulations, or unfairness, and this increases the chance of reputation damage. We recommend carefully checking each adverse outcome individually!
The AI application handles or informs decisions about large amounts of money, or involves significant financial exposure.	The algorithm handles for the company a significant amount of money, for example a pricing algorithm for a significant account or revenue stream, for an online web shop or trading algorithm. Failure of the algorithm may lead to losses for the organization or other stakeholders. Note that this circumstance is relevant for financial assurance.

different from testing regular IT functionality. The key difference is that the logic captured in the algorithm is not specified up-front but is discovered from the training data during model training. Furthermore, the logic may evolve through time as a result of offline or online retraining and automated feedback loops. The assessment should therefore focus on the assumptions and design decisions that were made by

Table 4 A matrix of audit approaches with coherent focus area, the difficulty and feasibility of the audit

Audit approach	Focus area	Level of comfort	Feasibility
Evaluation of algorithm entity level controls	Overall algorithm control environment	Low	High
Testing the model	Algorithm design and maintenance	Medium to high	Medium to low
Testing monitoring controls	Algorithm output	High	Low
Substantive testing	Algorithm output	High	Low

the algorithm developers in conceptualizing the initial business problem into a formalized AI problem. Of course, the quality of the data and data preparation activities should also be in scope of these audit procedures. To test an algorithm's implementation the same types of test procedures as in regular IT audits can be used as a starting point, although some types of procedures may be less applicable or feasible, depending on the characteristics of the algorithm. In the subsection on tools and techniques, we will go in more detail.

Testing the model can provide a high level of comfort, depending on the detail of testing. If for example advanced techniques such as algorithm replication are used, the level of assurance on the quality of the algorithm will increase, because it requires the auditor to independently reperform (part of) the algorithm's development process.

The feasibility of this approach depends heavily on the complexity of the algorithm and availability of data sets. For rule-based algorithms, feasibility is much higher as explicit business rules provide clear criteria to test.

6.1.3 Approach 3: Testing Monitoring Controls

As part of this approach, the auditor should test if the enterprise put internal controls in place to monitor the transactions performed by the algorithm and mitigate the risks of algorithm failure. Essentially, this is a sort of black box approach focusing on the output of the model instead of its inner workings. Testing monitoring controls might be a preferred approach as it circumvents the complexity of testing the algorithm itself. However, this approach also has some drawbacks. Firstly, the implementation of algorithms may render traditional monitoring controls obsolete (e.g., controls involving comparison of employee performance are not possible if all employees are replaced by a single algorithm). The auditor should carefully assess if the monitoring controls are sufficient to mitigate the relevant algorithm risks. Secondly, monitoring if individual algorithm outcomes are correct is often not possible or feasible (unless for some rule-based applications or very trivial classification tasks like image recognition). We notice that controls aimed at directly assessing the quality of algorithm output are still rare today. Controls are more likely to monitor if data

distributions in transactions stay between predefined boundaries and identify outliers for manual follow-up.

The level of comfort provided by this approach depends on the type of controls and their goal. In case monitoring controls directly assess the quality of the individual algorithm transactions, high levels of comfort can be achieved. In all other cases, for example when monitoring is only done on aggregated figures, the level of comfort is much lower.

6.1.4 Approach 4: Substantive Testing

As part of this approach, the auditor should test if (a sample of) transactions were processed by the algorithm in line with relevant criteria. Similar to testing monitoring controls, substantive testing should be considered as a black box approach potentially leading to high levels of comfort. But potential issues are also to be considered. Firstly, it cannot easily be determined if algorithm output was correct or incorrect (or such information may only become available with a significant time lag). If such information was readily available, the algorithm would not be required in the first place. This severely limits the applicability of testing the reliability of algorithms through transaction analysis (in fact a form of black box testing). For example, for mortgage loans it takes 30 years before the predicted probability of default can be validated. Or for recruitment algorithms, the actual job performance of rejected candidates will never be known (setting aside practical problems related to object job performance evaluation). Secondly, depending on transaction volume a key issue with substantive procedures is that testing a significant number of transactions may be very time consuming. After all, algorithms are used to automate complex decisions not easily captured in simple business rules. And thirdly, due to opaqueness of the input-output relationships it is hard to determine if a sample of transactions provides sufficient evidence for the entire population (representativeness issue).

This approach provides a high level of comfort, as long as the sample that is tested is sufficiently large to properly represent the algorithm's performance. In that case, substantive testing gives high levels of comfort as the outcomes are directly tested per transaction.

6.2 Tools and Techniques

When the auditor has selected the most feasible approach, or a combination of them, there are multiple tools and techniques in the standard auditor's toolbox that can be used to perform the algorithm audit. In principle, the same types of test procedures can be used as in regular IT audits. Some types of procedures may be less applicable or feasible, depending on the characteristics of the algorithm. We discuss five types

of test procedures, which can be used in combination, to test the design and implementation of an algorithm.

Inspection Similar to regular IT audits, all the relevant documentation as output of the steps followed during development is reviewed. In case of an algorithm audit, the documentation should at least provide detailed information about the algorithms' definition of success and how it aligns with the problem conceptualization, the ways data exploration was done, how feature engineering was performed and how feature importance was measured, the configuration of hyperparameters, how overall testing and validation has been done, etc. Of course, this type of test procedure can only be used if the algorithm development and maintenance processes of the organization are sufficiently mature.

Reperformance On top of inspection, the auditor can also choose to reperform certain activities executed by the development team. For example, in case of supervised learning, the training phase can be reperformed using the same training/test dataset and the same parameters as the algorithm's developers to establish if this results in the same algorithm with the same performance (small differences may occur due to different random seeds). This type of test procedure requires specific expertise on part of the auditor and the auditee must be willing to provide the auditor access to the original data and an environment to train the algorithm.

Code review A code review on itself would never be sufficient to get the required comfort for algorithm assurance. Code reviews should therefore always to be used in combination with other testing procedures. The added value of code reviews is sometimes a topic of discussion, as in most algorithmic solutions the machine learning algorithm itself is not really implemented in readable code itself, but rather an off-the-shelf asset. Code reviews are especially relevant for custom code or scripts or if uncommon libraries are used.

Independent testing This type of procedure involves testing the algorithm using an independent dataset developed by the auditor. Independently testing an algorithm would require deep expertise about the specific technological details of the algorithm under review. The data set should be representative for the dataset that was used to build the algorithm, which can be a great challenge. But in scenarios where the impact of the algorithm is great, and the auditee demands a great amount of comfort, there just might be sufficient justification to use this type of approach.

Replicating functionality Just like for independent testing, replicating an existing algorithm's functionalities also requires deep expertise of data science and modeling. With this approach, a similar or more simple reference algorithm may be developed in order to compare the performance of the reference algorithm to the actual algorithm being audited. It highly depends on the type and complexity of the algorithm that is audited whether this approach is feasible. In addition, it requires the dataset for training/testing from the client to be available.

7 AI Skills and Expertise in the Audit

When the audit plan and specific procedures have been considered and planned, an assessment should be made what skills and expertise are required in order to successfully complete the audit. And although the depth of the audit may vary greatly and may even be very limited, it is important to have, next to a certain level of diversity, the right AI-specific skills and expertise in the audit team to spot and investigate potential problems. The audit team should be able to:

- Recognize unrealistic problem specifications that are not likely to result in safe algorithm use.
- Investigate the origins of the data to spot bias and quality problems in the data.
- Interpret and criticize the metrics used to justify the reliability of the algorithm.
- Perform an exploratory data analysis and interpret the output of common explainable AI (XAI) algorithms.
- Pick and use the right metrics for measuring fairness, and give the measurements a reasonable explanation.

7.1 Realistic Problem Specification

A key skill, maybe even the defining skill, of AI as a discipline is translating real-world problems into problem specifications solvable in information space using an algorithm for that class of information space problems. Bad quality algorithmic solutions generally start with a bad problem conceptualization. Starting from a good business case for an algorithm, a good problem specification operationalizes business performance in such a way that it can be measured and optimized, and clearly outlines the intended use of the algorithmic solution by setting out the conditions that must be met before it can be safely assumed to perform as claimed. The translation of key performance indicators that are relevant to business into measurable indicators for performance is an important source of error.

The auditor judges the documented problem specification for risks and for gaps—important criteria that remain unmeasured and unaddressed. A large part of the review of the solution itself can be interpreted as a comparison between what was specified and what actually happened during development and what actually happens in use. If the problem conceptualization is good, and the algorithmic solution is an optimal solution to the specified problem, and it is used as advertised, the algorithm will generally score well on the integrity pillar.

Let us at this point return to our running example and apply the measures of recall, precision, and *F*-score that were introduced in chapter "Introduction to Advanced Information Technology," Sect. 3.3 of this book. The public body uses precisely these measures to quantify performance and has trained the algorithm to optimize *F*1-score. The public body has decided before development of the algorithm, without argumentation, that an *F*1-score of 0.9 seems acceptable for

performance based on a quick search of $F1$-scores of some other projects, and the algorithm clearly exceeds that benchmark.

There are two fundamental problems here. The first one is the arbitrary benchmark. One should always use a benchmark that is relevant for the task environment. There is no objective answer to what is a good $F1$-score. It depends on the alternatives methods available for making a risk-based selection of applications. The F-score is moreover sensitive to class imbalance, or differences in ratio between the two outcomes in the historical data. Class imbalances vary over projects.

When you are developing a medical diagnostic algorithm, you can often uncover an appropriate benchmark for roughly the same task environment through study of scientific literature. There are after all many hospitals doing roughly the same things. The public body executes a unique task, and has no such option. It has two directions to move in to produce an empirically grounded benchmark:

- Try to create a golden standard dataset of correctly processed application forms and measure the performance of the manual processing department compared to this golden standard dataset. To produce this dataset usually involves assigning multiple employees to the same applications, and spending far more time on it. This may be prohibitively expensive. On the other hand, this golden standard dataset is also useful for researching bias in the historical data.
- Play structured games with employees of the manual processing department or decision makers to determine what distribution of true positives, true negatives, false positives, and false negatives they tolerate. This approach leverages expert knowledge effectively, assuming the employees involved do understand their business well.

The second problem is that $F1$-score as a balanced score of precision and recall weighs false positive selection and false negative non-selections equally heavily as errors. It is a harmonic mean, after all. This is very unlikely to reflect the actual business objectives of the public body. As noted, when we introduced the running example manual processing capacity is scarce, and selecting applications for processing needlessly is a waste of effort. Besides that the organization specifically fears unfairly selecting people for manual processing, and this risk only relates to false positives. It should therefore be concerned with precision much more than recall when measuring performance. Fortunately, it is quite easy to modify the F-score to take a certain exchange rate between recall and precision, to reflect that employees would trade for instance five false negatives for one false positive in a structured gaming situation.

$$F\frac{1}{5} = 1 + \left(\frac{1}{5} \cdot \frac{1}{5}\right)^2 \frac{\text{precision.}\ \ \text{recall}}{\left(\frac{1}{5} \cdot \frac{1}{5}.\ \ \text{precision}\right) + \text{recall}}$$

This generalized F-score can be used for plotting precision against recall for an algorithm's performance to gain insight into what task performances are feasible depending on a chosen exchange rate between precision and recall. For a given task

environment, with an already determined exchange rate, only one point on the curve is important.[6] But the developers of the algorithm often do their work not knowing what that point is going to be.

7.2 Data Lineage

Whether a machine learning solution may be expected to do what it is claimed to do depends considerably on the fidelity with which the training and test data used for its construction reflects the task environment in which it is fielded. When we are forming an opinion about the usefulness of training and test data for an algorithm, we are looking for signs of lack of representativeness of the dataset for the task environment, and for signs of systematic misrepresentation of what actually happened in the task environment in the dataset. The first type of problem is an (inductive) bias problem. The second type is a data quality problem.

The concept of bias is widely applied, to describe (1) lack of representativeness of datasets for an environment, (2) the causes of that lack of representativeness (reporting bias, survivorship bias), and (3) the consequences of that lack of representativeness for decision-making based on the algorithm's output (popularity bias, algorithmic bias, and—as a convenience label—for any unfair decisions caused by bias). Here we limit ourselves to bias as a property of a dataset in a task environment.

If the algorithm used belongs to the class of supervised algorithms, it is trained and tested with data labeled with the (putatively) correct answer. The most obvious technique for researching bias is to compare data used for training and testing with the remaining unlabeled data, for which no correct answer has been determined, in an *exploratory data analysis* or EDA. Judging and performing an EDA is therefore part of the desk research skills one would expect of an audit team. Systematic differences found are in need of an explanation.

The auditor will in addition investigate and sometimes test the processes that created the data to gain insight in bias and quality problems and their causes. Part of these processes—from the master datasets that were sourced for the development process to the datasets that are fed into the algorithm—are under direct control of the developers of the algorithm. This is the data preparation pipeline. The pipeline should be documented well enough to allow for reperformance by an audit team. Bias and quality problems are however often already present in the master datasets that were sourced for development. At some point the audit team will be investigating where this master data came from.

At this point we run into an important scoping question. There are basically two ways in which the lineage of these datasets may be proven (Cheney et al., 2009). In *eager lineage* settings, the data is well-governed and the characteristics of the

[6] A very similar curve, containing similar information, is the ROC curve which plots recall against the true negative rate. This type of curve is more often encountered in documentation.

processes that created it are already routinely well-documented by the data controller. One may for instance expect this in medical settings. Data gathering is supervised by a medical-ethical authority, data management plans will be in place before gathering starts, and the process will be subject to an audit regime. In this case we would have an independent party assuring us of the quality and representativeness of the data. In *lazy lineage* cases research into business practices generating data had to take place within the context of the development of the algorithm because no such assurance already existed. In this case lineage should be fully documented as part of the development process and is clearly subject to investigation by the auditor in an algorithm assurance engagement.

7.3 Reliability of Trained Models

The auditor should understand empirical approaches to determining the reliability of a predictive model through resampling methods, and if necessary, should be able to apply them to the data. The most basic method for estimating performance is a train-test split. This gives us performance statistics, but no insight into how robust that statistic is going to be on new data. Validation of performance should take place on holdout data that was not available to the developers. Ideally the holdout data would be produced in an empirical impact study that is an exact simile of the prospective task environment.

Without access to new data, robustness of the algorithm can still be estimated by the developers and serves an important purpose in itself. The standard approach to showing reliability is to essentially make a lot of randomized train-test splits (cf. resampling methods like cross-validation; Kohavi, 1995). The average and variance of the performance statistics collected in train-test splits gives insight into the reliability of the model—assuming that the data reflects the task environment in which the algorithm will be used.

In addition, it is good policy to test any hypotheses one has about groups or time frames that can be found in the training and testing data in which the predictive model may perform less well to validate the problem specification, to ascertain there are no resilience problems to be expected (cf. so-called *underspecification* problems; D'Amour et al., 2020). One doesn't want to depend on an algorithm that doesn't work in winter, or doesn't work in Amsterdam. Measuring unfairness based on hypotheses about groups that may be treated differently is essentially a special case of this type of hypothesis testing.

7.4 Exploratory Data Analysis and the Use of Explainable AI (XAI) Techniques

While explainability can be considered a core goal of algorithm assurance, and we therefore favor transparent and self-explanatory algorithms, there are cases where either an alternative form of analysis is called for to uncover what the algorithm does, or where a parallel, more explainable algorithm with less performance is built to gain insight into the relation between inputs and outputs of a black box algorithm. The audit team is expected to understand exploratory data analysis and the use of common Explainable AI (XAI) techniques to uncover what the algorithm does. See for an overview of XAI techniques that can be used Linardatos et al. (2020) and for an understanding of the limitations of these as a tool for explainability cf. Lipton (2018). These techniques will occasionally be used by the audit team to gain the necessary insights and to explain its findings. Specifically, the audit team should be able to:

- Compare datasets collected from the same task environment.
- Apply feature selection and extraction methods to gain insight in the relevance of the data to the problem solved by the algorithm.
- Apply XAI methods for gaining insight into what role features play in how the algorithm solves the problem.

7.5 Measuring Fairness

Algorithm fairness is a hot topic, and for clients often a gateway into requesting algorithm assurance. It is moreover a central topic in our running example for this chapter. Making a judgment about fairness starts with identifying which groups or individuals may be differentially treated by an algorithm based on static descriptors. In a well-managed development process, these groups or individuals have been identified with the help of stakeholders during a prospective risk identification, and precautions have been taken to prevent differential treatment of the identified groups or individuals—including a requirement to measure whether the groups or individuals are indeed treated differently by the algorithm.

Identifying unfairness risks with stakeholders starts involves looking at how the output of the algorithm is used in decision-making, and how it affects stakeholders that may be unfairly treated. In a simple binary decision, it is usually simply a matter of deciding which of the four possible outcomes—true or false positive and true or false negative—are usually considered good or bad from the perspective of the stakeholder. If the decision is for instance a medical diagnosis the stakeholder wants the outcome to be true, regardless of whether it is positive or negative. If it is an accept-reject decision the stakeholder wants to be accepted, and will often be

happy to be a false positive. In some cases, both ground truths and outcomes are important.

Usually, we are looking at group fairness for specific, identified vulnerable groups. In rare cases, we may be concerned with unfairness towards individuals. If doors for instance don't open for someone whose face cannot be recognized by an algorithm (yes, this happens), this (1) is unfair, and (2) implicitly characterizes a new vulnerable group of people whose face was not learned by the algorithm. Although we are dealing with individuals, we can find those individuals in the data as a group of successive inputs relating to the same individual, and we can apply the same measurement tools to detect this unfairness to individuals. Fairness risks relating to individuals are usually characterized as *social exclusion* risks.

If the algorithm treats a group or groups of people differently, it is apparently capable of picking the members, or successive inputs relating to members, of the unfairly treated group based on the input data of the algorithm. This input data may contain *proxies* that function as static descriptors of group membership.

Assuming the risk identification is adequate, and static descriptors potentially identifying groups have been identified, measurements should be made to quantify the difference in performance or outcome for these groups. These measurements can be made using hypotheses about what the proxies in the data are for group membership, or by using an external data source not used by the algorithm that directly identifies group membership. If the organization has this external data for measurement of unfairness, it is usually personally identifiable data or sensitive data. Permission for its use will be required.

Although a large number of different measures have been proposed in the literature (Verma & Rubin, 2018), the problem in essence boils down to a simple choice between two approaches. We are either comparing the relative *outcomes* for a pair of groups to see whether the difference is within the organization's tolerance margins for outcome inequality, or we are comparing the relative performance of the algorithm for a pair of groups. Regardless of which choice we make, we do often encounter some difference. It is up to the client to decide whether this difference is tolerable, and what it means.

Let's reconsider our running example again. Using the AI application, the public body wants to know whether bias is present in the algorithm against single father household applicants because the benefits will only be awarded if the parent takes care of children the majority of the time.

As pointed out earlier, in the public body example case the two possible outcomes—being manually or automatically processed—are perceived as a punishment vs. reward scenario. Where earlier we addressed making a smart choice in which performance statistic to look at, we now address a similar problem with fairness statistics: which one is meaningful for the problem at hand.

The comparison that matters in this case is mainly the outcome: if it is fairly equal for both groups, there is little risk that fairness issues will be raised. The measure of choice will therefore be *statistical parity* (or *group fairness*; cf. Verma & Rubin, 2018): the probability of being manually processed is equal for both groups:

Table 5 Confusion matrix for the running example

		Predicted outcome of manual processing	
Actual outcome	Total population	**Predicted positive**	**Predicted negative**
	Applicants: 100		
	Single fathers: 10		
	Positive	True positive	Applicants: 80
		Applicants: 10	Single fathers: 6
		Single fathers: 3	
	Negative	False positive	(Distribution between false negatives and true negatives is unknown)
		Applicants: 10	
		Single fathers: 1	

$$\frac{\text{True positive} + \text{False positive}}{\text{Total Population}}$$

This measure is crude, but also one likely to be used by the media to support an accusation of unfairness. The algorithm does not use the gender of the applicant, but the public body does have access to data about the gender of the applicant and household composition from a third party. We can therefore set up *confusion matrixes* for the single father household vs. the rest to gain insight (see Table 5). Ideally, we would like to be able to fill in all four conditions, including the distinction between true negatives and false negatives, but for the negative predictions we don't have information about what the outcome of manual processing would have been.

A quick calculation shows that there is indeed a sizable outcome inequality as expected:

$$\frac{3+1}{10} = 0.4 \text{ vs.} \frac{10+10}{100} = 0.2$$

To justify that difference, it remains relevant to assess the relative accuracies for both groups. Only when the algorithm performs equally well for both groups, the difference can be accepted as a matter of fact. Although it is in principle possible to calculate and compare the weighted *F*-scores, it is more common to compare the precision scores (explained in chapter "Introduction to Advanced Information Technology," Sect. 3.3 of this book). We don't know the distribution between true and false negatives after all. In the context of assessing the problem specification we made the same choice. In the context of fairness, this comparison is labeled

predictive parity (Verma & Rubin, 2018). A quick calculation shows that precision for the group of single father households appears to be even higher than for the total population of applicants, assuring that the root cause of the difference is most likely in the datasets used for training and testing.

$$\frac{3}{3+1} = 0.75 \text{vs.} \frac{10}{10+10} = 0.5$$

Since the number of applicants in the single father household is rather low, we don't have reason to be confident about that conclusion. Ideally one would advise to gather some more data about the group of single father households, but that is obviously going to be difficult: only time will tell. In any case, the audit team neutrally reports differences, possible root causes of those differences it uncovered, and possible ways of removing or reducing those differences, for instance with the help of debiasing algorithms to reduce outcome inequalities (Agrawal et al., 2020). Debiasing should only be used in the understanding that optimizing equality for one type of measure usually worsens the other given the same, unchanged training and test datasets. The bias that caused the unfairness is still embedded in the data in some way. Besides that, if used unwisely, debiasing algorithms may introduce unfairness towards other groups, and may in certain cases be judged unlawful (Xiang & Raji, 2019). The reason for this is simple: giving a specific group a push in the back by definition disadvantages everybody else.

8 Discussion

In this chapter, we have presented a structured approach to define an audit plan for algorithm assurance, based on knowledge from scientific and popular literature and practical experience. Despite our aim to be as comprehensive and detailed as possible, the fact remains that this chapter is fully based on our knowledge and experience as assurance providers in a newly developing field. In this section, we discuss three critical pointers in order for algorithm assurance to mature.

8.1 Transparency and Standardization

Algorithm auditing as a profession is still young. In order for it to become mature profession, it needs, besides more scientific research, shared practical experiences from the field. This calls for a shared learning environment to everyone's benefit. The time of practitioners re-inventing their own wheel is over, especially because the increasing impact of algorithms requires systemic oversight, and governments increasingly realize that it does. Auditors can play a significant role in creating trust, but only if they agree on how algorithm auditing should work.

Standardization would be a logical next move up in the algorithm auditing maturity curve. Firstly, this will help the auditee to understand what is being audited. Even more importantly: one auditor's outcome would be the same as the outcome of another auditor, because the same methodology is followed. Secondly, it also helps to put expectation management in place. What may an auditee, or the receiver of the algorithm assurance report, expect from the auditor and what degree of assurance can the receiver get from the audit report? We truly believe that existing professional associations such as the International Auditing and Assurance Standards Board (IAASB)[7] of auditors have to play a crucial role. But auditors themselves should be open to the approach they follow as well.

The main complication is the diversity of task environment algorithms operate in. One size fits all solutions may impose a cost of control on developers and operators of algorithms that exceeds the business value of many trivial algorithm applications. It is likely that auditor specializations will develop over time for specific high-risk areas governed by different areas of law (medical device safety, consumer rights and legal liability for harm, financial reporting, privacy law, etc.) if standardization is to go deeper than the level of principles.

8.2 Skills and Expertise

In Sect. 7 of this chapter, we have described the specific skills that are required to successfully perform an algorithm audit with the required level of depth. We believe that existing (IT) auditors today do not have this skill set. Yet using the same criteria is just one aspect. Spotting the same risks is an entirely different one. It might be worth a discussion whether specific individual accreditation is required in order to perform algorithm audits.

8.3 Auditing AI with AI

A topic that we didn't discuss in the chapter is how AI technology can also help to perform AI audits. Although this is a fairly new topic, it is worth exploring. The use of AI technology to mitigate risk or exercise control on AI is a lively field. When talking about explainability, or fairness, many in the field of AI immediately think of the research into how to do these things automatically. Obviously. We have looked at a standard audit approach, including all the relevant methodological aspects that are part of it. This approach will not go away: behind any important automated control solution there will be auditor signing off on it. But it is possible to look

[7] https://www.iaasb.org/

beyond control automation and think of AI solutions to general purpose adversarial testing of algorithms in specific domains, for instance vision.

9 Conclusions

Based mainly on the professional experiences of the authors, we introduced the field of Algorithm Assurance in the audit practice. In the context of algorithm assurance, we use a non-standard meaning of the concept of an algorithm: The object of the audit is a computer program, or component of a computer program, containing implementations of a risky AI algorithm or algorithms, to be reviewed in the context of a task in which it is applied *or* a prospective class of tasks in which it may be applied. We distinguished a number of task environment types in which such computer programs may be encountered in an audit context, and the reasons why they may be subject to an audit.

After that we have successively laid the scope of an assurance engagement, the control objectives or principles that guide the assurance engagement, the risk assessment, audit strategy and action plan, and the typical AI-related skills and expertise required of the auditor to do an in-depth investigation of an algorithm.

The main area in which algorithm assurance is still under development is in standardization of what is being tested and how. Standardization is essential for the development of trust in algorithm assurance. The main problem in this area is the diversity of task environments to take into account, which may lead to the development of specializations in the field.

References

Agrawal, A., Pfisterer, F., Bischl, B., Chen, J., Sood, S., Shah, S., Buet-Golfouse, F., Mateen, B. A., & Vollmer, S. J. (2020). *Debiasing classifiers: Is reality at variance with expectation?* Retrieved from https://ssrn.com/abstract=3711681 or https://doi.org/10.2139/ssrn.3711681

Brown, S., Davidovic, J., & Hasan, A. (2021). The algorithm audit: Scoring the algorithms that score us. *Big Data & Society, 8*(1), 205395172098386. https://doi.org/10.1177/2053951720983865

Cheney, J., Chiticariu, L., & Tan, W. C. (2009). *Provenance in databases: Why, how, and where*. Now Publishers.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., ... & Sculley, D. (2020). *Underspecification presents challenges for credibility in modern machine learning*. arXiv preprint arXiv:2011.03395.

English, R. (2021, July 26). Discriminatory basis of child tax credit is justified, rules supreme court. *UK Human Rights Blog*. Retrieved March 23, 2022, from https://ukhumanrightsblog.com/2012/05/17/discriminatory-basis-of-child-tax-credit-is-justified-rules-supreme-court/

European Commission. (2019, December). *The assessment list for trustworthy artificial intelligence*. Retrieved from https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Ewals, R., Francot, J., Frins, C., Houtekamer, D., Van Helden, M., Matto, J., Boon, R., Meulendijks, J., & Bruggeman, A. (2019, December). Handreiking voor SOC 2® en SOC 3® op basis van ISAE3000 / richtlijn 3000A. NOREA. Retrieved from https://www.norea.nl/nieuws/6509/nieuwe-handreiking-voor-soc2-en-soc3-rapporten

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (Vol. 2, no. 12, pp. 1137–1143). Morgan Kaufmann. CiteSeerX 10.1.1.48.529.

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. *Entropy, 23*(1), 18.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*(3), 31–57. https://doi.org/10.1145/3236386.3241340. ISSN 1542-7730.

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D. & 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* (pp. 33–44).

Shen, H., DeVos, A., Eslami, M., & Holstein, K. (2021). Everyday algorithm auditing: Understanding the power of everyday users in surfacing harmful algorithmic behaviors. *Proceedings of the ACM on Human-Computer Interaction, 5*(CSCW2), 1–29.

Van Smeden, M., Moons, C., Hooft, L., Kant, I., Van Os, H., & Chavannes, N. (2021, December). *Guideline for high-quality diagnostic and prognostic applications of AI in healthcare*. Ministry of Health, Welfare and Sport. Retrieved from https://www.datavoorgezondheid.nl/documenten/publicaties/2021/12/17/guideline-for-high-quality-diagnostic-and-prognostic-applications-of-ai-in-healthcare

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

Wirtz, B. W., Weyerer, J. C., & Kehl, I. (2022). Governance of artificial intelligence: A risk and guideline-based integrative framework. *Government Information Quarterly*, 101685.

Xiang, A., & Raji, I. D. (2019). *On the legal compatibility of fairness definitions*. arXiv preprint arXiv:1912.00761.