Tampere University

Carita Logrén

TOWARDS QUALITY ASSURANCE OF MACHINE LEARNING SYSTEMS

Faculty of Information Technology and Communication Sciences
Master of Science Thesis
October 2020

ABSTRACT

Carita Logrén: Towards Quality Assurance of Machine Learning Systems
Master of Science Thesis
Tampere University
Information Technology
October 2020

Too often the quality of a machine learning system is measured by the quality of the machine learning model's predictions. Academics and practitioners have awoken to the latent quality issues in machine learning systems only during the past few years, as machine learning is increasingly applied in complex and safety-critical domains. While research on the topic is emerging, its focus is scattered across different tracks, and the cohesive view is missing.

The purpose of this work is to increase the knowledge around the prominent machine learning quality characteristics and their assurance in order to help the machine learning practitioners in developing reliable and safe systems. With the groundwork of a comprehensive review of the current research tracks in machine learning, semi-structured interviews with 13 professional machine learning practitioners were conducted with the aim of clarifying the important quality considerations in machine learning systems as well as of identifying the quality assurance activities that have the biggest positive impact on the quality of the developed systems.

Seven important machine learning system characteristics were ascertained from the interviews, including machine learning service quality, interpretability, fairness, and development ethics. While research over these domains exist, the findings reveal that the issues faced in practical work do not fully reflect the problems studied in the literature, and certain perspectives, such as service validity and development ethics, have gained little attention from the academia before. Aside from the quality characteristics, five meaningful practices for quality assurance were identified: data and domain understanding, design, verification and validation, documentation, and engineering practices. In particular, the engineering practices appear to have a significant impact on the quality of the machine learning development work in general.

The findings also reveal that currently a major part of the quality issues with which the industry struggles are originated from the lack of systematic development methodologies, causing the research advancements in the machine learning characteristics to be left unapplied. The results highlight the need for collaboration between the machine learning developers, organisations, and academia in the development, validation, and education of a systematic methodology for machine learning engineering.

Keywords: machine learning quality, quality assurance, machine learning methodology, machine learning systems

TIIVISTELMÄ

Carita Logrén: Koneoppivien järjestelmien laadunvarmistuksesta
Diplomityö
Tampereen yliopisto
Tietotekniikka
Lokakuu 2020

Koneoppivan mallin ennusteiden tarkkuutta käytetään yleisesti synonyyminä koko koneoppivan järjestelmän laadulle. Tämä käsitys laadusta ei ole riittävä, erityisesti kun koneoppimista sovelletaan enenevissä määrin turvallisuuskriittisissä ympäristöissä. Koneoppivien järjestelmien laatuongelmiin on kuitenkin havahduttu vasta viime vuosien aikana, minkä seurauksena keskustelu ja tutkimus näiden järjestelmien laadusta on hyvin pirstaloitunutta.

Tämän työn tarkoituksena on lisätä tietämystä merkittävistä, koneoppimista koskevista laatunäkökulmista sekä niiden varmistuksesta, jotta teollisuudessa kehitettävät koneoppivat järjestelmät olisivat luotettavia ja turvallisia. Työssä tehtiin laaja kirjallisuuskatsaus ajankohtaisiin, tutkittavana oleviin koneoppivien järjestelmien ominaispiirteisiin, ja sen perusteella suunniteltiin ja toteutettiin teemapohjainen haastattelu 13:lle koneoppivien järjestelmien parissa työskentelevälle ammattilaiselle. Haastatteluilla selvennettiin käytännön työssä kohdattavia laatuongelmia sekä niitä toimia, joilla on merkittävin positiivinen vaikutus kehitettäviin järjestelmiin.

Haastatteluiden perusteella varmistui seitsemän tärkeää koneoppivien järjestelmien laatupiirrettä, muun muassa koneoppivan järjestelmän palvelun laatu (service quality), selitettävyys (interpretability), oikeudenmukaisuus (fairness) ja kehitystyön eettisyys. Vaikka tutkimusta näistä teemoista onkin jo olemassa, haastatteluissa paljastui, että tämänhetkinen tutkimus ei täysin heijasta käytännön työssä kohdattavia ongelmia, ja eräitä piirteitä, kuten palvelun validiteettia ja kehitystyön eettisyyttä, ei juurikaan ole käsitelty kirjallisuudessa aiemmin. Laatupiirteiden ohella haastattelumateriaalista voitiin tunnistaa viisi merkittävää laadunvarmistuksen osa-aluetta: ymmärrys datasta ja toimialueesta (data and domain understanding), suunnittelu (design), varmennus ja validointi (verification and validation), dokumentaatio sekä kehityskäytännöt (engineering practices). Erityisesti juuri kehityskäytännöillä vaikuttaa olevan huomattava merkitys koneoppivien järjestelmien laatuun.

Tutkimuksen tulokset paljastavat myös, että tällä hetkellä suurin osa koneoppivien järjestelmien kehityksessä kohdattavista ongelmista johtuu kehitysmenetelmien puutteesta, minkä seurauksena tutkimusmaailman edistysaskeleita laatupiirteissä ei päästä hyödyntämään. Työn tulokset korostavatkin koneoppivien järjestelmien kehittäjien, organisaatioiden sekä akateemisen maailman yhteistyön merkitystä koneoppivien järjestelmien tuotantomenetelmien kehityksessä, validoinnissa ja koulutuksessa.

Avainsanat: laatu koneoppivissa järjestelmissä, laadunvarmistus, kehitysmenetelmät koneoppimisessa, konenoppivat järjestelmät

PREFACE

> *Stuckness shouldn't be avoided. It's the psychic predecessor of all real understanding.*
>
> — Robert M. Pirsig, *Zen and the Art of Motorcycle Maintenance*

I would like to express my gratitude to my thesis instructor Matti Nelimarkka for his feedback during the writing process and for the examiner Hannu-Matti Järvinen for the final comments for the work. Above all, this study would not have been possible without my wonderful interviewees, whose insights will hopefully stick in your mind as they did in mine.

In Tampere, 3rd October 2020

Carita Logrén

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CCPA	California Consumer Privacy Act
CRISP-DM	CRoss-Industry Standard Process for Data Mining
GDPR	General Data Protection Regulation
IEEE	The Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ML	machine learning

1 INTRODUCTION

Machine learning is widespread today. Its effectiveness has been demonstrated in a variety of domains, ranging from everyday applications, such as speech recognition (Torfi et al. 2020) and recommendation engines (S. Zhang et al. 2019), to complex, high-risk systems in finance (Ozbayoglu, Gudelek, and Sezer 2020), transportation (Grigorescu et al. 2019), and healthcare (Hong et al. 2020). Although machine learning techniques have only recently become employed for more critical tasks in real-world systems, problems in reliability, safety, and trustworthiness have started to emerge. For example, the lack of robustness in autonomous vehicles has already led to accidents, fatal at worst (D. Lee 2016; The Guardian 2018). Voice commands inaudible to human ears have been demonstrated to be able to control devices through Amazon's Alexa, Google Assistant, and Apple's Siri (C. S. Smith 2018), questioning the security of such applications. Biased algorithms used to allocate healthcare resources have been found to unfairly discriminate black patients (Obermeyer et al. 2019), and unprotected medicine dosage models have been shown to risk privacy by being vulnerable to revealing genetic markers of patients (Fredrikson et al. 2014).

In this light, the increasing ubiquity of machine learning draws attention to the need for ensuring the robust, safe, and fair behaviour of the systems using it. Namely, it seems necessary that the quality assurance of these systems is given more consideration. Unfortunately, to date, the literature knows no generally accepted approaches to support this – the previous academic efforts related to the topic have been limited in the scope of proposed approaches or breadth of considered machine learning characteristics (e.g., Ma et al. 2018; Nakajima 2018). While practices can be drawn from classical software engineering, it has been noted that classical software does not reflect machine learning development well (Ma et al. 2018), consequently being of limited use for assuring the quality of machine learning systems.

In fact, it is often not clear, which of the machine learning characteristics are relevant to consider in the first place – the literature over quality in the context of machine learning has long focused on benchmarking the machine learning models' predictions (Ma et al. 2018; Wagstaff 2012), which is arguably a narrow perspective considering the expectations that a production scale machine learning system is likely to face. While research on different quality characteristics in machine learning is continuously increasing, its focus is scattered across different tracks, and the cohesive view is missing. Few studies have considered machine learning systems pervasively (e.g., Nishi et al. 2018; Villalobos, Fer-

rer, and Alba 2018; J. M. Zhang et al. 2019), and the definition of quality in machine learning remains vague.

Another limitation of the existing literature is that the research on the topic has nearly invariably been based on the theoretical grounds of machine learning as well as previous papers discussing the topic, leaving a question of how accurately the research reflects the problems encountered in the real-world machine learning development. Indeed, a few studies have investigated with empirical methodology how the different machine learning characteristics appear for the industry practitioners (Baier, Jöhren, and Seebacher 2019; Holstein et al. 2019), revealing a potential discrepancy between the research world and industry. Consequently, it is uncertain, whether the practitioners of machine learning find the course of the current research beneficial.

The purpose of this work is to help the machine learning developers in developing reliable and safe systems by increasing the knowledge around the different quality characteristics and their assurance in the context of machine learning. As the quality problems already are faced in the industry at practical machine learning development work, this study is based on the hypothesis that the practitioners have also developed implicit knowledge about the prominent quality characteristics and about the prevention of issues related to them. Consequently, this work contributes to the research and industry by gathering and sharing that knowledge. This is done by collecting experiences related to quality issues and quality assurance from machine learning developers working on real-world systems. Semi-structured interviews are conducted with 13 professionals from 6 different companies. By analysing the empirical, qualitative material resulting from the interviews, this study aims at answering the following two research questions:

> ***RQ1**: What quality issues are faced in the real-world machine learning development work?*
>
> ***RQ2**: Which types of quality assurance have the biggest impact on the quality of machine learning systems?*

The remainder of this work is constructed as follows: Chapters 2 and 3 provide background for this study by investigating how quality is defined and assured in classical software, and by introducing the most prominent machine learning characteristics in the literature, respectively. The used methodologies for the interviews and analysis of the empirical material are described in Chapter 4. The findings revealed by the analysis are then presented in Chapter 5. Chapter 6 reflects on the findings, and discusses the results in relation with the overlapping, prior work. Also the reliability and validity of this work are considered. Chapter 7 concludes the thesis.

2 QUALITY WORK IN SOFTWARE ENGINEERING

Quality, in the field of machine learning, is not comprehensively defined, making it difficult to specify a frame in which different characteristics of machine learning are examined. Fortunately, the situation is different in classical software[1] engineering, where the research and practices have had much more time to mature. Being essentially software systems, machine learning applications are often compared to classical software, and therefore, before diving into machine learning, it seems useful to take a look into what is considered quality in classical software systems and what measures can be taken in order to assure the production of quality systems. The following sections introduce quality and quality assurance in the context of classical software, and looks then into the similarities and differences between machine learning and classical software.

2.1 Quality in software engineering

What is quality? The question is multifaceted: for example, a loan applicant from an ill-reputed neighbourhood perceives the quality of a bank's risk estimation system from a very different angle than the engineer that developed the system. Moreover, the official at the bank might find it difficult to answer the applicant's inquiries for decision rationale, if the risk level is everything that the system gives out. While the system performance indicators can indicate a high operational quality, the loan applicant and official at the bank might perceive the quality of the system very low.

The problem is, of course, not new. One of the most famous takes on defining quality has been the five perspectives proposed by Garvin in 1984. These five perspectives, described also briefly in Table 2.1, are the transcended, product-based, user-based, manufacturing-based, and value-based approaches. Garvin also argues why several definitions for quality are needed — for example, by the manufacturing-based definition, the above-mentioned bank's risk estimation system is a high-quality product, if the requirements stated that no more than a risk level for each applicant must be available, although the subjective experience from using the system, captured by the user-based definition, might turn out bad. In fact, Garvin (1984) suggests that it is necessary to shift the quality perspective as the product moves from design to production.

In addition to the universal attempts to describe quality, also standardised definitions for

[1] This work uses the expression *classical software* to refer to software where the computation logic is completely determined and described by the programmer.

Table 2.1. *Garvin's (1984) approaches to defining quality.*

Transcended approach	Quality cannot be defined, but it can be recognised when met.
Product-based approach	Quality is a directly measurable attribute, such as number of features.
User-based approach	Quality is defined by the subjective experience, "fitness for use".
Manufacturing-based approach	Quality is measured as the "conformance to requirements" specified during design.
Value-based approach	Quality is defined with respect to the costs and perceived value.

the quality of software specifically exist. For example, ISO (International Organization for Standardization) and IEEE (The Institute of Electrical and Electronics Engineers) articulate *software quality definition* as:

1. *degree to which a software product satisfies stated and implied needs when used under specified conditions*
2. *degree to which a software product meets established requirements*

(ISO/IEC/IEEE 24765 2017). In other words, according to the standard, quality means that the specification of a system captures the needs of all stakeholders – whether the needs are explicit or not – and that the system is implemented correctly so that it conforms to the specification. The first part of the definition encompasses Garvin's user-based, product-based, and value-based approaches, which must result into a set of requirements for the software. The second part clearly takes the manufacturing-based approach, focusing on meeting the predefined requirements. Notably, by this definition, the quality of a software product depends on how accurately the specified requirements capture the real needs of the stakeholders (ISO/IEC/IEEE 24765 2017).

Yet the question remains – how to ensure that the system specification indeed comprises all the necessary requirements? The most practical deconstruction for software system quality is perhaps a quality model. For example, the "ISO/IEC 25010 Systems and software Quality Requirements and Evaluation" (SQuaRE) standard defines the *ISO 25010 Software product quality model* that is composed of eight quality characteristics, as illustrated in Figure 2.1 (ISO/IEC 25010 2011). The characteristics of the quality model describe the important quality attributes for a software system, and these attributes are used to define the concrete requirements for a software product (Galin 2004).

2.2 Quality assurance

Quality assurance is a vaguely used expression, especially in the context of software. Most often it is used as a synonym for testing (Laporte and April 2018, p. xvi), although the quality assurance activities comprise much more than the verification and validation

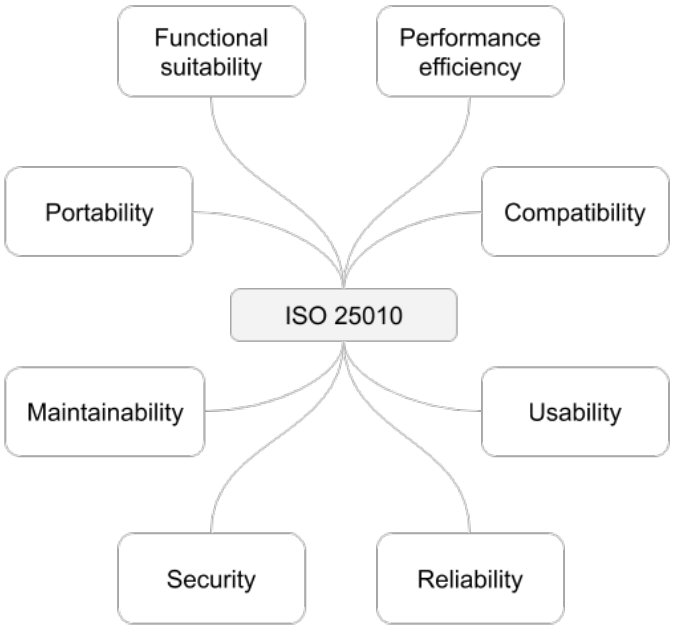

Figure 2.1. *ISO 25010 Software product quality model characteristics (ISO/IEC 25010 2011).*

of the system only. For example, IEEE defines software quality assurance as a "set of activities that define and assess the adequacy of software processes to provide evidence that establishes confidence that the software processes are appropriate for and produce software products of suitable quality for their intended purposes" (IEEE Std 730-2014 2014). In other words, quality assurance consists of all planned and systematic actions carried to produce software that meets the implicit and explicit quality expectations, including not only the activities that assess and control the software quality but also those that support building the software in a suitable manner (IEEE Std 730-2014 2014).

Why is quality assurance needed in software systems? Galin (2004, p. 4) summarises the rationale in the following way:

> *No developer will declare that its software is free of defects.*

This claim is attributed to the high complexity and invisibility of software, and furthermore to the temporally limited opportunities for detecting defects before shipping the product (Galin 2004, pp. 4–7). Consequently, the field of software engineering has adopted quality assurance as a crucial part of the discipline. Software quality assurance aims at ensuring that the activities carried during the system life-cycle support building a software product that meets its quality expectations (e.g., Galin 2004, pp. 25–28; S. Wagner 2013, p. 19).

As mentioned above, a quality model, or a predefined set of quality characteristics, is often used as a starting point for system requirements specification (e.g., Galin 2004; S. Wagner 2013). The purpose of this quality model is to support designing the software

in a way that all of the important quality characteristics are considered in the requirements (Galin 2004, pp. 51–52). Quality assurance then defines the activities that support achieving these quality goals. Also the use of a quality model can already be considered a quality assurance activity. (e.g., Galin 2004; Tian 2001; S. Wagner 2013)

Apart from the quality model, there are multiple deconstructions for quality assurance. For example, Galin (2004) classifies quality assurance activities into *pre-project* components and *project life-cycle* components that are supported by other components; namely quality infrastructure, quality management, standards, and organisational base. S. Wagner (2013, pp. 19–21) makes a clear division between the *constructive* quality assurance activities, which support implementing the software product correctly, and the *analytical* quality assurance activities, which focus on evaluating the quality of the implemented software. Tian (2001) takes a *defect centered* view on software quality assurance, in which the focus is on software defects and activities are classified into *preventive*, *reductive*, and *containing* measures based on where the defect is managed.

Despite the various definitions of these approaches, often the quality assurance components in each definition have corresponding activities in those described by others. The most notable deviant is perhaps the *defect centered* view, which implicitly emphasises the functional correctness of the system. It also clearly includes failure prevention (i.e., defect containment) as one of the quality assurance activities (Tian 2001). However, the most known software quality models, ISO 25010 and McCall's quality factor model, contain the notions of fault tolerance and recoverability already within the quality characteristics, and consequently failure prevention is not explicitly considered in the quality assurance activities (e.g., Galin 2004, pp. 49–50; Laporte and April 2018, pp. 69–84).

In the following, the most common quality assurance components are grouped under S. Wagner's (2013) two categories.

Constructive quality assurance activities

Constructive activities in quality assurance facilitate building a system that meets its requirements (S. Wagner 2013, pp. 19–21). It comprises the actions taken before building the system as well as the processes and tools that support preventing problems and achieving the quality requirements during the system implementation and maintenance.

The foremost activity in this category is perhaps *planning the quality* and also *planning how to assure quality* (Galin 2004, pp. 95–108; S. Wagner 2013, pp. 91–110; Laporte and April 2018, pp. xvii–xix). These plans are necessary, for example, for designing the system, scheduling the development activities and implementing the quality control activities. The development and quality plans are also required for compliance with several quality management standards. (Galin 2004, pp. 96–97) As described above, the plans are often based on a quality model or predefined set of quality attributes that comprise all the characteristics that might be important to consider in the system design.

The tools that support building the systems correctly form another important group of

activities under constructive quality assurance. These tools can be software, such as a specific programming language (Tian 2001) or an integrated development environment (IDE) (S. Wagner 2013, p. 129). Also standards can be considered as tools in quality assurance (Galin 2004, p. 59; Laporte and April 2018, pp. xvii–xix). Similarly, all guides, templates, and checklists used by an organisation can be seen as quality assurance tools (Galin 2004, p. 66).

A third group of constructive quality assurance activities support managing the "human component" during the system lifecycle. These activities include, for example, engineering methodologies, processes, and practices, as well as training for the required domain-specific knowledge (Tian 2001, Galin 2004, p. 59; Laporte and April 2018, pp. xvii–xix). Notably, in recent years, organisations have increasingly adopted the *DevOps* (Development Operations) approach in software development (Erich, Amrit, and Daneva 2017; Leite et al. 2019). DevOps attempts to "automate continuous delivery of new software versions, while guaranteeing their correctness and reliability" (Leite et al. 2019), and the practice necessitates collaboration or merging of the software development, operations, and quality assurance skills (Erich, Amrit, and Daneva 2017; Leite et al. 2019). Although there has not been enough research to validate the effectiveness of DevOps (Leite et al. 2019), it is widely in use (Bordeleau et al. 2019).[2]

Analytical quality assurance activities

Analytical activities in quality assurance aim at assessing the quality of the system against the requirements as well as detecting and removing problems in the system (S. Wagner 2013, pp. 19–21). The analytical activities can also be understood as *quality control*, according to the IEEE definition (IEEE Std 730-2014 2014). The importance of analytical quality assurance increases along with the system complexity and invisibiliy (Tian 2001, Galin 2004, pp. 4–7).

Analytical quality assurance has two primary categories of activities. The first one of them, and possibly the best known category of quality assurance activities, consists of verification and validation (V&V) – often described as *building the system right* and *building the right system*, respectively (Laporte and April 2018, pp. 251–254). These activities are usually performed as testing at different levels of the system, including white-box testing; such as unit and integration tests; as well as black-box testing; for example operational system tests and usability tests (Tian 2001; Galin 2004, pp. 209–211; Laporte and April 2018, pp. 251–254).

The second category consists of human efforts for analysing the system quality. These activities include reviews and inspections, which focus on assessing the quality of the software, documentation, or other system artifacts (Tian 2001; Galin 2004, pp. 61–62; Laporte and April 2018, p. 169). While reviews and inspections are usually conducted within the development team, it is also possible to acquire external expertise, so called

[2] DevOps approach can also be applied in machine learning development, where it is generally called *MLOps* (machine learning operations) (e.g., Saucedo 2019).

"expert opinions", for difficult scenarios (Galin 2004, pp. 62–63, 170).

2.3 Relation to machine learning

Although machine learning systems are also software systems, the different programming paradigm of machine learning introduces new types of issues that do not correspondingly exist in classical software world. Consequently, while the *definition of quality*, as proposed by ISO and IEEE, does seem applicable for machine learning systems, the deconstruction of quality, i.e., the general software quality models, seem inadequate for defining the requirements for a machine learning product. This claim is supported by the fact that both ISO and IEEE have ongoing standardisation efforts for artificial intelligence applications. Unfortunately, the scope of the first standards seems to be limited: the focus is more on ethics and market efficiency, and less on, for example, accountability and safety (Cihon 2019). Moreover, as standardisation depends on the level of maturity of the technology (Cihon 2019), the work for a comprehensive set of standards will most likely continue for some time still. Consequently, an existing, generally accepted deconstruction for machine learning system quality is not available for this work, and instead, a new enumeration of the machine learning specific characteristics, providing perspectives for quality, is proposed in the following chapter.

While quality is ill-defined in the field of machine learning, *quality assurance* is discussed in the literature even less. In fact, more often the expression has been used in studies that were focusing on a subset of the definition: *testing* (e.g., Nakajima 2018). Apart from testing frameworks, only process models for data science development have been proposed, including the *Secure Deep Learning Engineering Life Cycle* (SDLC) model by Ma et al. (2018) and the older *CRISP-DM* (CRoss-Industry Standard Process for Data Mining) process model by Chapman et al. (2000) (see Figure 2.2). However, these models do not address the different characteristics of machine learning comprehensively, and moreover, they do not consider quality assurance in particular.

In conclusion, no prior reference for machine learning quality assurance exists. However, the problems necessitating software quality assurance that were discussed above appear to be further exacerbated in machine learning systems: the size of the input space and probabilistic nature of machine learning further add to the system complexity and opaqueness. Moreover, these properties of machine learning also complicate the quality assurance activities (Ma et al. 2018). Although the existing software quality assurance practices and tools are not directly usable in machine learning systems (e.g., Ma et al. 2018), the deconstruction of the quality assurance activities, as described in the previous section, seems to provide an applicable framework for machine learning as well. Consequently, these activities are used as a guidance for classifying the practices that are studied with the interviews. A more detailed description of how the framework of quality assurance is used in the empirical part of this work is given in Chapter 4.

The introduction of this chapter mentions that the field of software engineering is much

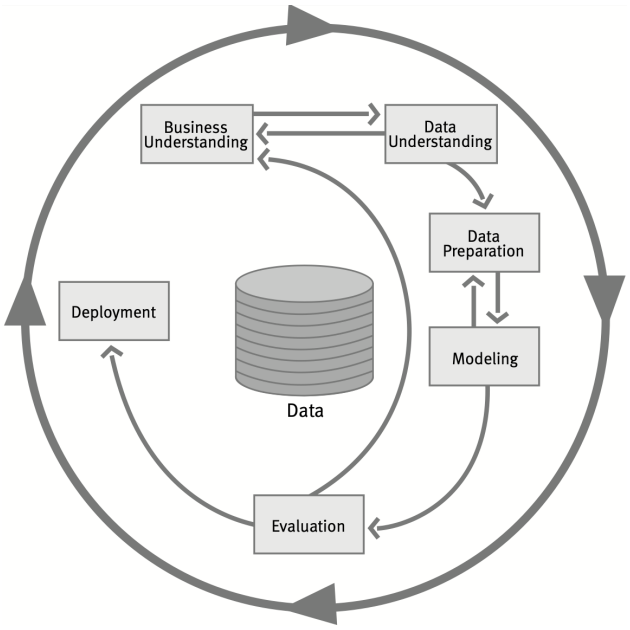

Figure 2.2. *The CRISP-DM process model (reprinted from Chapman et al. 2000).*

more mature in comparison to machine learning. As demonstrated, the software world has quite established, comprehensive perspectives into quality and its assurance. This systematicity and thoroughness of software engineering world guides also this work, which aims at considering the two research questions from a similarly wide perspective.

3 CHARACTERISTICS OF QUALITY IN MACHINE LEARNING: A REVIEW

Until recent years, the machine learning research has focused predominantly on improving the performance metrics of machine learning models (Ma et al. 2018; Wagstaff 2012). Perhaps for that reason, quality of machine learning systems has long connoted the quality of the *predictions* of the models in these systems, although clearly this is a very narrow perspective for system level quality. Even today, the word “quality” rarely has any other meaning in the context of machine learning in academic literature. As shown in this chapter, research on the different characteristics of machine learning has lately emerged, but its focus is split into separate tracks, and no general image is available.

The previous works that have discussed machine learning quality at a system level have largely focused on a subset of the system characteristics. For example, Nakajima (2018) considers only the robustness of and verification methods for the machine learning model predictions. Villalobos, Ferrer, and Alba (2018) use the ISO 25010 Product quality model for software to derive the quality characteristics to assess, and consequently, the work captures only issues relevant to the machine learning software. By far, the most pervasive view on the different machine learning characteristics has been taken by J. M. Zhang et al. (2019) in a survey for machine learning testing. In the work, the authors discuss the properties of machine learning that appear in the literature concerning testing, including correctness, overfitting, robustness and security, efficiency, fairness, and interpretability. Also Ma et al. (2018) mention several characteristics in machine learning, although their work later focuses on secure deep learning only.

Having a wider understanding on the relevant quality considerations in machine learning seems a prerequisite for comprehensive quality assurance work. Following the convention of the software quality assurance models, as well as of the previous works, this thesis first gives an overview of the perspectives into machine learning quality that should be addressed in the quality assurance activities. However, as the machine learning specific standardisation effort is still ongoing (ISO 2020), there is no generally accepted reference for the important quality characteristics in machine learning, and instead, the characteristics are drawn from the most prominent research tracks present in machine learning literature. These characteristics, described in the following sections, are later used to structure the interviews, as discussed in more detail in Chapter 4.

3.1 Technical Quality

The previous works that have considered machine learning at a system level (e.g., Nakajima 2018; Sculley et al. 2015; Villalobos, Ferrer, and Alba 2018) have almost invariably focused on what is called *technical quality* in this thesis. Technical quality views the machine learning system as any other software system, and intelligence in this perspective is merely regarded as just another feature of the system. Therefore, a software quality model can be used to plan and assess the quality of the system at technical level. Notably, here machine learning prediction quality is considered as a part of the technical quality of a system, because improving and maintaining the correctness of the machine learning model is seen primarily as a technical problem.

This work uses the ISO 25010 Software product quality model, discussed in the previous chapter, as a base model for the technical quality of a machine learning system. In the context of machine learning, however, certain characteristics in this quality model have gained more traction in the literature than the others, and this section discusses only those [1]. In the following, the technical quality of a machine learning system is considered as of its functional suitability, performance efficiency, maintainability, and portability.

Functional suitability

Machine learning is employed in applications in which logic cannot be implemented with traditional software development tools (e.g., Sculley et al. 2015). The programming paradigm of machine learning is inherently different from that of traditional software, because the logic of the system is largely determined by data (e.g., Ma et al. 2018). This makes machine learning both powerful but also characterised by uncertainty, complicating the verification and validation of the system's functional suitability, or more precisely, its functional *correctness*. Although *accuracy* is perhaps the most common evaluation metric for correctness in machine learning, the choice of metric largely depends on the application, data, and type of machine learning task at hand (e.g., classification or regression) (Dinga et al. 2019; Japkowicz 2006). To avoid confusion, this work will follow J. M. Zhang et al. (2019) with the notation *correctness* for referring to the score of any evaluation metric(s) that the model is trying to optimise.

The existing verification means of traditional software systems are not apt for ensuring the correctness of machine learning systems because of their incapability of adapting to, i.a., shifts in the system behaviour, size of the input space, indeterminism of the output, and the latent internal state of machine learning models (Ma et al. 2018; Marijan, Gotlieb, and Kumar Ahuja 2019). The problem of verification is exacerbated by modern machine learning toolchains and packages, which are technically easy to use but build only black box models, creating solutions that are hard to interpret, validate, and debug (Ma et al. 2018). However, recent studies (e.g., Sherin, Khan, and Iqbal 2019; J. M. Zhang et al.

[1] *Security* in the software quality model is considered only as cybersecurity here. Security in machine learning is discussed in Section 3.3

2019) show that the field of testing approaches for machine learning is growing rapidly. Many tools for testing machine learning applications already exist, and many of those considered in the studies are generally applicable, i.e., can be used to test black box systems.

Similarly to verification of machine learning systems, also validation of whether the system meets its requirements introduces additional challenges in comparison to traditional software systems. Firstly, often the quality of a machine learning system's predictions cannot be measured directly, and a proxy is used. These proxies might alienate the system functionality from the original requirements. (e.g., Mehrabi et al. 2019; Russell, Dewey, and Tegmark 2015) Secondly, autonomous systems have the ability to find unexpected ways of meeting the requirements – or proxy targets. Care must be taken to specify the targets in a way that prohibits the system from learning behaviour with unwanted consequences, especially as the capacity of learning systems increases and they are applied to safety-critical tasks. (Russell, Dewey, and Tegmark 2015)

Maintainability

Sculley et al. (2015) and Breck, Cai, et al. (2016) have written few of the most pragmatic papers regarding the technical quality of machine learning systems, in which they describe the characteristics of machine learning that subtly make intelligent systems considerably more complex in comparison to traditional software systems. This accentuated complexity makes maintaining and improving these systems difficult (Sculley et al. 2015).

Perhaps most importantly, the data-intensiveness of machine learning causes strong *entanglement* within the system. This is caused by the fact that each of the input features and hyperparameters in a model, as well as every model in an ensemble, are tied to the values or configurations of each other. Sculley et al. (2015) refer this as to *CACE-principle*, i.e., Changing Anything Changes Everything. Consequently, the abstraction boundaries in the system leak via data features and configuration parameters, resulting in potential impasses for improvement. Unnecessary data dependencies, often present in machine learning systems, needlessly worsen the situation. (Sculley et al. 2015)

Another problem for machine learning development are the general-purpose machine learning packages that often are not directly compatible with each other. Sculley et al. (2015) argue that using these general-purpose packages might hinder the system development and describe a *glue code* design pattern where large amount of data-transforming code is written to achieve interoperability and to integrate new functionality. It should be noted, however, that since publishing their paper in 2015, there have been some efforts to achieve greater interoperability between the general-purpose machine learning frameworks and tools supporting them. The most prominent of these are perhaps Open Neural Network Exchange (ONNX) [2] and Neural Network Exchange Format

[2] https://onnx.ai (ONNX Project Contributors 2019)

(NNEF)[3].

As the machine learning model code accounts for only a fraction of the full system software, disregarding the software architecture and design patterns results unnecessarily in complex, unmaintainable, and unreliable systems (Sculley et al. 2015). However, the software complexity in machine learning systems can be estimated and monitored (Breck, Cai, et al. 2016). The complexity can be reduced by, for example, refactoring code, improving abstractions, and pruning dependencies, and the risk of introducing defects can be reduced by improving tests and documentation (Ma et al. 2018). In addition, Sculley et al. (2015) argue that in a mature system also the machine learning configuration should be treated as rigorously as traditional code. Finally, to avoid increasing technical debt in the machine learning system software, the cost of new functionality or improved model performance should also be measured in terms of system complexity (Sculley et al. 2015).

Performance efficiency and portability

As the field of machine learning is still young, the current frameworks, such as TensorFlow, may not be directly compatible with different platforms (Ma et al. 2018). Porting the software into a compatible form contains a risk of introducing defects in the system, and it might add to complexity via lack of direct compatibility of data formats, as described above. Apart from the interoperability issues, the computational heaviness of machine learning systems might inhibit porting machine learning applications on devices with limited performance capabilities or high energy efficiency requirements (Ma et al. 2018). Susceptible examples of these are wearable electronics and intelligent medical implants (Sze et al. 2017). Compressing the application or data to overcome these constraints introduces a risk of incurring defects into the system (Ma et al. 2018) or decreasing the correctness of the machine learning model (Sze et al. 2017). In addition, small pieces of hardware also increase the risk of hardware-based faults, such as *soft errors* (e.g., flipping bits) (Borkar 2005). If such risk is acknowledged, countermeasures would need to be implemented in the software. (Hanif et al. 2018)

3.2 Data Quality

In data-intensive systems, errors are often caused by problems in the data instead of defects in the software (Batini and Scannapieca 2006, pp. 1–2). In academia, data quality is not given much consideration, and high-quality data is generally taken as granted (Ma et al. 2018). Outside research, however, the poor quality of data is one of the most prevalent issues in machine learning (Sculley et al. 2015) – for example, the inputs consumed by a machine learning model can be expected to contain noise (Ma et al. 2018), and the data relevant for the system is rarely stable over time (Sculley et al. 2015). Indeed, Ehrlinger, Rusz, and Wöß (2019) note that data-driven decisions can hardly be trusted, if there is

[3]https://www.khronos.org/nnef/ (The Khronos Group 2019)

no knowledge of the quality of data.

Unfortunately, there is often no consensus on the definition of “data quality”, as its meaning depends on the context (Ehrlinger, Rusz, and Wöß 2019). Data quality is commonly referred as to *fitness for use* (e.g., Batini and Scannapieca 2006, pp. 221–222, Ehrlinger, Rusz, and Wöß 2019), which makes automated quality evaluation of data difficult. In fact, the survey conducted by Ehrlinger, Rusz, and Wöß (2019) discovered that approximately half of the tools for data quality management, present in literature, were not applicable for general use. In order to consider the quality of data comprehensively, the following describes how data quality can be assessed using quality dimensions. In addition, a few common dimensions in machine learning data quality are discussed; notably also the ethical bias that may be present in data.

Data quality dimensions

As data quality can be evaluated from multiple perspectives, it is often approached with different *dimensions* (Batini and Scannapieca 2006, p. 19). According to Ehrlinger, Rusz, and Wöß (2019), the four most commonly used dimensions are *accuracy*, *completeness*, *consistency*, and *timeliness*. These dimensions are described briefly in Table 3.1.

Table 3.1. *Common data quality dimensions*

Accuracy	Accuracy refers to the syntactic and semantic correctness of the data (Batini and Scannapieca 2006, pp. 20–21), although this definition is not used unanimously (Ehrlinger, Rusz, and Wöß 2019).
Completeness	Completeness of data can be desribed as the “breadth, depth, and scope for the task at hand” (R. Y. Wang and Strong 1996), meaning the extensiveness of different data features, the level availability of these features for each data point, and the representativeness of the data regarding the reference set, respectively (Batini and Scannapieca 2006, pp. 23–24).
Consistency	Consistency of data requires the data to conform to certain predefined semantic rules, for example the integrity constraints in a database schema (Batini and Scannapieca 2006, pp. 30–32).
Timeliness	Timeliness (also currecy and volatility) of data capture the fact that the quality of data changes with respect to time regardless of whether the data itself has changed or not (Batini and Scannapieca 2006, pp. 28–29).

Many other data quality dimensions, such as accessibility, relevancy, and credibility, can be described (e.g., Batini and Scannapieca 2006, p. 38), and also the *ISO/IEC 25012 Data quality model* defines dimensions or characteristics for data (ISO/IEC 25012 2008). In the context of machine learning and other data intensive systems, more focused data quality dimensions can be drawn – for example, Gudivada, Apon, and Ding (2017) list more than ten additional dimensions specifically for machine learning data quality, includ-

ing outliers, data confidentiality, and access controls[4].

The data quality dimensions facilitate data quality assessment, but themselves do not provide metrics for its quantitative measuring (Batini and Scannapieca 2006, p. 19, pp. 48–49). The difficulty of defining data quality, however, should not result in the absence of data quality measurement. While data quality cannot be universally assessed, objective data quality evaluation is feasible *within the application domain* (Batini and Scannapieca 2006, pp. 221–222). In addition, the field of data quality measurement is growing (Ehrlinger, Rusz, and Wöß 2019), making data quality assessment more accessible.

Skew

It is often said that the machine learning model will only be as good as the data with which it is trained (e.g., Ma et al. 2018; Sculley et al. 2015). However, *skewed* data is a problem in many real-world machine learning systems. In the field of statistics, skew generally denotes the asymmetry of distribution of data. In machine learning this often refers to class imbalance or *distribution skew*, where the distributions of target classes are disproportionate (C. Zhang et al. 2019), or *training-serving skew*, where the distribution of feature values differ between training and serving data (Breck, Polyzotis, et al. 2019). Also other types of training-serving skews exist: in *schema skew* the training and serving time data schemas do not correspond, in *feature skew* the training and serving time features undergo different processing, and in *scoring/serving skew* an online learner does not generate new training data for all predictions (Breck, Polyzotis, et al. 2019; *TensorFlow Data Validation: Checking and analyzing your data* 2019).

Stability

A machine learning system interacts with its operating environment via its inputs and outputs. Slow evolution of the environment causes the system performance to degrade over time, for example because of *concept drifts* that emerge in the data (Widmer 1996). Moreover, the decision thresholds in machine learning systems are often fixed, and consequently, the performance of the system might degrade more than what the decrease in the true model correctness is accountable for. Unstable data dependencies also reduce the stability of the machine learning system, as sudden changes in the environment may cause the machine learning system to behave unexpectedly and irrationally. (Sculley et al. 2015)

The instability of a machine learning system can be caused by feedback loops, where the machine learning system directly or indirectly influences its own training data. As machine learning systems do not operate in isolation, live systems almost inevitably create feedback loops gradually over time. Feedback loops, especially when unrecognised, are dangerous because they can slowly shift the system behaviour into unpredictable and

[4] Some of these proposed, machine learning specific dimensions, such as data privacy, are considered within the other characteristics described in this chapter and are not discussed here.

undesirable directions. (Sculley et al. 2015)

3.2.1 Ethical bias and fairness

Ethical bias refers to such properties or *disparities* in data that arise from the types of demographic characteristics that are unjustified to use for differentiation (Barocas, Hardt, and Narayanan 2019, ch. 1). For example, the Equality Act 2010 of the United Kingdom defines a list of *protected* demographic characteristics, including, i.a., race, religion, marital status, and age (The National Archives 2020). The expression *fairness* is established in academia for denoting the research related to detecting and countering ethical bias in the context of machine learning, and has been gaining more awareness quickly in recent years (e.g., Mehrabi et al. 2019).

The most famous example of a biased data-driven system is perhaps COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) – the software that was used to predict recidivism in the United States, but turned out to be biased against non-white people (Angwin et al. 2016). The trail of unfair machine learning is, however, wide. Studies have shown that stereotypes have affected the labelling of a popular image dataset (Miltenburg 2016), and screening algorithms for child maltreatment are susceptible for disadvantaging the poor (Chouldechova et al. 2018). Moreover, a commercial gender classification system has been shown to perform worse on people with darker skin colour (Buolamwini and Gebru 2018).

Most often, bias in machine learning systems is learned from biased data (Papernot et al. 2016), sometimes referred as to “garbage in, garbage out syndrome” (DeBrusk 2018). Also missing data can introduce or amplify biases (Martínez-Plumed et al. 2019). It is well acknowledged that social data sets reflect the historical biases of the society (e.g., Papernot et al. 2016), but bias enters the system in complex ways – for example, the lack of diversity in data causes bias against the underrepresented groups, and naïve aggregations or assumptions on data may result in misinterpretation of data (Mehrabi et al. 2019). Moreover, Barocas, Hardt, and Narayanan (2019) demonstrate that machine learning is rarely not related to people, although the system might not use social data directly. Also the interfaces of systems can introduce bias: for example, the operating system of a mobile device can have a significant impact on how emojis are interpreted by the user (Morstatter et al. 2017). The predictions of an algorithm can also lead into biased datasets, such as in the overpolicing case in Oakland (Barocas, Hardt, and Narayanan 2019). The bias introduced by algorithms into an originally unbiased dataset is often called algorithmic bias (e.g., Baeza-Yates 2018; Mehrabi et al. 2019).

Depending on the data, the machine learning system might become a discriminator in different ways. In the most intuitive case, the algorithm directly uses a protected characteristic as a feature. Often, however, these demographics are not present as features in the data but have correlations to other features instead. Consequently, the machine learning model might learn the protected demographics even if the corresponding fea-

tures are absent from the data. For example, postcodes are likely to be correlated with the race of the residents in the neighbourhoods, and using the postcodes as features for the model might then cause discrimination against certain races. (Mehrabi et al. 2019)

There exist numerous metrics for defining and measuring fairness in machine learning (Verma and Rubin 2018). These definitions are usually classified into either of the following two categories: *individual fairness* and *group fairness* (Mehrabi et al. 2019). Whereas individual fairness cares about the equity between individuals – "similar individuals are treated similarly" (Dwork et al. 2011) –, group fairness is evaluated on an aggregate level only (Mehrabi et al. 2019). Moreover, within these categories, different fairness goals can be optimised: for example, certain metrics optimise the equality of treatment, whereas others care for the equality of the opportunity of the potential outcomes. The difference between these definitions might not be intuitive, and moreover, it is often not clear which metrics are the most suitable for which situations. (Verma and Rubin 2018)

Enhancing fairness of a machine learning system is often referred as to bias *mitigation* (e.g., Barocas, Hardt, and Narayanan 2019; Mehrabi et al. 2019). Three categories of mitigation strategies are generally recognised (Friedler et al. 2019): *Preprocessing* methods change the data so that the correlation with the label stays but is the same for different subpopulations. *Algorithm modifications*, or *in-processing* methods use constraints to optimise the model for fairness in addition to the business target. *Post-processing* approaches change the predictions, e.g, by using different thresholds for different groups. (Friedler et al. 2019) Depending on the application, there might be approaches also partly or completely outside of the model or system – for example, collecting more data is often the foremost approach among data scientists (Holstein et al. 2019).

3.3 Security

Security in machine learning is a rapidly growing field of research: machine learning models have been found vulnerable to, i.a., manipulation of behaviour, deliberate misclassification, and revealing identities. The large input space makes machine learning models inherently vulnerable to security risks (Papernot et al. 2016), as data offers a Trojan horse for adversaries trying to attack the system (Ma et al. 2018). Still, machine learning security risks are not well known or understood to date, and there are no generally effective defences for machine learning specific attacks (Carlini, Athalye, et al. 2019; Koh, Steinhardt, and Liang 2018; Schott et al. 2018). The problem is exacerbated by the lack of understanding of how modern machine learning models, deep neural networks in particular, actually work (Ma et al. 2018).

The following sections give an overview on how machine learning security risks can be assessed using a *threat model* and look into the most common attacks towards machine learning systems: *poisoning attacks*, *evasion attacks*, and *model and training data ex-*

traction. Privacy threats are discussed separately after these.

Machine learning threat model

In their paper, Papernot et al. (2016) use a threat model to disentangle the machine learning security vulnerabilities thoroughly. A *threat model* is a common tool for evaluating the attack surface of an information system and estimating the capabilities and objectives of the possible attackers (Shostack 2014, pp. xxii–xxiii). Papernot et al. (2016) define the attack surface with respect to the different machine learning system pipeline steps and attacker capabilities, and take also into consideration the attacker's visibility — white or black box — into the system. Finally, they use the CIA (confidentiality, integrity, and availability) model to estimate the goals that the attacker might have.

The pipeline-based approach for machine learning system security inspects the security threats with respect to the different stages in the machine learning system pipeline: data collection, feature extraction, model building, and model serving (Papernot et al. 2016). Attacks can happen almost at any step of this pipeline (Ma et al. 2018), but a simpler approach considers the machine learning system pipeline in two major phases: training and inference (Papernot et al. 2016). Attacks at training time make the system vulnerable to manipulated behaviour; most often adversaries attempting to alter the model via tampered data; whereas attacks during inference usually aim at fooling the model with crafted input data or at extracting training data or model information (Papernot et al. 2016).

Another approach on security is whether the attacker has a white-box or black-box visibility into the system. Generally, training-time attacks operate with a white-box visibility and inference-time attacks with black-box visibility. However, during inference, the attacker may gain white-box visibility into the system by stealing the model architecture and training data. (Papernot et al. 2016)

A common approach for assessing information system security is the confidentiality, integrity, and availability model, or *CIA triad* (Andress 2014, pp. 5–7). The approach seems also suitable for assessing machine learning system security, as it takes the data and information content into consideration. In the CIA model, confidentiality specifies that data is available only for those that are authorised for it.[5] Integrity implies that data cannot be unauthorisedly modified and that the undesirable modifications of data, whether done maliciously or not, can be reversed, restoring the integrity of data. Availability means that data is accessible when needed by those who are authorised for it. (Andress 2014, pp. 5–7) Within the context of machine learning, confidentiality attacks attempt to steal the model or the data that was used to train it, integrity attacks aim for changing the model behaviour, and availability attacks generally focus on preventing the system from providing consistent or meaningful outputs. In addition, where a confidentiality attack targets personal data, it is also a privacy breach. (Papernot et al. 2016) These attacks are described in the following.

[5]Confidentiality is also related to privacy, of which definition is however more ambiguous (Andress 2014, p. 96) and which will be discussed in Section 3.3.1.

Poisoning attacks

Poisoning attacks attempt to compromise a machine learning system's integrity. These attacks target the system during training-phase: the adversary has a white-box (or grey-box) visibility into the system and is able to modify the training data set in order to alter the model's decision boundaries. (Papernot et al. 2016) Moreover, Kearns and M. Li (1993) demonstrate that poisoning attacks are relatively powerful – intuitively, the accuracy of the model cannot be higher than the accuracy of the data used to train it. While poisoning attacks are applicable to any machine learning model, the main focus of the current research is on attacks towards supervised classifiers. (Papernot et al. 2016)

Training data can be poisoned by modifying the target labels. In the most simple scenario, the adversary is capable of randomly perturbing the labels of a subset of the training data. A more sophisticated adversary focuses the manipulations to the labels of those samples that are classified with high confidence. However, the computational heaviness of searching these *poisoning points* often makes attacking them infeasible. (Papernot et al. 2016)

Another way of poisoning the machine learning model's training data is to manipulate its features, and consequently shift the model's decision boundaries. This can be done, for example, by injecting malicious samples into the training data set. Adversaries able to interfere with the data only before its pre-processing are much weaker in comparison to those with access to the pre-processed training data. (Papernot et al. 2016) Poisoning the training data features appear an alarmingly prospective scenario, especially in dynamic online learning systems – a famous real-world example of this was the Twitter bot, published by Microsoft in 2016, that quickly learned racist behaviour from internet trolls (Victor 2016).

To date, anomaly detection and malicious sample rejection has been shown to be an effective technique against optimal poisoning attacks (Collinge, Lupu, and Muñoz-González 2019; Paudice et al. 2018; Y. Wang, Jha, and Chaudhuri 2019). For example, certain complex settings can utilise the temporal coherence in behaviour to detect an attack (Lin et al. 2017). However, poisoning attacks are far less well studied in comparison to evasion attacks, and stronger attacks that evade the developed detection mechanisms are likely to be found in future (Koh, Steinhardt, and Liang 2018).

Evasion attacks

Evasion attacks, or *misprediction* attacks, target the integrity of a machine learning model during inference. Evasion attacks in machine learning utilise perturbed inputs, *adversarial examples*, that fool the model to mispredict these inputs. Because the attacks occur during inference, the adversary's capabilities increase along with the knowledge of the model internals: if the adversary has enough information of the model architecture, also the target label of the input to misclassify can be determined. (Papernot et al. 2016)

(a) A stop sign classified as a speed limit sign (adapted from Eykholt et al. 2018).

(b) The target 80 mph speed limit sign (Federal Highway Administration 2004).

Figure 3.1. *A perturbed stop sign (a) has been shown to be classified as an 80 mph speed limit sign (b) with 80 % success rate in real-world settings (Eykholt et al. 2018).*

Evasion attacks have gained a lot of attention in the academia – potentially because the amount of perturbation needed to cause misclassification is small and in practical applications often inconspicuous for human senses and imperceptible at worst (Szegedy et al. 2013; Wiyatno et al. 2019). While initially, the research on the topic used to feed the adversarial samples directly to the models, evasion attacks are possible even when the adversarial input has to preserve its properties in the data processing pipeline. A practical example of this is systems in the physical world: for example, road-sign classifiers have been proven to be fooled by a stop sign perturbed with black and white stickers (Eykholt et al. 2018) (see Figure 3.1). Another alarming property of adversarial examples is their *transferability*: an adversarial input, able to evade one model, is very likely to be able to evade also another model trained for a similar purpose (Szegedy et al. 2013).

The academia has developed a number of different types of attacks, i.e., methods that produce adversarial examples, as well as defences for them, but no generally robust protection mechanism exists (Wiyatno et al. 2019). The most intuitive defence is perhaps *adversarial training* (e.g., Goodfellow, Shlens, and Szegedy 2014; Wiyatno et al. 2019) that increases the robustness of a machine learning model by augmenting the training data set with adversarial examples. For example, a machine learning system can be periodically retrained with the misclassified samples used by attackers, if the necessary information is available or obtainable. Adversarial training has also been shown to be a relatively effective defence, although a systematic retraining with *searched* optimal adversarial examples can increase the impact of the method notably (Chen, B. Li, and Vorobeychik 2016; B. Li, Vorobeychik, and Chen 2016). However, the achieved robustness will not generalise on all types of adversarial examples (Carlini, Athalye, et al. 2019).

Evasion attacks in intelligent systems are not new – misprediction was originally studied in the context of spam filters by Dalvi et al. already in 2004. However, the discoveries made by Szegedy et al. (2013) have sparked the research field again. Nevertheless, in the context of neural networks, there is still no knowledge of *why exactly* adversarial ex-

amples exist. The hypotheses vary from the original speculation of non-linearity (Szegedy et al. 2013) to a speculation of local linearity (Carlini and D. Wagner 2017) and a more recent theory of them being caused by predictive but non-robust features of the training data (Ilyas et al. 2019).

Model and training data extraction

Model and training data extraction can be classified as attacks on confidentiality. Here, the adversary has only black-box visibility into the machine learning model and tries to reverse-engineer either the model or the training data – or both. The goal of the adversary might be, for example, to gain white-box visibility to the model and further to perform other types of attacks against the system. (Papernot et al. 2016) When the training data contains personal information, the attack also targets privacy (see Section 3.3.1).

Alarmingly, the extraction attacks do not require a special access to an unprotected model, in order to succeed. Tramèr et al. (2016) have shown the efficiency of model extraction attacks against ML-as-a-Service (MLaaS) cloud platforms by extracting trained models using the API provided by Amazon Machine Learning. Fredrikson et al. (2014), on the other hand, have demonstrated the ability to reverse engineer the model's training data from a pharmacogenetical system used to recommend personalised dosages of medicine to patients. The authors of these papers also discuss the defence mechanisms against these types of attacks, but similarly to the other security attacks, discussed above, no generally applicable, strong protection mechanisms exist.

3.3.1 Privacy

Apart from general threats to security, also privacy can be compromised in machine learning systems. However, *privacy* is an ambiguous and multifaceted concept. Andress (2014, pp. 95–99) notes that the dictionary definition for privacy – the "state in which one is not observed or disturbed by other people" (Oxford University Press 2019) – is lacking in ways, and generally there exists no unanimous definition for privacy. As noted above, model training data extraction attacks also privacy, if personal data is being used by the model. In legislation, privacy usually addresses the rights of individuals when their personal information is processed and stored by someone else (Andress 2014, pp. 95–99). In addition, data privacy regulations, such as GDPR (General Data Protection Regulation) (Publications Office of the European Union 2016) and CCPA (The California Consumer Privacy Act) (California Legislative Information 2020), attempt to defend the ethical rights of individuals to their data. In the following, privacy is discussed in two, somewhat separate contexts: privacy as a security concern and privacy of an individual's personal data.

Attacks on privacy

As machine learning models have a vast threat surface and the applications need large amounts of data, machine learning systems are both attractive and vulnerable target for adversaries attacking on privacy. Using machine learning on personal data always presents a privacy risk: as machine learning models have the capacity of "remembering" details in their training data (L. Yu et al. 2019), individuals become susceptible to privacy attacks by the same mechanisms that can be used to extract any training data from the model. Moreover, the risk stretches also to those whose data is not used for training, as details can be inferred from the model using some *auxiliary information* outside of the system (Dwork 2006).

Privacy attacks on machine learning systems aim at extracting personal information from the machine learning model. The attacks are generally targeted at the system during inference while the adversary has a black-box visibility to the system. The most common types of attacks are membership tests and (partial) recovery of training data: a membership test is conducted to determine whether or not the data of a certain individual was used in the training, whereas a stronger adversary may attempt training data extraction to recover unknown or partially known training data points (Papernot et al. 2016). Anonymisation as a defence is ineffective against these types of privacy attacks (Taeihagh and Lim 2018), and even aggregation is not necessarily an efficient way of protecting privacy. It can be mathematically proven that, in the presence of some auxiliary information, absolute privacy cannot be achieved. This auxiliary information could be any prior knowledge of an individual, and is practically present in any real-world case. (Dwork 2006)

Privacy-preserving machine learning attempts to minimise the risk that a machine learning model or its development could reveal personal information or allow identifying an individual (Al-Rubaie and Chang 2018). One of the most popular privacy defending mechanisms is differential privacy, originally formulated by Dwork (2006). *Differential privacy* is a mathematical framework for privacy, providing guarantees that nothing additional can be learned from individual who is participating in a data set in comparison to someone that is not. Further, it is not possible to ascertain from differentially private algorithms, whether or not data from an individual was used in the computation of the results. The mathematical framework can be applied in machine learning to achieve differentially private models. (Ji, Lipton, and Elkan 2014; L. Yu et al. 2019) Differential privacy works by adding certain amount of random noise to the data (either directly or during aggregation) (Dwork 2006). As such, it is bound to decrease the accuracy of differentially private machine learning models (L. Yu et al. 2019), and is not an applicable solution in all domains (e.g., Fredrikson et al. 2014).

Data privacy

In addition to the different forms of privacy attacks, machine learning systems utilising personal data present also other types threats for an individual. For example, the individual's

personal data may be sold to third parties, such as advertisers or insurance companies, that might utilise the information unethically (Taeihagh and Lim 2018). In this work, *data privacy* addresses the collection, handling and using of personal and sensitive data from ethical and regulatory perspectives. *Privacy* and *data privacy* are not completely separate concepts, but privacy in the context of security does not address all issues, such as ethical questions, raised by data privacy. Similarly, data privacy is insufficient for ensuring privacy as a security concern. Therefore, these two topics are discussed independently.

The European Union's (EU) General Data Protection Regulation, commonly known as GDPR, was established for protection of the data of individuals. The regulation applies to organisations that process personal data and specifies, for example, that an individual has the right to access their personal data that is used by an organisation and to know the purpose of using their personal data, as well as the right to forbid processing their personal data and to request the removal of their personal data. However, GDPR applies only to EU citizens and to organisations that operate within the EU. (Publications Office of the European Union 2016) The California Consumer Privacy Act, CCPA, poses similar obligations to companies that collect data from consumers in California (California Legislative Information 2020).

Federated learning has been proposed as a solution for privacy-preserving machine learning. It is a form of distributed or *edge* computing, where the training of machine learning algorithms occur on remote nodes, such as end user mobile devices or hospitals. Instead of the personal data of individuals, only secondary information, such as the model updates, are transferred back to the service provider. However, federated learning does not guarantee to *secure* privacy of individuals as such, but require further privacy measures, such as differential privacy, in order to be effective against privacy attacks. In addition, there are numerous open issues regarding the technical implementation of such decentralised system, but research on the topic is increasing. (T. Li et al. 2019)

3.4 Explainability

As machine learning models learn their decision logic from data without explicitly programming, their reasoning is opaque by nature, making these systems hard to trust and difficult to develop. Even for experts, machine learning systems can be "easier to experiment with than to understand" (Golovin et al. 2017), and the problem is exacerbated by the ever-increasing complexity of the systems (Ma et al. 2018). The field of explainability (XAI) aims at improving the transparency and trustworthiness of machine learning systems by making them more *interpretable*, i.e., allowing understanding *why* the machine learning model made the predictions it did (e.g., Carvalho, Pereira, and Cardoso 2019; Doshi-Velez and Kim 2017).

The need for explainability

Explainability, often used interchangeably with *interpretability*, differs from the previously described three machine learning system quality characteristics in that an uninterpretable, black-box system is not broken and does not present an immediate risk to its users. However, interpretability is a prerequisite for confirming the reasoning behind a machine learning model's decisions, which further enables improving the system's reliability, fairness, and trustworthiness (Carvalho, Pereira, and Cardoso 2019; Doshi-Velez and Kim 2017). For example, interpretability is said to help in detecting and reducing ethical bias (e.g., Doshi-Velez and Kim 2017; Sokol and Flach 2020). Moreover, explainability can help in ensuring the system's safety, because it allows understanding the causal relations between the machine learning model's inputs and its predictions (Doshi-Velez and Kim 2017), supporting testing, auditing, and debugging (Carvalho, Pereira, and Cardoso 2019). Consequently, by improving the safety of machine learning systems, explainability can also be thought to increase the social acceptance for intelligent systems.

The ability to explain a machine learning system's decisions is fundamental when its predictions have direct consequences for people. As accountability and liability cannot be determined without certain amount of transparency into the system's logic, individuals cannot argue against the decisions made by extrajudicial, opaque machines (Carvalho, Pereira, and Cardoso 2019). Lack of explainability could therefore prohibit adopting new technology, especially in high-risk and safety-critical domains that are heavily regulated. Generally, for these reasons, the ability to interpret machine learning predictions is necessary whenever the public sector is involved. (Carvalho, Pereira, and Cardoso 2019; Taeihagh and Lim 2018)

Although many benefits of reducing opaqueness can be enumerated, explainability is not indispensable for all machine learning systems. Doshi-Velez and Kim (2017) argue that this might be the case either when there are no severe consequences for mispredictions or when the problem has been studied and validated in real-world use thoroughly enough in order to trust the system. An example of the former are recommendation and advertisement systems, where the consequences of incorrect predictions are usually not critical (Carvalho, Pereira, and Cardoso 2019; Doshi-Velez and Kim 2017) and accuracy could be a more important quality attribute to enhance. In these settings, the cost of increasing the system explainability can be weighed against the usefulness of it.

From explainability to explanations

Explainability methods can be categorised into two groups: models that are directly interpretable, and explanation methods that can be applied to black-box models. Interpretable models, such as linear regression and decision trees, can be understood to some extent directly from their coefficients, decision boundaries, or other meaningful parameters, and are the easier approach for building an explainable system. The benefit of the second cat-

(a) An image to classify with Google's Inception network.

(b) The explanation provided for the classification.

Figure 3.2. *An explanation created with the LIME algorithm (b) for an image classified as "electric guitar" (a) (adapted from Ribeiro, Singh, and Guestrin 2016).*

egory, explanation methods, is that they can be model-agnostic, making them applicable in many systems. (Carvalho, Pereira, and Cardoso 2019; Molnar 2019)

Explanation methods do not attempt to make the machine learning model directly interpretable, but use *explanations* as means instead. Explanations can help in understanding, for example, single predictions, group of predictions, or global trends. Concrete examples of explanations are feature summaries, simple descriptions of the model internals, or representative data points. (Carvalho, Pereira, and Cardoso 2019; Molnar 2019) Also more sophisticated frameworks for interpreting machine learning predictions have been proposed – for example, the LIME (Local Interpretable Model-agnostic Explanations) algorithm can be used to retrieve the features in the inputs based on which the model makes its predictions (Ribeiro, Singh, and Guestrin 2016), whereas SHAP (SHapley Additive exPlanations) combines multiple explainability approaches to achieve explanations more faithful to human intuition (Lundberg and S.-I. Lee 2017). Figure 3.2 illustrates an explanation created with LIME.

Assessing the quality of explanations is difficult. There cannot be a formal definition for interpretability, as the concept is subjective and explanations are tied to the context in which machine learning is used. (Carvalho, Pereira, and Cardoso 2019; Doshi-Velez and Kim 2017) Simply *correct* explanations are rarely the *best* explanations (Molnar 2019) — ideally, explanations should be as accurate, understandable, and efficient as possible, but these goals often conflict with each other. For example, a very truthful explanation can be incomprehensible to a human, making the quality of the overall system explainability hard to assess. As the field of interpretability is young in comparison to other tracks in machine learning research, finding reliable, automated ways to assess explainability remains largely a research question. (Carvalho, Pereira, and Cardoso 2019)

3.5 Summary

As mentioned in the introduction of this chapter, it is necessary to define the different quality characteristics that concern machine learning systems in order to consider quality assurance in machine learning work. As no prior definition for machine learning quality exists in literature, the quality perspectives have been drawn from the prominent research tracks related to machine learning. The found perspectives can be classified using four main categories: technical quality, data quality and ethical bias, security and privacy, and explainability. These categories are briefly described in Table 3.2.

Table 3.2. *Summary of perspectives into machine learning quality*

Technical Quality	Technical quality views the machine learning system as any other software system, and a software quality model can be used to assess the system's quality. The most studied characteristics in technical quality of machine learning systems are correctness, maintainability, performance efficiency, and portability.
Data Quality and Ethical Bias	Data quality is integral to machine learning systems, as the system logic is largely determined by data. The most common problems associated with data in this context are the different forms of skewness and data instability over time. In addition, a notable issue in machine learning data quality is ethical bias, present in practically all data produced by humans, resulting in unfair systems. Data quality dimensions can be used to assess the quality of data from different perspectives.
Security and Privacy	Machine learning systems present new types of security risks, as the system's confidentiality, integrity, and availability can be compromised in many ways. The most common security attack types are poisoning attacks, evasion attacks, and model and training data extraction or reverse-engineering. In the context of privacy, the last group of attacks, training data extraction, have special names membership test and partial recovery. In addition to privacy attacks, also data privacy, i.e., individuals' rights to their data, is gaining traction.
Explainability	Explainability, or interpretability, attempts to increase the understandability of modern, opaque machine learning systems. Explainability allows assessing other machine learning system desiderata, such as trustworthiness and safety, making it essential for high-risk and safety-critical systems.

There is also a vast, growing literature around machine learning and ethics. The focus of the research is on building "moral agents", i.e., how the universal ethical and moral considerations could be built into artificial general intelligence. (e.g., Cervantes et al.

2020; Tolmeijer et al. 2020; H. Yu et al. 2018) While the topic is arguably important regarding independent, artificial agents, it is not likely that such ethical problems are encountered in today's machine learning systems. As such, the theme goes beyond the scope of this work and is not discussed further here.

In Chapter 2, the existing software quality models were claimed inadequate for capturing the specificities of machine learning systems. This claim is not completely true, as all of the quality perspectives discussed in this chapter can be mapped to these models. For example, as machine learning systems learn their logic from data, the quality of data can be thought to support mainly the functional suitability in the ISO 25010 Software product quality model, although arguably the quality of data has an effect also on security and reliability. Similarly, explainability appears to contribute to both usability and reliability in the model. However, the inadequacy of software quality models for machine learning does not arise from the completeness of these models, but rather the emphasis and ambiguousness of the different characteristics in them in the context of machine learning. As Nickerson, Varshney, and Muntermann (2013) have pointed out, a useful model is *concise* and *explanatory*, and the existing software quality models do not fulfil these requirements for machine learning systems.

4 METHODOLOGY

This chapter presents the rationale for selecting a methodology for the study and describes the design and the course of collecting of the empirical evidence. A closer look into the reliability of the used methodology is taken in Chapter 6.

4.1 Conducting qualitative research

Generally, qualitative research aims at understanding and explaining phenomena by using qualitative analysis methods (e.g., Packer 2010, pp. 1–7; Eskola and Suoranta 1998, ch. 1) and allows creating new theory from empirical material without prior constructions (Eskola and Suoranta 1998, ch. 1). The setting of this study was qualitative in nature: to answer the research questions, qualitative material of working with production machine learning systems was needed. Interviews were selected as the method for data collection: interviews are the most used method for collecting empirical material in qualitative social research and they can be used to study a variety of phenomena (Packer 2010, pp. 42–43). It was thought that by interviewing professionals that have been working with real-world machine learning systems, it would be possible to gather the richest data that would answer the research questions. Moreover, the possibility to have an interaction with the informants was seen necessary, because as discussed in the previous chapter, there is a lack of general clarity around the topic. A dialogue was thought to help the informants in providing a higher variety of experiences.

Alternatives

Another common method for collecting subjective information qualitatively are questionnaires (e.g., Hirsjärvi and Hurme 2008, pp. 34–37). However, questionnaires are less adaptive than interviews (Hirsjärvi and Hurme 2008, p. 36). Moreover, semi-structured and unstructured interviews in particular give the examinee considerably more freedom in terms of how the topic is approached and to what extent it is discussed (Packer 2010, p. 43). The hypothesis in this study was that quality in machine learning systems has not been systematically considered as such, but the work practices have evolved to support and assure quality in different ways. Questionnaires were determined deficient for discovering this implicit knowledge, because it would have been difficult to formulate the questions in such a way that would not limit the narrative of the informants, especially in cases where quality characteristics that were not discussed in Chapter 3 would appear.

Also a case study can be used to collect material in a qualitative research. Case studies often employ interviews as method for collecting qualitative material, but also other types of evidence can be used. (e.g., Eriksson and Kovalainen 2008, ch. 9; Gillham 2000, pp. 9–14, 20–21) Consequently, an *extensive case study* (Eriksson and Kovalainen 2008, ch. 9) might have given more in-depth understanding in the research questions in comparison to mere interviews. However, it was considered too resource-demanding with respect to the potentially added value. Moreover, in industry, the non-disclosure agreements (NDAs) of the respondents might have prohibited thorough examination of the cases.

Interview process

For the interviews, it was necessary to search machine learning developers and other experts that had experience from production machine learning systems in industry. The experience length in years was not regarded as a meaningful criterion because of the novelty and fast evolving pace of the field. Also the role of the person was not considered important as long as the experience profile was suitable. While many of the data scientists working in the industry also have a history in academia, professionals with *only* academic experience (i.e., no industry experience at all) were not considered, because it was assumed that the quality issues in productionised real-world systems are more versatile in comparison to research projects and proof-of-concept applications (Breck, Cai, et al. 2016).

The interviews were decided to be organised either individually or in small groups (three interviewees at maximum). Group interviews were used mainly for convenience. It was acknowledged that group interviews may introduce bias by imbalanced group dynamics, dominance effect, and groupthink (e.g., O.Nyumba et al. 2018; Hirsjärvi and Hurme 2008, p. 63). However, the potential of bias was considered insignificant, because the interview topic was not sensitive and the selected interviewees would be experienced in the field. The possible dominance effect was prepared to address during the interviews (e.g., O.Nyumba et al. 2018). In addition, by limiting the group size, the effect on the recording quality (Hirsjärvi and Hurme 2008, p. 63) was avoided.

The interviews were constructed as semi-structured (theme) interviews. Semi-structured interviews were determined as a suitable approach because of the simultaneous ambiguity and clear scope of the research. This interview type also allows collecting very profound information (Packer 2010, pp. 2–3). Structured interviews were thought to have the same problems as questionnaires. On the other hand, in comparison to unstructured interviews, themes were thought to frame the topic more clearly for the interviewees (Hirsjärvi and Hurme 2008, pp. 66–67): It was assumed, that the interviewees would generally consider the machine learning model correctness and robustness as machine learning quality. Similarly, testing is often used as a synonym for quality assurance. The selected themes were supposed to guide the interviewees to think more comprehensively of the topic.

The themes for the interviews were constructed as a matrix. The first axis was formed by the four main quality aspects found in literature, summarised in Section 3.5. These quality aspects were also described for the interviewees at the level of detail of the summary (see Table 3.2). The second axis was formed by a very high-level perspective to quality assurance. This perspective contained actions taken before building a system, such as design decisions; actions taken while building the system, such as testing; and actions taken to control failures or disasters. Notably, the quality assurance theory, used in the interviews, was much less detailed than what is described in Chapter 2, and the purpose of the quality assurance axis was mainly to demonstrate how broadly quality assurance can be considered. The interview theme matrix is illustrated in Table 4.1.

Table 4.1. *Interview theme matrix*

	Constructive QA	Analytical QA	Failure control
Technical quality			
Data quality & Ethical bias			
Security & Privacy			
Explainability			

Before discussing the themes, the interviewees would be asked to tell about their education and work history. Then, the interviewees would be asked whether their work history contained some quality critical work, with the purpose of understanding the interviewees better in terms of how aware of quality they were in general. After that, the interview would proceed to asking the interviewees' definition for quality in machine learning systems to collect genuine experiences and views on the topic. Only then, the interview would move on to the theme matrix. The discussion around the themes would be quite open, and the interviewees would be allowed to describe their experiences around the themes or outside of them, if the topic was related to the purpose of this study. The interview protocol was validated with one test interview before the actual interviews, but as the protocol was not changed as a result and the profile of the test interviewee was suitable for this study, also the test interview was included in this work. The protocol can be found in Appendix A.

4.2 Data

All the interviewees were found in information technology consultancies and product companies within Europe, majority of these being consultancies that operate in Finland also. The interviewees were recruited using snowball sampling (e.g., Hirsjärvi and Hurme 2008, ch. 5). New interviewees were taken until thematic saturation of the responses but minimum of twelve respondents, as proposed by Guest, Bunce, and Johnson (2006). There was also an option to search more interviewees if the analysis would reveal a gap in the findings.

Altogether there were 13 interviewees from 6 different companies. Almost half of the respondents had also a background in research before moving to the work in the industry. The experience profiles of the interviewees are summarised in Table 4.2.

The professional networks in the field of machine learning are relatively small in the industry in Finland, making the respondents vulnerable to deductive disclosure or identifiable within the network (Kaiser 2009). To ensure the confidentiality of the interviewees, this work follows the dominant approach, proposed by Kaiser (2009). Consequently, no further information of the demographics of the interviewees is provided.

Table 4.2. *Profiles of the interviewees*

	Background		
Experience (years)	**Academia**	**Other**	**Total**
Less than 10	1	3	**4**
Between 10 – 15	4	2	**6**
Longer than 15	1	2	**3**
Total	**6**	**7**	**13**

All of the interviews were conducted between October, 2019 and February, 2020, and the interview language was either Finnish or English. The interviews were reserved approximately one hour of time, except for the group interviews that were reserved two hours.

At request, the interviewees were provided with the themes before the interviews. However, this did not affect the planned course of the interviews. To prevent biased results, these interviewees were asked to think of their own definitions for quality in general and quality in machine learning systems before proceeding to the themes.

In the beginning of each interview session, the interviewees were explained the confidentiality of the interviews and their rights with respect to the interviews. The permission for recording the interview was then requested in writing with an interview consent form that also contained the same disclosure (see Appendix B). All of the interviewees permitted recording, and each interview recording was transcribed by myself within two weeks of the interview.

The course of the interviews generally followed the same routine, although additional questions were presented for more details when necessary. As planned, the themes were not introduced for the interviewees directly after the general background questions – instead, the interviewees were asked to define quality in general, as another “background” question, and after that to think of the quality in machine learning systems specifically. However, majority of the interviewees took the context of machine learning systems already when asking about quality in general, and these two questions were therefore merged. The general view on quality was not further queried. The interviewees were allowed to talk about their own experiences of quality in machine learning systems as long as they had something to tell and they stayed within the theme matrix or on a relevant

subject. The quality aspects were then presented, either all at once or one at a time, depending on the state of the discussion, and the interviewees were asked experiences related to them. Examples of these questions are listed in the interview protocol in Appendix A. Also the quality assurance axis was used in most of the interviews, but it was not seen necessary if the interviewee spontaneously discussed experiences along the complete definition.

The quality themes were discussed in the interviews to the extent that the interviewees had experiences of them. Most of the time, the one-hour period for the individually interviewed participants was not quite enough to cover all of the themes from where the participant had experiences, but the sessions were extended up to the necessary time (15 minutes at most). The group interviews took generally a little less than the reserved two-hour period.

The atmosphere in the interviews was good, and there were no surprises that would have affected the interview plan. An interview diary was used to keep a track of details that would affect the reliability of the study.

4.3 Data analysis

Altogether, there were 123 pages of transcribed material. The interviews were first transcribed verbatim (excluding confidential information), but for the analysis these transcriptions were converted into universal language. As the recordings were of good quality, the transcriptions did not cause unreliability into the analysis.

The approach to the analysis was inspired by Grounded Theory, originally described by Glaser and Strauss in 1967. Grounded theory is an analysis methodology that focuses on constructing a new theory from the collected research material. The discovered theory would be *grounded* in data, making grounded theory a more reliable method in comparison to logico-deductive approaches (Glaser and Strauss 1967, pp. 2–6). Grounded theory is also one of the most popular analysis approaches in qualitative research (Packer 2010, p. 60).

According to Glaser and Strauss (1967), *theoretical sampling*, i.e., the joint process of collecting, coding, and analysing the research material, is required if a research aims at developing a new theory (1967, p. 45, 71). In this study, coding and the actual analysis were started only after having the initial set of interviews. However, there was an option to conduct more interviews, if it would seem necessary after the analysis. One additional interviewee was searched after the initial analysis, as there seemed to be too few experiences around one of the quality characteristics.

The analysis was carried iteratively using the *constant comparison* approach, as described by Glaser and Strauss (1967, pp. 105–113). All incidents were coded, even if saturation had clearly been reached. In addition to using code memos, the coded segments were attached comments which were also used for synthetisation. Although the

interview themes expectedly reflected into the material, the coding or results were not constructed around the themes.

5 FINDINGS

This chapter presents the findings from the interview material: Section 5.1 describes the participants' experiences regarding the different quality perspectives into machine learning systems, and Section 5.2 introduces the components of quality assurance that the material suggests having the largest impact in ensuring the quality of those systems. These sections are outlined in Tables 5.1 and 5.2, respectively. Finally, Section 5.3 summarises the findings with respect to the research questions.

Table 5.1. *Outline of the findings on quality characteristics*

Quality characteristics	
Service quality	5.1.1
Technical quality	5.1.2
Data quality	5.1.3
Security & Privacy	5.1.4
Interpretability	5.1.5
Fairness	5.1.6
Ethics	5.1.7

Table 5.2. *Outline of the findings on quality assurance*

Quality assurance activities	
Domain & data understanding	5.2.1
Design	5.2.2
Verification & Validation	5.2.3
Documentation	5.2.4
Engineering practices	5.2.5

5.1 Disambiguating quality

The classification of the quality perspectives, presented in Chapter 3, seemed to be nearly complete in the sense that almost all of the discussed quality problems could be placed under the four categories. Notably, this was true even before the categories had been presented for the participants [1]. In addition, most of the interviewees thought that the perspectives capture the essential quality considerations in machine learning systems. However, the results show that the classification is somewhat misleading in that the literature perspective into the categories did not necessarily reflect well the issues appearing in real-world system development. This section discusses the findings as of the quality problems that emerged in the interview material.

[1] Including those participants that did not ask to see the interview themes in advance.

5.1.1 Service quality

The machine learning *service* constitutes a distinctive part of all machine learning systems. It is therefore understandable that the quality of the machine learning predictions was given much attention in the discussion. Following Nakajima (2018), this work adopts the notion of *service quality* to indicate the quality of the functional and non-functional properties of the system that are inseparable from machine learning. Notably, in comparison to what was presented in Chapter 3, here service quality is considered separately from technical quality, because the emphasis of the described issues deviates from that proposed by the literature.

The interview material highlighted two perspectives into machine learning service quality: the quality of machine learning predictions, and the validity of the machine learning service. These aspects are discussed further in the following.

Prediction quality

In machine learning, *prediction quality* was generally the first concern for the interviewees and clearly the most deliberately considered perspective in service quality. The interviewees frequently returned to the *correctness* of the predictions in their talk, although similarly to the trend in literature, the term *accuracy* was used in place. The interest on correctness was quite expected, as the indeterministic nature of machine learning creates the fundamental difference between classical software systems and those using machine learning:

> *You have the benchmarking, this is an important thing. You need accuracy. In software development you can have that, but you don't tend to think about accuracy . . . , you tend to be saying: if you put inputs, you get certain outputs. Machine learning doesn't work that way because of the variability of the outputs. I mean, you're not going to get 100 % of something, you're going to get 87 % or something like that.*

Another property in prediction quality that appeared in the discussion was robustness, i.e., the ability of a model to learn the essential information from the data and to resist perturbations during inference (e.g., J. M. Zhang et al. 2019). In general, the interviewees discussed robustness indirectly – "robustness", specifically, was rarely mentioned, but its subcomponents, such as stability and generalisability of the predictions, were referred to instead. Overall, it appears that high machine learning robustness was not considered elementary in many types of systems directed for consumers. In fact, activity related to ensuring robustness was discussed only by those that had experiences in safety-critical and high-risk domains, where its importance was demonstrated in the following manner:

> *For example, thinking about self-driving cars – it's really simple to develop a self-driving car that in standard circumstances stays on road and dodges pedestrians. But it's the model's behaviour in all possible exceptional cases*

and situations that will define whether it can be taken into use.

While most of the discussion touching prediction quality was either about improving or ensuring it, it was not always considered as the most important characteristic of the system. Usually, the reason for this was that something else was more meaningful with respect to the business logic – or purpose – of the system. For example, a few participants brought up cases where interpretability could be more important for the users in comparison to increased correctness.

> *We've implemented a few means of communicating to the user why a prediction is important for them. So that the user can understand how the machine works, but it also makes their job easier: [they don't need to go through all the results] – rather, they can see the reasoning behind the predictions, like, "okay, maybe this is a good result". Or the other way around, they are quickly able to see that "this is not a good enough reason". And* $that$ *is essential for the product quality. If we can help the user in processing the results, it's as valuable as improving the quality of the predictions.*

Despite its apparent prominence in the interviews, machine learning prediction quality was usually not the topic of the discussion. Instead, the reason for its prevalence is that often the other quality aspects were discussed *in relation* with it. In fact, it was well demonstrated that the prediction quality of a machine system depends on and is affected by nearly all other parts of the system. For example, low data quality was described as a "*bottleneck*" for many projects, whereas high software complexity was noted to hinder development of the service.

> *So, data quality. I think it's always, often a big bottleneck. The data is just garbage. Especially, when I was doing shorter projects where I was just looking what data the client has, maybe trying to write some model. Usually the project just ends at that point, because there's just no signal.*

> *The fact that we took a proof-of-concept and put it into production and then incrementally improved it ended up causing us all manner of headaches. . . . it was very fast, very performant, pretty good at what it did, but basically the person had written down the math and then translated the math directly into code . . . Then when it came down to like, "okay, we need to consider trying different algorithms or trying to get our classification accuracy up a bit", it was like, "we can't work with this". Then rewriting the whole machine learning component.*

Overall, the perspectives provided here function as an introduction for the rest of the findings presented in this chapter. While it is evidently necessary to focus on prediction quality in machine learning, there are many other aspects in real-world production systems that can be as indispensable to consider. These aspects are described with more details in the following sections.

Service validity

Often, machine learning service quality is considered solely through its measurable attributes, such as correctness and level of overfitting. A perspective of service quality, prevalent in the interviews but quite absent in literature, was the difficulty of soundly translating the desired business logic of a machine learning service into a mathematical form using the available data.

> *The expectation might be that we want to build the product that makes customers happy, but it's difficult to formulate that as a mathematical evaluation meter.*

In this work, the term machine learning service *validity* is used to denote how accurately the machine learning model implements the business logic that is attempted to achieve. Validity is different from correctness in that whereas correctness measures the prediction quality against the available data, validity captures the more abstract idea of a business objective, which is not necessarily ensured by high correctness. In other words, a machine learning model can have very high correctness with the available data, but often it is, in fact, optimising something else than the actual (business) target, creating a discrepancy between the measurable and perceived performances.

Several cases that incurred problems in functional validity were discussed in the interviews. For example, in recommendation systems the choice of target labels and evaluation metrics can be challenging, because there might be no straightforward way of measuring the succeeding in the recommendations. Similarly, in optimisation systems "*it is not always so trivial to ensure the correctness*" of the predictions, making it even more challenging to assess the validity of the system.

> *If I recommend a content and you tend to click, it doesn't mean that it's related to you or relevant to you; doesn't necessarily mean that.*

> *When we're talking about predictions, in principle, it's easy to validate the result; we compare the truth and the prediction, do they match. But when we are talking about optimisation, for example price optimisation – we get a recommendation, "okay, the price is this", but how do we know if it's correct or not? So it's a lot harder to say, if it is the optimal price. If there's some [expert] saying "I think the price is this but the machine says that", how do you know which one of them is correct?*

Problems in machine learning validity stem from the impossibility of directly measuring the succeeding in the business objectives. When proxy targets are used, there is a risk that those proxies do not appropriately reflect the actual target. Also proxy features can lead into similar disparity.

> *One thing that is really coming across in these projects is that often, when . . . we inform decisions based on machine learning or any data-driven approach . . . , we create these proxy target variables that usually disconsider*

the overall performance.

It is generally said that data defines the logic of a machine learning system, but in fact this only happens via the targets that are being optimised. The same data will result in a number of different functional variations depending on the model configuration, most importantly the chosen targets and metrics. Consequently, for a single data set, the differences between resulting models can be notable. One of the interviewees demonstrated this with a case where the productivity of different departments of an organisation had been modelled: although the available data and business objective had remained the same between the models created by different parties, the results of the different models were opposite at worst.

> *For example, ["A"] claimed that [department "X"] was one of the best. Then some critique came up: "actually, your estimations are wrong, because you didn't take into account that part of the services have been relocated . . . , so there are no costs but the revenue is the same". This type of differences, caused by calculation logic and data semantics. So, then they fixed their calculations and got that it's somewhere in the middle. But at the same time, ["B"] had estimated that it's one of the worst. . . . So when we choose to trust [some version] of artificial intelligence, it might be that its predictions are totally wrong, and then we start to converge the whole [organisation] towards the worse department, because we thought it was the best.*

Another manifestation of low validity of machine learning systems are ethically questionable, discriminating systems. As discussed in Section 3.2.1, ethical bias is most often caused by biased data. However, often this biased data needs to be used, because there is no better data of the physical world phenomenon available.

> *Because bias can come from the sample. As I said, it can be a problem of the label, which is probably the worst form of all, because we are using the labels as well to measure fairness, and the labels might come from a biased process. Imagine, you want to predict crime, but you have arrest data. So people might get arrested and they should not be arrested. But you use this data to predict* $crime$*. But who really decides who arrests who is the police. So there is an external entity that is influencing the label of the thing that you want to predict.*

As mentioned above, machine learning service validity has gained little attention in the academia. However, based on the material it is a fundamental issue when machine learning is applied in real-world settings. Echoing Wagstaff (2012), it seems that the research community has been overly focused on improving the correctness of machine learning models while largely ignoring the problem for which machine learning is employed that classical software cannot solve.

5.1.2 Technical quality

Technical quality was one of the most prevalent topics along with machine learning service quality. However, while the literature, as presented in Section 3.1, focuses largely on the quality of the machine learning software, the quality characteristics of software were not as prominently present in the interviews, and only a few of the participants were aware of the ISO software quality model. Instead, the interviewees generally considered technical quality of machine learning systems more from the software *development* point of view.

> *In machine learning development, machine learning product – there are many moving parts: inputs, not only data but also code, hyperparameters, configurations. And then there are many phases: there is training, inference, prediction. And then you always need to update the model. And then, . . . there's deploying of the model or product, operating the servers, monitoring the model and servers. . .*

Namely, an important topic of the discussion over technical quality was the quality of the processes and system infrastructure when the machine learning application was taken into production. It appears that productionisation drew much effort from the development teams – the technical complexity caused by machine learning components, such as models, data, and training, also made the system infrastructure, development processes, and deployment operations more complicated. In addition, the lack of reference for, e.g., architectures and processes, seemed to worsen the problem.

> *The decent amount of that project was just on making the thing an actual usable, reproducible, production environment.*
>
> *When the system is taken into production, it's not possible to directly duplicate the system that has been used for training – like, so that you would have the same inputs in production or the same data sources would behave similarly in production. . . . It could be difficult for many reasons; technical restrictions.*
>
> *We – and many other companies; we don't have a very established way of doing things.*

Several of the interviewees also noted that one of the important features in production machine learning systems is the ability to trace back errors, potentially weeks or months after the occurrence. The participants mentioned versioning of the machine learning system as a prerequisite for this traceability, but it seemed that in many of the described systems, *reproducibility* was to some extent compromised. Although versioning machine learning software "*just like any other code*" appeared obvious for the interviewees, a few of the participants wanted to discuss the topic specifically:

> *Of course, we version all code, just like any other software, but apart from that we also need some notion of data versioning. That might be kind of a new thing that hasn't been done too systematically in the field of machine learning,*

> *until recently. I mean, reproducibility necessitates that we . . . version data, like we version software, because . . . if, for example, . . . one of our algorithms is not working as we though, we need to be able to reproduce the training of the model. We are not able to do that, if we have no idea, what data was used in the training.*

The lack of full reproducibility in many systems might be explained by the fact that many of the interviewees considered reproducibility as an "*emerging theme*" in the context of machine learning. Although tools specifically for machine learning versioning purposes have emerged (e.g., The Institute for Ethical AI & Machine Learning 2020), the novelty of the tooling was seen problematic. Consequently, the increased complexity of system management seemed to cause reluctance towards adopting data versioning as part of the development practices.

> *There are kind of frameworks now – or not frameworks but environments – where they do versioning. They are trying to push this to us, I mean version control, data version control. I hope it can be more natural to use with time, and it will be part of our work, but for now, I think, it's still somehow messy.*

The discussion in the interviews generally echoed the literature only in the lack of maintainability in machine learning systems. However, in the cases described by the participants, the cause was not necessarily the lack of software development skills of the data scientists but the limited resources provided for machine learning development. Although technical debt in machine learning systems has drawn attention in recent years, and the machine learning developers acknowledge the problem, it seems that organisations react more slowly to the issue. Consequently, the increased cost of building robust systems in comparison to the proof-of-concept implementations may come as a surprise for companies, and the developers are not provided the necessary resources to write quality software.

> *The project started as a proof-of-concept, and then the PoC was taken into production, as usually happens. And it actually worked quite nicely, but it was very difficult to improve it . . . and during the proof-of-concept phase there were several experiments that had been tried out, and some of the experiments didn't work out, but the code was still there. . . . But I feel that the situation is a lot better now in that sense, than it was a few years back.*

> *The customers don't yet understand, what else the productionisation can mean. We've noticed that they might be eager to try out proof-of-concepts, but they haven't realised that in order to make those productionable, it can be a much larger investment and a long project. I've ran into this in software engineering back in the days, when the request often was, like, "so, make this small user interface PoC and then surely we can just deploy it?" And only after that they started to look into automated testing and those things. Now we're in a quite similar situation; the customers don't yet completely understand what*

it takes to really integrate artificial intelligence as a part of their production processes.

Overall, in technical quality, the software itself was not considered such a significant problem in system development in that the technical side of machine learning applications did not present unsolvable issues for the participants. Rather, the awkwardness of introducing machine learning specific components to the system and development processes with the traditional software tools seemed to cause most of the inconveniences.

We do machine learning, quite a lot actually, . . . but on the other hand, it's part of the software, nothing more.

5.1.3 Data quality

Data quality was the third prominent topic in the interviews. Given the role of data in machine learning systems, it is understandable that its quality was one of the biggest concerns regarding the quality of the service. Interestingly, *data* was by far also the most often mentioned word in the interviews.

Generally, from the data quality issues discussed in Section 3.2, only *skew* was widely present in the interviews. While all of the forms of skewness that are described in literature could be recognised in the interview material, especially a significant class imbalance seemed to be a common starting point for many systems.

There is always skew. I mean, if you are talking about classification, there is this label that has the most observations and you have a few labels that maybe have a single observation, and you have to make it appear somehow. So . . . there are always unbalanced data sets.

The high prevalence of distribution skew might be due to the nature of the problems that are usually tried to solve with machine learning. Many of the classification applications discussed by the interviewees were different forms of anomaly recognition systems, where the setting of the problem is inherently skewed.

Another, significant problem with data that was present in the interviews was noise. Especially the participants working in consultancies had a common experience of data being generally very noisy – to the extent of making the development of machine learning onto it practically impossible.

We are always kind of worried, like, "OK, is the dataset . . . usable, is it just noise".

The signals were very noisy . . . and it affects the reliability of predictions, how much you need to clean the data. And when you have cleaned the data enough, is there any signal left in it.

It appears that often the data collected by companies is either not *big data* or *big enough* data in the sense that even a moderate level of noise can turn out a blocker for system

development. Notably, the participants from product companies did not generally have such a strong experience of data being unusable because of noise. This might be related to the different business models of product companies and consultancies – if machine learning is central to the product of a company, the model would have been established only on a dataset onto which machine learning is feasible to build.

While the data quality problems that were causing difficulties for machine learning model development heavily reflected the literature, the experiences of the interviewees were less aligned regarding data quality issues that are prominent during *runtime*. Two major causes of problems could be identified from the material, the more prominent of them clearly being "*broken data*". The term *broken data* was commonly used to denote the types of faults in data syntax and integrity that could cause sudden malfunctioning of the system. Many of the interviewees described incidents caused by broken data – in fact, it was estimated that broken input data is "*the most common reason for a system to break in production*".

> *So, data is always broken. Always. There will always be something. . . . Sure, when we start to train a model we clean the data a lot . . . but in the end, the model needs to work with the production data that is just garbage. . . . How do you make sure that your data is clean in different environments, when it's more or less broken in each of them? And it is broken in different ways in each of them.*

> *For example, one client changed [part of their system], completely, so basically all our predictions became useless. It used to be, like, this [sample has label X], but now the same [sample would have label Y]. So basically, all the targets for our model just changed overnight.*

The less discussed problem during system runtime were the issues in data stability, or semantic evolution of data over time. Although, as explained in Section 3.2, part of the evolution can be attributed to the natural distribution drift of data, the distribution change can also be triggered by external factors. One of the participants described that data-intensive systems will always face feedback loops, because the systems do not operate in isolation. In other words, a system will inevitably change its environment, changing also the process that produces the data, and the data along with it.

> *Data quality has this main issue that data quality changes when that data is used. Even though the system remains identical, when some new use for the system emerges, it starts to change. For example, let's say we start to collect crime statistics and publish them in the internet. The system will be implemented perfectly, and it will perfectly predict where the police should patrol. But when people notice that the system is in the internet, they will become selective in where they are buying houses. Then the others start to manipulate, "let's not report crimes because it decreases our house prices". So, even though the system remains perfect, if a single new use appears, it's*

a feedback signal . . . and the data will change.

Although broken data was recognised as a more common issue during runtime in comparison to semantic problems, this might be partly caused by the difficulty of automatic evaluation of the semantic quality of data. While problems in data syntax often produce very explicit errors – and at worst the failure of the whole system – semantic data quality issues silently erode the machine learning model performance. Moreover, if the meaning of the data changes in a way that does not affect the model performance, it may still harm the validity of the machine learning service, as in the previous example. Detecting such issues by looking at the data only was noted to be very difficult or impossible.

> *If data has problems in the sense that, for example, a person's weight in real life is 100 kg but the data says 110 kg, there's nothing you can do about it, there's no way to separate that from the real information.*

While the machine learning literature does not generally seem to report the actions taken to evaluate and ensure data quality, one of the interviewees stressed that it is essential to focus on the quality of data production, collection, and storing, as these processes largely determine the quality of the data. This claim is also supported by the rest of the interview material, although it was noted that the machine learning engineer, building the system, is often not responsible for the data collection.

> *The intormation management representative said that data quality in the company had been the second lowest priority so far. . . . So, because the data hadn't been utilised so much yet, they hadn't invested into it very much either.*

> *The client has like a data lake built, already, and it's actually really good; there hasn't been any big surprises regarding the data quality. And they have a team maintaining it, and if you need some kind of change, . . . they are very reactive to that stuff. That has been a pleasant change. Like, if I want data, it usually exists and it's in a right format.*

Interestingly, although the literature regarding data quality in information systems nearly invariably assesses data using data quality dimensions, the notion of these was completely missing in the interview material. It might be that machine learning developers are generally not familiar with data quality dimensions, because as already mentioned, the research over machine learning widely disregards such detailed assessment of data quality. Moreover, in academia, benchmark datasets are often used, and the results are not transferred into real-world settings (Wagstaff 2012), avoiding issues with data during system operation. Consequently, it seems that more attention is needed to the rigorous evaluation of the quality of data in the context of machine learning. Especially the long-term semantic evolution of data and the feedback loops emerging during system operation appear to be important problems regarding the future world where machine learning systems are increasingly in interaction with each other.

5.1.4 Security

Security in machine learning systems, as described in Section 3.3, was one of the less discussed topics in the interviews. Defensive actions against poisoning and evasion attacks had been taken in systems where the prospect of adversaries was imminent, but these types of security threats were considered irrelevant in many types of applications. For example, machine learning systems that were not connected to the public internet relied on cybersecurity measures to protect the system from adversaries. Also certain application domains were generally assumed to be uninteresting for adversaries conducting these machine learning specific attacks, and the systems in these domains relied on solidity of the implemented cybersecurity measures.

> *We rarely do that kind of public systems, but more of those that are installed within the organisations' firewalls. So in most cases, that sort of attacks . . . that someone is tampering with the labels aren't relevant for us.*

> *In practice, you could feed any data in our system by [creating an item that is accepted by authorities], and then it would be available in our system. But the iteration speed with the current process, to get the data in our system; it's about a year at minimum. And then [with evasion attacks] you just get crappy results, which I think is not very critical, at least hasn't been so far. And to be able to get information of the model and extract the training data, that's not critical in a way because we are using only public data as our training data.*

A few of the interviewees had experiences in training and serving the machine learning system in adversarial settings. Adversially robust machine learning service performance was evidently difficult to achieve, but not only because of the lack of robust defence mechanisms. In real-world applications, the ability of the adversaries to *adapt* to the existing security measures necessitated adopting defence mechanisms that could similarly be adapted; for example, periodically updated.

> *All of our adversaries are trying to find the holes, how to break into the system. Even if we get a 99.99 % accurate classifier on a test set, when they find that one per mille gap, they will push everything trough it. So, in practice, it works so that for a while it works well and after that everything goes through. Then we retrain the model, and then they'll search where the hole is now. So it's like this, all the time they're trying to figure out how the system works.*

Generally, the systems described by the interviewees were centralised and built in cloud environments, where periodic security updates are in principle easy to deploy. Updating might be more challenging in distributed machine learning applications, such as self-driving cars, where the edge devices are not necessarily connected to the centralised system at all times.

Although many of the systems were not thought to attract adversaries as such, one of the interviewees took a more general approach to the security and privacy threats. Complex

and data-intensive systems face versatile security threats, and focusing narrowly on certain types of attacks may leave the system vulnerable to other types of security flaws. For example, sometimes the system itself might not be the target of an attack but provide a means to cause harm somewhere else instead:

> *You have to think about all the ways you can destroy [the system]. . . . "How can I turn it into something negative?" There was a jogging tracking app, few years ago, where they figured out that some military personnel could be tracked, because the tracking data was not held securely. It was quite public. So they tracked running loops of military staff, and realised that it could be used to actually attack military staff base.*[2] *So the app developers had to modify the app. Well, that's because of their perspective was, "we're a California company, we're mostly going to have people trying to be fit, running down beaches and everything else, we aren't thinking about people in Iraq in the military base trying to exercise in the morning and maybe we're opening them to vulnerability."*

While the literature over machine learning security is very focused on defending and protecting the machine learning model against particular categories of attacks, it does, again, miss the system level consideration of security, which is equally important for real-world systems, as noted by the previous participant. Moreover, when implementing security defences against specific threats within the systems, the other potential security vulnerabilities are not the only risk to consider – it was noted that the security measures might also harm the normal users, or disturb the normal use of the system. The defences might even become discriminative, for example, if they do not work equally well for all groups of people.

> *We should have validated it for each individual language and area.*

Privacy

Although the types of privacy attacks, described in Section 3.3.1, were discussed to some extent, the more traditional privacy leaks were generally considered a more acute problem also in machine learning systems. While these privacy leaks could be caused by an attacker gaining access to the data despite the cybersecurity measures, a targeted attack is not necessarily the cause for a leak. For example, the participants were concerned of the privacy risk caused by careless handling and inadvertent disclosure of sensitive data.

> *When you do experiments, you handle data sets less cautiously. Sometimes you get them on your laptop, you play with it, and you forget about it.*
>
> *Those cases are quite common, where someone is able to see something; some information is available somewhere that should not be there. Cases like*

[2] The Strava fitness mobile application tracked its users and shared the data in a heat map, revealing identifiable and sensitive information, such as military base locations (e.g., Lewis 2018; Pérez-Peña and Rosenberg 2018).

these. And then those systems are taken off from production.

The risk of deductive disclosure of identities or other sensitive information was a less discussed topic. Similarly to the other machine learning security threats, this type of privacy risk was not thought to apply to applications that were not connected to the public internet. In addition, many of the participants assumed that although they were collecting certain personal data in their systems, this data would be difficult to access and not interest the adversaries.

> *It's all anonymised with IDs . . . . There is some personal information . . . , so you could probably infer the approximate where this person lives, like, there is definitely personal data. But the only thing that is exposed through the API is the user ID and a big list of content ID:s and . . . scores. So I guess, if you have an API key, you can find out more [sensitive information], but unless you have some way to recover who they are from the ID, there's, at least in my opinion, no huge security risk.*

On the other hand, a few of the interviewees had experiences in privacy-critical domains. These interviewees were generally more sensitive to the risk for deductive disclosure of an individual's sensitive information from pseudonymised or even anonymised data, and knew more of the approaches for preventing such deductions. Still, despite the popularity of differential privacy in academia, only a few of those participants were familiar with it. Moreover, the participants' opinions about differential privacy were mixed: whereas tools providing differentially private algorithms were considered as part of the solution by one participant, another did not find the concept very useful in practice because of the complexity of taking the formal mathematics into reality.

> *Differential privacy– – yeah, it's a nice thought, but taking it into practice is really difficult. . . . In the end, in practice it's a pretty philosophic concept. I mean, yeah, you can calculate it, but then there are all these questions that whether the person who has access to the data; do they have some auxiliary information or are they able to easily get it from somewhere. And then all your estimations are completely wrong, because the person was able to make another deduction about the data.*

It seemed that the discrepancy in the views towards differential privacy was caused by whether it was considered as a framework for calculating the risk for privacy, or as a framework to reduce the risk for privacy. Indeed, the differentially private algorithms do not take into account the different levels of auxiliary data, and the tools often make certain assumptions that do not necessarily hold true in real world (e.g., Wilson et al. 2019). In fact, it was concluded that regardless of the tool for anonymisation, *perfect privacy* is impossible to achieve.

> *In privacy, in the end, you need to accept the fact that you just can't get rid of the misuses. For example, revealing someone's identity – you just can't get rid of it. There will always be one way or another. So, of course, in my opinion,*

> *in these cases you need to consider the benefits against the potential harms. You need to remember that it is possible to get considerable benefits by using and sharing data. So then you need to accept certain unprobable but potential risks. . . . And then you also have to accept that you shouldn't publish the data, even in statistics, if there are no benefits that you can achieve by doing that.*

It was interesting that so few of the interviewees were familiar with differential privacy, although it is one of the most studied definitions of privacy from the security perspective and the only one giving rigorous guarantees for privacy. Differential privacy appears often also in the machine learning literature concerning privacy. In comparison, for example the machine learning specific security risks were generally known among the participants, although not everyone had first-hand experience of building systems under adversarial settings. One explanation might be that the privacy risks that machine learning enables are currently much less present in the machine learning literature compared to the other security risks, and consequently the privacy threats and countermeasures, such as differential privacy, would not have come across.

Data privacy

Data privacy was another of the topics that divided the participants. Although none of the interviewees underrated the importance of data privacy, many of them seemed to trust the law and regulations to require such a level of rigour in data handling that individuals would be protected by complying with them. In contrast, some of the interviewees appeared to be disappointed to the overall state of data privacy today, and had intense opinions of the topic.

> *Privacy is– people keep saying it's dead. Privacy is not dead, it's just we keep giving it away.*

However, the material suggests that it is difficult to motivate organisations to invest in data privacy more than the law and regulations necessitate. This also reflected into how the participants described their experiences regarding data privacy: the topic was generally discussed in relation with public relations (PR) and GDPR. One of the participants criticised this attitude from the organisational side, and called for more responsibility from the companies.

> *I try making everything privacy first. I don't see anyone doing that effectively, until they have a financial reason to do it, or until they have a PR reason to do it. . . . So building privacy-first models is important, and we have a rich set of tools to do it technically, we just don't have the input from the business side or from the organisational side of "why should I be building privacy into this, why should I read checkpoints in where I'm gathering the data from, how long I'm using the data, how that data's stored, how I'm engineering that data".*

As it is generally acknowledged that data is an asset, one cause for the indifference might be that companies are reluctant to voluntarily give advantage to their competitors

by complicating their own access to data. It was also admitted that without prior experience in privacy-first models, the development is likely to be more time-consuming and consequently more expensive for the companies.

In addition to these highlights, the material also demonstrated the overloaded use of the term *privacy*, which can be used in these two contexts of security and data privacy. While a few of the interviewees were thinking of the security risks related to privacy when the term *privacy* was first mentioned, the others turned their attention into data privacy, i.e., the more personal and ethical connotation of the word. This might also partly explain why the security risks related to privacy were not well known – if the word is usually associated to only one of the perspectives, the recent publicity of data privacy, reinforced by GDPR, would steal awareness from the security side of it.

5.1.5 Interpretability

The literature does not make a distinction between the terms *explainability* and *interpretability*. However, based on the interviews, it seems that developers perceive differently the expressions *to explain* and *to interpret*, the first connoting a deep understanding of how machine learning works, and the latter taking a more practical meaning of a simplified rendition of the phenomenon. Because this work discusses the topic in the latter of these meanings, in contrast to Section 3.4, the term *interpretability* is used here.

The discussion related to interpretability had two main themes: interpretability for developers and interpretability for end-users. In comparison, the literature largely focuses on the technical implementation of interpretations, on the more abstract quality attributes, such as reliability and accountability, that can be improved by increasing the interpretability of the system, and on the understandability of the interpretations. The first two of these literature perspectives to interpretability reflect more what is called “interpretability for developers” here – most of the participants were using interpretability simply to better understand and assess the model logic and to debug the system.

> *In the beginning we started to collect samples for those difficult cases where the model failed, and then we checked if we had some problems in the data, in [the feature extraction], or in the model. . . . And then the explanations [for the end users]; these we can use ourselves too. We’ll find quicker whether the problem is in the data, in the model, or in the feature extraction.*
>
> *In the development phase, when we were looking at how well the models work, we tried to understand, like “okay, how is the model weighting different things . . . ”. I mean, there was a feature, and we looked if it has the same effect on the prediction that a human would think – or how we had though – that it would have.*

Against the literature’s suggestion of using interpretability for accountability and liability, it was noted that interpretability does not necessarily satisfy the regulatory obligations

regarding the traceability of decisions. Similarly, while the literature proposes using interpretability for countering discrimination, the idea was heavily criticised[3]. Consequently, the uses of interpretability for developers, apart from model understanding and debugging, appeared scarce.

The academia has also studied "human-friendly explanations" (e.g., Carvalho, Pereira, and Cardoso 2019), and the importance of those increase when interpretability is directed for the end-users of machine learning systems. In most systems the end-users cannot not be expected to understand as detailed interpretations as the developers, and consequently the manner of representation of the information must be different. Most of the participants discussing the topic did state this in some form, but those who had studied the topic seemed to have a much clearer image of the consequences regarding the practical implementation.

> *One side of this is the technical part, . . . , but it's more about psychology. . . . It is actually quite well studied what type of explanations people consider useful. Generally, . . . if you go to see a doctor and ask: "why did you predict that I have a flu", you don't want to hear a narrative of how the brain of the doctor is working; you want to hear: "okay, which of the symptoms revealed that I have this flu". And you don't want to hear 400 explanations; you want to hear [a few], selected reasons that will appropriately explain the phenomenon.*

However, in comparison to developers, the rationale for interpreting the machine learning predictions to end-users is less obvious. Despite the concept of human-friendliness of interpretability, the literature still seems to lack insight into what kind of interpretability actually benefits the end-users of machine learning systems. While the academia has focused on how to transfer information of the system logic as accurately and concisely as possible, there is less notion of *what* to interpret and *why*. Based on the experiences of the participants, the reasoning presented in the literature, such as curiosity and safety (e.g., Carvalho, Pereira, and Cardoso 2019), applies poorly to the end-users, because generally the users are not specifically interested in the machine learning component of the system, but performing a task using the system instead.

> *It's what the user experience should be, and the machine learning supplements that, rather than thinking about "How smart can I make this? Now, let's present all that smartness." . . . What you want is the best experience as quickly as possible to reduce it to the minimum amount of information that is transferred. "I don't need to find out about what data the system's got, I just want to make sure the result's correct".*

It seems that the lack of understanding for *why* and *what* should be interpreted had caused impractical applying of interpretability, leading into an inconsistency between the experiences of the interviewees regarding the usefulness of interpretability. While a few participants described cases where interpretations had practically been indispensable

[3]This claim is further elaborated in Section 5.1.6.

for making the actual predictions useful, several had experiences from cases where the end-users had not been interested in the interpretations at all.

> *(Describes a customer churn forecast) So we showed the results to the client but then they interrupted us, like, "the solution looks really nice, but it's completely useless to us". The point was that there they have some machine predicting that a customer has a 60 % risk to migrate, but if even the salesman – in the entrance going to see the customer –, if he has no idea what's wrong with the customer, there's nothing he can do about it.*
>
> *So it was a nice feature, it was cool, but in practice no one seemed to care, except for the team.*
>
> *Nobody cares.*

Judging from the experiences of the interviewees, value of interpretability for end users is generally not *intrinsic* but *instrumental*. In the cases described by the participants, the end-users found the interpretations beneficial only when they were necessary for the user to carry on with the task at hand. Interestingly, in the named cases, even assessing the reliability of the predictions was not interesting for the users, if the interpretations were otherwise not needed. In fact, the only exception to this general lack of curiosity that appeared in the interviews were systems utilising the end-user's personal data for the predictions.

> *When you say "hey Siri", you don't want it then tell you all things it's looking at; all the resources, looking what your data is and then tell you. You just want it to respond with the answer. . . . So you don't want that kind of thing – until its personal. Until you're thinking, "how the hell do you know that information". This comes through like ad networks, when you sit on your couch, pick up your phone; on Instagram it comes with an ad, and you're like "How the hell do you know I was just talking about that and what was happening". That you realise it was a conversation I was having 20 minutes ago. So that's when the paranoia leads you to want to know, what's going on with that.*

It could be argued that systems using sensitive data should provide interpretations to reduce risks related to sensitive decisions and to provide ethical transparency. However, it was noted that interpretability increases the security and privacy risks, because it gives adversaries more information of the system. Consequently, in sensitive applications the interpretations cannot be too detailed.

> *. . . I do think you should have an option to ask, "Why you got that result". It's not always possible to show you exactly those answers. And if you build privacy first machine learning models, it's very hard sometimes to show the traceability of how you got that information as well, because that leaks certain information back out as well.*
>
> *If you explain the model, in a way, it would be nice to always explain the model*

completely. However, this also gives these trolls, or people who try to attack the system, a lot of knowledge about how the system works, and it's easier for them to trick the system and get into the system.

5.1.6 Fairness

As in the case of machine learning service quality, fairness is discussed here as an independent quality characteristic. In comparison to the literature perspective (e.g., Gudivada, Apon, and Ding 2017), it seems that the issues that fairness presents to the machine learning development should not be considered as data quality problems because of the complexity and pervasiveness of ethical bias. In addition, this organisation reflects more the increasing body of literature around the topic.

Similarly to security, machine learning fairness was a topic of less experience among the interviewees. Although the participants were generally watchful for potential problems in fairness in their work, not many of them had been involved in development of systems where discrimination would have been a fundamental problem. Moreover, a few of the interviewees noted that ethical issues are not as common as could be thought by the discussion in public, because currently machine learning is often applied in non-social contexts, such as industry process improvements[4].

The public discussion over algorithmic fairness received criticism from the participants also because of the apparent misconception about why algorithms discriminate. According to the interviewees, the public discussion often ignores the fact that discrimination already exists in the society and is only made known by machine learning. However, the lack of knowledge of discrimination in the current practices creates the public an illusion that issues in fairness are only introduced by algorithms. Consequently, the interviewees thought that algorithmic discrimination is also a positive problem, as it reveals the original, underlying cause of the discrimination.

People are not used to quantify discrimination in decision-making, people are not used to quantify the discrimination of the predictions of the doctors or of the prediction of the judges.

This (discovering ethical bias in machine learning systems) makes the problem visible, and it can be solved. There are ways to reduce discrimination mathematically, but when it's part of people's normal behaviour, this current process, the discrimination can be neither seen nor fixed.

I think it is great that the algorithms reveal hidden biases in people's heads.

In addition, it was noted that the responsibility of addressing the algorithmic discrimination cannot be on the machine learning engineer only. In fact, one of the most central aspect of fairness that emerged in the interviews was the dependency of fairness on policies. Although there exist numerous ways of defining fairness and the problem is

[4]It should be noted that it was predominantly the Finnish participants who had this experience.

nowadays well defined, many of the mathematical fairness definitions are mutually exclusive (e.g., Kleinberg, Mullainathan, and Raghavan 2016), meaning that improving a certain fairness metric can make another substantially worse. One of the interviewees emphasised that a *policy context* is therefore necessary for deciding which behaviour of a system can be considered fair, and further for choosing the fairness metrics to focus on.

> *Google and other big tech companies have also incorporated some sorts of (fairness) audits as well, and tools for measuring that, but they are lacking the policy context. They are lacking this societal context of— you know. You can have the same machine learning model discriminating, and that discrimination can be bad, if the intervention is assistive, or it can be bad, if the intervention is punitive. . . . So, just having the tools is not enough to allow the policy-makers to really understand what the results are telling them. . . . There really is no* one *fairness goal that fits all projects.*

Although the machine learning engineer would not be in the role of choosing the fairness metrics and mitigation strategies for discrimination within the systems, it appears important that they interpret the effects and trade-offs of the different types of mechanisms for those who need to take the decision. Specifically, one of the participants underlined the inevitable existence of "*tension between fairness and utility*" – mitigating discrimination generally decreases the machine learning model performance, as "*there is always a point where you can no longer optimise for accuracy and for fairness*". While this compromise might not be intuitive for those deciding on the policies, it nevertheless seems necessary to consider, when choosing the mitigation strategies.

> *We need to educate the client to be aware that it's not just about maximising the global utility. There are some trade-offs that, if they really care about these values, and they want something to be in practice, they have to make an informed decision. Because there is always this tension on the utility versus fairness trade-off. . . . In the public sector it's something like that, on the private sector it's more. Are you willing to lose specific money to ensure that you are fair? . . . So there is always this tension between fairness and utility. And often people are not aware of it.*

Unfortunately, this participant confirmed that there is no generally applicable rule for choosing the best mitigation strategy, although in similar cases, i.e., for certain domains and groups, the strategies are likely to be the same. Consequently, the proposed solution was to test many approaches and select the one that produces the most optimal outcome. However, a remaining problem was brought up – there usually is limited knowledge available from the different groups that are using or that are affected by the system. For example, often the users or subjects of systems are anonymous, making evaluating the fairness and mitigation strategies difficult.

> *So there are two things here. One is the quality of the data to train the model.*

> *One concern. The other concern is that if I have accurate data to evaluate the fairness. . . . If you don't know, how can you really measure?*

Interestingly, although the literature often mentions using explainability for assessing fairness and potential discrimination (e.g., Doshi-Velez and Kim 2017; Sokol and Flach 2020), the concept was criticised in the interviews. As explanations can only show how a machine learning model uses its input features, the explanations will miss the correlations of those features to the sensitive variables which exist outside of the system. Consequently, the model may use sensitive features in its predictions, and individual predictions may be affected by sensitive variables, but the group level fairness metrics remain at a good level. Similar criticism can also be found in the recent literature (e.g., Aïvodji et al. 2019; Slack et al. 2019).

> *But people are still very academic and theoretical on this regard. They think that if, based on explainability, the most important reason (for a prediction) is gender, then this explanation shows that this system is a discriminator. I say that this is misleading, because if that happens, it might be because gender is a proxy for other things. Imagine: gender might be a proxy for income. So the problem is not that the system is using gender. The problem is that the distribution . . . from the society has inequality on payment between genders. . . . And in the law, in many audits for fairness in the group level, if there are no significant differences, it's actually relevant to use gender or race in the model.*

5.1.7 Ethics

> *One thing that I think is missing here*[5] *is the quality of the outcome. Impact. . . . The other quality attributes don't really matter, if the outcome is good, or the other way around, nothing else really matters, if the outcome is bad. . . . Impact is about the final, real effect, that follows the decisions of the AI.*

As the ethics presented in machine learning literature focuses largely on intelligent, somewhat futuristic, agents (e.g., Cervantes et al. 2020), it was surprising that many of the interviewees raised ethics for discussion proactively. In contrast to the literature, however, ethics discussed in the interviews did not focus on how ethical and moral considerations could be trained into the machine learning model. Instead, the interviewees were more concerned of the ethics during system development, and the *impact* that the system has on its environment.

> *(On fairness) I believe in terms of impact to society. You know. You are developing decision-making systems; they often have a threshold. And what happens – there are people below the threshold and there are people above the threshold and they are very similar, but they will get different predictions. So I*

[5] Refers to the four quality themes presented in the interviews.

> *don't see that it would be feasible to really to ensure individual fairness when the world by itself is unfair . . . and it's all about money and these constraints. . . . I believe in group fairness, . . . because thinking of society, you don't want to increase inequality in the society – on group level, not on a specific "me versus you" level.*

Interestingly, one of the interviewees highlighted that the responsibility for creating machine learning systems with a sustainable ethical trace is not only on organisations defining the machine learning products, but on the individual developers building the systems as well. In comparison, the academia generally focuses on how organisations' ethics affect individuals' ethical decision-making (e.g., Elango et al. 2010; Schminke 2001). The perspective of this participant is noteworthy, as it reminds that the organisation ethics do not completely override the individual ethical decision-making (e.g., Elango et al. 2010). Consequently, the participant called for higher ethical standards in decision-making from individuals.

> *We as humans, we should be able to say "I'm not happy with shipping this until these things have been put in place". And we should be very okay with doing that. We shouldn't just be getting excited over solutions, we shouldn't be getting excited over paycheque and we shouldn't get excited over delivering something, we should get excited about "Have we done the best job we can with this solution to hit all of our ethical benchmarks?"*

Although there exist certain laws and regulations that strive for ensuring the ethical treatment of individuals, it was demonstrated that regulations cannot replace the above-mentioned ethical understanding of individual developers. These laws or regulations, such as GDPR, might attempt to protect the equality, privacy, and rights of individuals, but they were seen incapable of guaranteeing that the systems meet the ethical objectives that the regulations eventually aim at.

> *I still believe it's important to measure the impact, not just the process and the treatment. . . . And there's also an important relation to legislation. . . . Because the legislation is really about the process, not the impact. And that's where people are washing their heads, because I said "oh they didn't use gender on my credit approval". "I just use income." "But you know that income for men and women correlate." "But I obeyed the law". If you care about the impact, it's after a year, how many loans did women get in this bank versus men. How many loans people from specific minorities got in this bank. So starting to care more about the impact and not just on treatment. But caring about impact is more tricky, requires more effort.*

> *I think the problem with GDPR is it's one of those regulations that comes in and everybody somehow drops their ethics and thinks "well this regulation handles ethics I don't need to think about ethically any more . . . ". But GDPR is not a benchmark; it's like the lowest level you need to be at, and we should*

> *really be aiming higher on how we design systems. . . . And when you come to privacy-first design, it's something you should be very proud of – building a very intelligent system for somebody and saying this is how we protect your information. Like explainability; that's a better thing to explain. How we protect your information.*

However, while ethics regarding machine learning was generally considered "*an important thing*", the subjectivity of ethical decisions was seen problematic. For example, certain ethical problems can be tied to culture, and not all individuals are equally sensitive to ethics. In addition, ethics can also be considered as a private matter, and one of the participants noted that this makes ethics difficult to teach.

> *And it's kind of difficult, is it bad or not, because it's based on our expectations and our views of the world and our own ethics.*
>
> *I'm not really that sensitive, but I should be sensitive about it.*
>
> *It's individual, personal system. . . . That's a lot harder; you can't teach that, because then you're being kind of a prick.*

Because of the lack of organisational support and personal sensitivity to ethics, it seems that sometimes the only driver for building ethical systems are the public relations. While the problem could be thought to be similar in the field of classical software engineering, it is evident that machine learning increases the capacity of digital systems. Consequently, it can be argued that also the ethical questions become more important to consider.

> *And people don't do that so often unless it's PR driven. And I understand there's value in the PR and there's value in reducing your risk. But there's also value in just being a good person, building systems in an ethical way. . . . As data scientist, you handle so much personal data, you're building so many systems that are potential for violation, tracking and all of these things – we should understand it as an ethical thing and not a PR thing, not marketing thing and not just a final slide in a teaching program.*

5.2 Assuring quality

> *When it comes to machine learning, there is absolutely zero quality assurance.*

This excerpt summarised the observations of many of the participants regarding the quality assurance of machine learning systems. It seems that the lack of tradition regarding the topic was resulting in a high variance in thoughts and levels of effort for preventing the quality issues, while in comparison, the experiences over the quality problems in machine learning, discussed in the previous sections, were rather consistent. In addition, most of the participants were immediately thinking of *testing* when quality assurance was mentioned, and not all of the described activities were necessarily perceived as quality

assurance. This section presents the discussed activities and components that the material suggests to have the most significant impact on the quality of a machine learning system.

5.2.1 Data and application domain understanding

The interview material echoes the CRISP-DM framework in that business and data understanding are elementary preliminary steps of a data science project or product. While in classical software engineering the first quality assurance activity is usually planning of the desired quality attributes and their assurance, in machine learning systems it seems that this step is unlikely to produce realistic output before there is an understanding of the capabilities and limitations of the available data. For example, one of the participants described a project where the targets had been defined prior access to the data, and consequently needed to be revisited after the data was available.

> *In the beginning of the project we had some goals, but when we saw the data, we needed to discuss, because [it would have been impossible to reach the target]. And then, well, we had to start from somewhere, so we did something. So we had to do some compromises with the KPIs*[6]*.*

Moreover, as expressed by CRISP-DM, building data intensive systems is characterised by iterativeness, as the understanding of the data grows during the system development. In many of the cases, described by the interviewees, predicting the issues appearing in the system would have been difficult or impossible early on in the development, and especially data seemed to be a significant source of uncertainty. Consequently, also the quality assurance activities are likely to evolve as the model development proceeds.

> *It's a learning process. I mean that your client, your company, should know that this product will have some issues with time, but with more time it will be more stable. So at first we should expect some noise and some fluctuations.*
>
> *And it's difficult to think of all the problems that might come up, when teaching the models, because our sample was just too small to be able to detect these problems in that phase.*

As noted by the previous interviewee, an important area that is affected by the growing understanding of data and the application domain are also the fault tolerance mechanisms, and the verification and validation activities. For example, a few participants brought up the potential stability issues of dynamic, online learning systems. As demonstrated in Section 5.1.3, a system will interact with its environment via data, causing the system to gradually change its environment and further to have an effect on itself. This feedback loop between the system and its environment can be difficult to predict in advance, and according to the interview material, a deeper understanding of how the data evolves and is being used outside of the system can be useful when attempting to control the loop.

[6] Key Performance Indicators

One approach for deepening the data understanding, proposed by one of the interviewees, was to gather more metadata and provenance information[7] of the data. Although the literature does not discuss metadata in the context of machine learning, the interview material seems to support the concept. As discussed in Section 5.1.3, two identical-looking data sets can significantly vary as of their semantics depending on the process that produced the data, and metadata would help in understanding these hidden differences. Similarly, the semantic evolution of data can be understood better with the knowledge of data origin.

> *If you create even one new use for the system, or even slightly change the existing use, the original data starts to change with respect to the environment. So, we need to have metadata, understanding of the data in the system and of the data origin, its uses, because these change the data, its meaning, and its effects.*

Metadata appears also essential for countering discrimination and enhancing fairness. One of the interviewees emphasised the importance of understanding the data and its origin when assessing the data quality regarding fairness, because, virtually, "*every decision that the human makes, every parameter that you define, is also biased*".

> *When do we really understand that the data that we use for training is really biased and is going to produce really biased models? So once again, it's a question of how was the data collection . . . . All these design specifics will have some sort of impact on the quality of the data. . . . So, [in the beginning] we are really trying to understand the data they have, how the data was collected, how recently the data was introduced, when it was introduced, if there is some sort of noise that can be there. We really want to understand where data comes from.*

It appears that certain problems require even more deliberate search for metadata. For example, an issue in machine learning fairness, discussed in Section 5.1.6, is the lack of knowledge of the groups that the discrimination mitigation mechanisms attempt to protect, making evaluating fairness impossible. In comparison to metadata discussed until now, it might not be enough to look at the immediate components along the data pipeline for finding this information. Consequently, it was proposed to search for external data, such as public databases, that can give more insight into the various, invisible factors in the data.

> *It's important to think about external data that can give us more socio-economic background, because often people have just data from the users of their systems. . . . You don't have the income or socio-economic background, but we can use external sources like [statistical databases] to see income*

[7] In general, data provenance denotes the origin of data and seeks to explain how the processing and transformation history of data affects how it is now (J. Wang et al. 2015). Data provenance has been proposed as a method for evaluating and managing the semantic quality of data (e.g., Buneman, Khanna, and Wang-Chiew 2001).

> *and inequalities across different zip codes or even blocks, really understand specific backgrounds in education and how that can be grouped with the socio-economic indicators, so how can we reach the data that they have to give us a better picture of the population that they are serving.*

Although the literature generally mentions *data understanding* as one of the important steps when building data-intensive applications (e.g., Chapman et al. 2000), for a machine learning system it seems equally important to have an understanding of the application domain as well. This is important not only for translating the business objectives into optimisable machine learning targets, but also for specifying constraints that are relevant in the domain. For example, legislation and regulations can set restrictions to the operation and level of automation of the machine learning system, but constraints might also originate from the existing system as part of which machine learning is integrated.

> *It's important to know what is specific legislation in the area that you are developing data science on. . . . You need to know specific regulation for specific region where the model is going to be applied. And then how can you guarantee that whatever the spirit of the law that is in the regulation, how that gets encoded in your practical data science constraints and goals. So there is this translation goal . . . and that might be more or less tricky depending on the legislation.*

> *Especially, when we start to speak about automated decisions, I've run into this. [You have a single model that is used in a similar process across clients] . . . , it might be that you have three clients where it works perfectly . . . . Then you get a client where the process works differently . . . . And then it happens that if you have a certain type of false prediction, the whole process gets stuck. . . . So, in short, the same algorithm works in the same way and with a high enough accuracy that has been validated together with the client. But, depending on how the the process of the client has been defined – it can be defined in many ways –, the quality from the client's point of view can be very low or very high. Depending on how the system works with the process. . . . So, when you start to automate things, understanding the client's process is really difficult, because there are assumptions and rules that you don't necessarily know.*

It was noted that sometimes the legal or regulatory constraints might even prevent using machine learning because of the invisibility and uncertainty of the decision-making process. For example, a few of the participants mentioned that there are auditing frameworks that do not have a process for opaque, indeterministic systems. Similarly to policies, also many other environments, such as industry processes, have traditionally been built to work with rule-based systems, presenting various problems when introducing probabilistic decision-making. As demonstrated by the previous interviewee, it is important to try to understand the environment where machine learning is applied in order to avoid these problems.

5.2.2 Design

As mentioned, planning the quality of a system is often the foremost quality assurance activity in classical software systems. The interview material does not suggest that machine learning systems would differ from classical software in that quality ought to be planned – the desirable quality attributes are more likely to be achieved, if already the design of the system takes those attributes into account.

> *I think about what the user experiences, before I start thinking about how to apply intelligence to it all, or logic. So, I start off with a design program, and I start designing the user experience completely. And that is entirely the quality base that the product should have. When you're thinking about delivering it to somebody, you're thinking about what their experience is. Bugs come down to that – they're bad experiences. So you try to solve those as part of quality checks. But the entire user experience should account for that. And it even comes to understanding that what happens, if there is a fault at this point within this design. How does the actual product handle the fault, in order to provide the best experience to the consumer. . . . Think about all possible faults and how your app can adapt to that.*

Similarly to the previous interviewee's principle, the most noteworthy design considerations that emerged in the interviews can be discussed in two categories: the design should account for all desired quality attributes, but also for the mitigation of the negative behaviour of the system. While the rationale for designing the wanted characteristics of a system is quite intuitive, it appears equally essential that the design also considers the prevention of the potential, unwanted behaviour. Despite the testing efforts, the correct behaviour of machine learning systems seems difficult to guarantee, drawing attention to the mitigation of the effects of false predictions and erroneous system operation. Consequently, design has an important role in estimating, preventing, and mitigating the possible error scenarios and faults within the system.

> *In the end, even the simplest classifiers are somehow black boxes. You can't evaluate the more complex models with code reviews. You can check that the code is valid, of course, but you cannot review the trained model to see if it's correct. . . . So, it calls attention to thinking about how the system works. You need to think about validation of the inputs – what types of inputs are accepted in the first place –, and then all accepted inputs must have an accepted output. The whole space must be somehow covered. . . . In machine learning, you can't [review similarly to classical software], because the data and input are always such a significant source of uncertainty.*

As noted, data is unarguably one of the most significant sources of uncertainty in machine learning systems. Unfortunately, as described in Section 5.1.3, data syntax problems are frequent. Consequently, it appears clear that it is not safe to expect data to be syntactically correct or complete. One of the interviewees noted that it is important to focus on

data validation as part of the machine learning system runtime infrastructure, instead of *data cleansing* before the model training.

> *I think it's a mistake that many fall into; to make the data too clean throughout the model development and then the model won't work in the real world. . . . The majority of data validation is that you just check that the data is somewhat in the right form and you make sure that you don't accept missing values if you can't handle them. That you recognise missing values, recognise absurd values – like minus infinity, plus infinity – and so on. You'll get very far already with those. But to clean them from the data and to expect that you're not going to see any of that again – I mean, that's just not reality.*

Another area where the importance of failure prevention appeared in the interviews was the security of machine learning systems. As described in Section 3.3, the machine learning security defences are not guaranteed to protect the model from malicious actors. Apart from defending the systems according to the best practices found in the literature, the only additional, proactive defence mechanism that was mentioned for the security issues was building mechanisms to detect successful attacks during runtime and mitigating their effects.

> *(Discussing the security flaws) I do think it is one of the weakest parts of ML, as well. . . . So I think it's something important to look at. But when it comes to things like poisoning, the problem is, it's hard to test for those kind of effects. So there needs to be methodologies that are quite easy to adapt to the machine learning pipeline to understand how that affects the model. And again, it's the testing for unknown behaviour. Like, when I talked about design originally, and saying you design a system to fail, and you make sure the failure is acceptable . . . . In machine learning models it's the same. So you remove all the user interface stuff, you remove all coding, you look at the machine learning. How you get that data in, how you build a model – it should be build in a way that handles those failures, handles those attacks in the right way. All these test those attacks in the right way.*

5.2.3 Verification & validation

Many activities related to verification and validation of machine learning systems were discussed in the interviews. However, given the connotation between *quality assurance* and *testing*, it is understandable that testing was usually the first mentioned of these activities. More precisely, the discussed testing activities often concerned the machine learning model correctness, which was ensured against a test data set.

> *Of course, testing of the machine learning functionality; . . . before pushing the model into production we test that it's working on a test set.*

> *And one kind of machine learning specific aspect is that you usually have a*

model, you retrain it every now and then, and you have to make sure that the retraining doesn't decrease the quality, or the accuracy, of the system.

So you have to test the actual model, but this is, maybe, the straightforward simple thing.

Beyond correctness

It seemed that the majority of discussions over testing the machine learning model focused on testing for correctness with data drawn from a similar distribution with which the model was trained. One of the interviewees noted that it is easy to preform this type of "positive" testing, but verifying the *robustness* of an opaque machine learning system can be more challenging. In addition, the interviewee thought that the importance of testing the system robustness is more highlighted in machine learning in comparison to classical software, because the acceptable functioning of opaque machine learning models cannot be formally verified.

One of the characteristics (of machine learning) is that they (the models) are always black boxes, in a way. . . . And it puts a lot of pressure on the testing activities. So, in my opinion it underlines the importance of estimating risks, faults and failure modes in advance, and testing for them. . . . In real world you are going to face situations where the input is outside of the training dataset. . . . So [the acceptable model behaviour for the whole input space] has to be ensured somehow through the testing activities. . . . In my opinion, it's more important that the models are robust, and not that the data should be good. . . . So, if the model behaviour hasn't been tested, the output is completely– it's impossible to know what it is. So somehow, I think that is something that must be done through the testing, because you can't do that by using logic.

The rest of the interview material supports the views of this participant. While it was mentioned that the opaqueness of machine learning is not a similar problem for the machine learning algorithm developers in comparison to the application developers using these algorithms, the vast majority of machine learning application development is done by using existing libraries. In addition, the interview material revealed no cases where the false predictions, which in general are quite inevitable in machine learning, would have been considered a problem once the desired correctness level had been achieved – unless those false predictions were plainly unacceptable with respect to the given input or the system context. Although these examples support the previously discussed principle of designing and building fault tolerance mechanisms within the system, it appears equally important that these mechanisms are also tested.

The participants also discussed verification and validation of other characteristics than the machine learning service quality. One of these topics was validating the system security and privacy. Although the design phase for security was described analogous to designing fault tolerance, it was argued that security validation is advisable to do with

an external *red team* in order to not leave "*side channels*" into the system. However, the often limited resources for red teaming were acknowledged.

> *And so you have to think about all the ways you can destroy it (the system). And all security models; when you're trying to test a system, you always need a red team that's going in and trying to break that system to make it effective. . . . Machine learning models have so many variabilities, and these cases need to be checked. . . . So those kind of things need potentially a cold start, somebody from outside coming in and going, given a free minute to destroy the system. And that's essentially successful red teaming. . . . Within small teams it's hard to do that. So you have to somehow take that role on yourself and think about attacking the system. "What would I do, . . . all the methodologies I can think of to break the system, how can I break it".*

Another machine learning characteristic of which validation was discussed separately was fairness. One participant argued that apart from the development team evaluating fairness during model development, the organisation using the system should conduct an "*independent audit*" for fairness. The prospect of *fairwashing*, discussed by Aïvodji et al. (2019), seems to promote this idea. However, as fairness might not be not comprehensively addressed by the law (e.g., Hacker 2018), the organisations are not necessarily proactively interested in auditing, given the apparent apathy regarding ethics in general.

> *There are specific stages in which you should measure. So, when data scientist is developing the model, usually you don't develop just one model, you develop thousands of models, and you pick one that maximises whatever performance metric that you want to optimise. And then that process of model selection should also comprise these fairness tests. . . . That's one step. The other step is that whatever entity that is going to use the model, they should also audit. So then, it's the team – can be a different company that develops the model – they do audits. And then they give the model . . . to the client . . . . These organisations, they should also audit the model. Because it's an independent audit that they should do.*

Finally, also software testing was discussed: software defects appeared a similar problem in machine learning systems as in classical software systems. In addition, it was noted that the performance tests for the machine learning model are advisable to replicate at system level as well, because the software serving the model predictions might also contain defects.

> *In this case, you have a model that works well, but the predictions were mixed up. So you also have to replicate the tests out of this black box. This is one thing that should be taken into consideration. That "OK, my model works well, but does my whole process work well".*

> *But it was a software bug, so in that way it's really important to also continue with these software practices, when you go to production. . . . It took us a long*

time to find the bug – we were several weeks like "why is this happening, don't run the model again".

Although software testing seemed to be an obvious activity for all of the interviewees, some noted that testing the machine learning specific software comprehensively can be difficult. The literature might provide an explanation to this: the machine learning model code often resembles a configuration file (Sculley et al. 2015), making it potentially difficult to write tests that would not need updating on every change.

You cannot write too detailed unit tests, because they will break all the time. And then they easily become slow or unreliable. But for example, we've found it useful to test that the weights of each of the layers in the model must change during the training.

Reviews

Apart from testing, several of the interviewees mentioned also reviews as part of their verification and validation practices. In software engineering, modern code reviews or *peer reviews* have been found beneficial in detecting software defects, improving code quality, and supporting team awareness of the software (Bacchelli and Bird 2013). Unfortunately, a few of the participants had an experience of being the only data scientist of the development team, and consequently reviewing the machine learning code was considered impractical.

Usually, when you work on this, you work alone. Maybe on a specific model, I'm saying. So, unlike a regular software product, you don't really do code reviews or peer reviews.

In addition, as noted by several of the interviewees, the machine learning model code reveals little of the logic and correctness of the trained model. However, in place of a traditional code review, one of the interviewees recommended a *concept review* for the model. The aim of a concept review is assessing whether the model is conceptually reasonable as of its input features and their relation to the output.

And then you easily forget that – yes, you do a code review, but in principle – you should also do a $concept$ *review for the model. Like, think if it makes any sense. The model takes these features in, they have been computed like this, does it make any sense.*

While concept reviews could benefit in verifying the model logic, it seems that they could also efficiently serve as means of knowledge transfer, if the model author explains the concept of the model to the rest of the team. Collaboration on a single model was considered challenging by a few interviewees, because the logic of the system is not explicitly visible from the code, as in classical software. Concept reviews might therefore be useful, because they clarify the otherwise invisible machine learning specific software logic to the other team members.

It's not really easy to look into other's works. Especially, sometimes preprocessing . . . can be a very messy environment, unlike regular software products.

If there are two data scientists collaborating, then if they can somehow share the overview idea, what each part (of the machine learning software) is doing, then it's easier to communicate.

Monitoring

While testing and reviews were the most discussed verification and validation approaches during development, another important area is evaluating the system quality during system operation. Virtually all of the material regarding this focused on monitoring: several of the interviewees discussed cases where monitoring had revealed a problem in the production system or in the incoming data. However, while extensive production system monitoring was generally considered as an extremely beneficial practice, some of the interviewees had experience of rarely having the resources to implement such a tool.

When you make modifications, you easily go backwards on some other direction.

It was like, looking at the dashboard one morning, like, "oh no, the sky is falling".

When we are in production, we monitor the data, like, if there are some unknown values; some new class or negative values or missing values. We try to monitor the prediction quality, but it can be difficult . . . if the feedback loop is slow.

In the ideal case you would have some kind of monitorings or some dashboard that shows a curve that starts to go down when something's wrong, but of course that rarely happens. First of all, because it takes time to implement that, and secondly, it's difficult to define what exactly the curve should measure.

Monitoring was also discussed in the context of individual quality aspects. Notably, monitoring was seen quite essential for ensuring the continuity of fairness of a system. As demonstrated in Section 5.1.6, fairness can be affected by nearly every decision in the context of a system, and moreover, the discrimination mitigation mechanisms necessarily *aim at changing* the system environment, causing both the training data and the data to evaluate fairness with to change. Consequently, it is rather intuitive that one of the participants suggested monitoring fairness similarly to the machine learning prediction quality.

So you should continuously measure and monitor for fairness in the same way you monitor performance. So if you measure that daily, you can do daily, although, . . . it's good to have some sense of monthly or yearly, or on a rolling

window of one year, to measure the fairness of the system in production.

Timeliness of verification & validation

It was also noted that depending on the project scope and phase, it can be inappropriate to implement exhaustive verification and validation mechanisms for the system. Moreover, similarly to the design, the verification and validation approaches are likely to evolve as the understanding of the data and application domain increases. Consequently, the approaches are likely to be more relevant to the system, if they are implemented at a suitable phase during the development.

> *But in some systems those kind of things are not that essential. And it depends on the scale – if you have a small project, it can be completely premature [to implement extensive testing] because it can just slow down development.*

> *You can write a software test, but you can't write data tests; you have to find other ways of returning the quality of your data. And most of the time you'll figure it out as you're building your model, going back to your data.*

Nevertheless, it is arguably unwise to push too much verification and validation debt towards the end of the system development. One of the interviewees brought up that in classical software engineering testing is rarely allocated the time that it needs. Machine learning system development is unlikely to differ from this. Moreover, it seems that the potential challenges that machine learning presents to verification and validation (see Section 3.1) might make these activities more time-consuming than in classical software development, placing more pressure on planning the quality assurance of the system.

> *We had the waterfall processes, well, years ago, where you go for requirements, development, testing. And it always ended up, testing never happened. So, now we have this agile process, but nobody thinks about testing in the agile process. They say "it is in there, it's in this little loop, we are doing testing at this checkpoint", but that testing is never really thought about. It's like mini waterfalls going over and over again. "We'll push it to the end; oh we didn't get enough testing, fine, next time." Until you get to the end of the project, and there's a huge push of your user acceptance testing, quality assurance testing, before you go alive, rather than doing it at checkpoints during development before it goes alive. Because you can't, if you push all of the quality checks to the end of a project; you still have to schedule them. But we don't do that. We don't do that in other fields as well. We don't do security checks, we don't do privacy or regulatory checks. We always push everything to the end; only the writing of the code is iterative, which is not really how agile is meant to work.*

5.2.4 Documentation

Based on the interview material, it seems that machine learning poses certain requirements for the explicit documentation of a system. More precisely, the invisibility of machine learning, which has already been discussed in many contexts, also makes sharing the development work difficult. As one of the interviewees mentioned above, "*It's not really easy to look into other's works*". Although knowledge sharing between development team members seems to be beneficial, as discussed earlier, it is not always possible to transfer the understanding of the system in person between the developers.

> *When it comes to software development, you provide a library, you have an API, and that API should be well documented, and you ship that library and the API for the developers. Machine learning models don't have that, in much of a rich sense – you tend to have to more explicitly document things like how model is working, what the inputs what the outputs are, and explain them thoroughly before you ship something to a third party person. But you don't have that implicit way of reflecting through that API, and trying to get that information.*

The need for documentation that appeared in the interview material was not necessarily for an exhaustive description of a system. Instead, an overview of the responsibilities of individual parts of the system, as an addition to more detailed, conceptual explanations for the machine learning specific code, might have benefited the developers enough. Although the previous of these is arguably important also in classical software, the latter – explaining the concept of individual components – might not be equally necessary, as classical software can be written in a more self-documenting manner. While *documentation* precisely was discussed by a few participants only, the challenges caused by lack of it were visible in the described cases.

> *It (the model) was very fast, very performant, pretty good at what it, but basically the person had, like, written down the math and then translated the math directly into code. And then you had like variable names that were just like "l" and "xs" . . . . So, that was interesting. So I think, just code quality, in terms of variable names and comments, would have run a way in that point.*
>
> *Because when, for the first time, we had some production issues, it was really stressful, because it wasn't yet a super mature system and because it was quite a big system, and I came in there in the middle of things. So it was like, "oh no, how does everything work, how do I do this?!", and then towards the end it wasn't a problem any more. So everyone needs to actually understand the system well enough.*

Notably, according to the experiences of the previous interviewees, it seems essential that documenting is done on a regular basis during development, instead of writing all of it in the end of the project. This is similar to the verification and validation activities, which

are important to do regularly during the system development, as argued in the previous section. Especially, as the nature of machine learning development is experimental and the machine learning model and its inputs are likely to evolve during the system development, it is probable that a single, final documentation will not capture all the rationale that led into the last machine learning solution.

In addition to documenting the machine learning model concept, it seems that there are certain needs for documenting data as well. As discussed in the previous sections, the semantics of data have a massive effect on machine learning model performance. Moreover, these semantics are often tied to the uses of and expectations for data, which cannot be seen by looking at the data only. Consequently, it appears necessary to gather metadata of data collection, its other uses, and its origin, as this information is an important complement to the rest of the system documentation.

> *Collecting metadata is something that I think has an impact, because metadata is often missing from statistics and datasets, and people don't understand that the data has been collected for different purposes and by different means, and still these have a significant impact, even though you cannot see it from the system.*

As the previous interviewee noted, metadata is often missing from the datasets. In addition, there are no similar conventions for documenting data as there are for documenting systems, which might make the practice somewhat cumbersome. However, the literature does recommend this convention (e.g., Gebru et al. 2018), and it seems especially beneficial in mature systems, where the development team is likely to change in time.

5.2.5 Engineering practices

> *But people need to be aware of this. If they are not educated for this, they don't do that.*

Overall, these results suggest that assuring the quality of a modern machine learning system is generally not a problem in terms of the technical implementation of the means ensuring quality. Instead, the difficulties seem to arise from the lack of awareness, or lack of common understanding, for what are the important quality characteristics in machine learning systems. Participants with backgrounds from more established practices for building machine learning systems were typically enumerating rather clearly the quality characteristics that were relevant for them, and consequently also the conventions for quality assurance were in place. In contrast, while the participants with less organised approach did generally identify many central quality aspects, the lack of systematicity in thoughts for quality seemed to transfer into the practical work.

> *We have, in a way, described or "productised" our way of doing things, in order to do things consistently in a same way, so that it would assure the quality.*
>
> *But I'm not sure, if it was actually considered quality or not. So far there is no*

clear process like in software engineering, I think.

Most of the participants highlighted the importance of having traditional software skills also when building machine learning systems. This is not unexpected given that the quality of the technical implementation was one of the most discussed topics in the interviews. What makes it surprising, however, is that while the technical quality of machine learning systems has been gaining awareness since the seminal paper by Sculley et al. (2015), most of the interviewees thought that there still exist no generally accepted practices for building production machine learning systems. While different types of architectures and design patterns is an established field of research in classical software engineering (e.g., Alshuqayran, Ali, and Evans 2016; Washizaki, Ogata, et al. 2020), a similar effort in machine learning is only at its naissance (Washizaki, Uchida, et al. 2019). Consequently, machine learning systems were built as custom architecture software systems, necessitating deeper software development competence.

> *It is easy to do analysis, proof-of-concept implementations, but it takes more [software competence], if we want to run them in some bank system.*
>
> *Why can't we understand that data scientist is a software developer – a software developer writing a singular language typically – who understands variability, math, statistical analysis; all these things much better than a regular software developer.*

However, building machine learning systems was not considered *difficult*, once the technical skills were at an adequate level. Moreover, one of the interviewees noted that machine learning development is continuously being made more accessible by various tools and learning material. Indeed, nearly all of the discussed machine learning quality aspects in this work can be addressed to some extent with an existing tool (e.g., The Institute for Ethical AI & Machine Learning 2020). It is also true that studying machine learning does no longer necessitate a university education, as free learning material for different base knowledge levels exist plenty[8]. Consequently, this participant called for a wider understanding of designing and building digital systems from the machine learning developers, as well as for more personal responsibility in machine learning development.

> *Technically, we have so many learning material and tools right now; things are getting easier and easier, the next year is going to be a lot easier that it was this year, and so on. . . . So I think, we do need to merge the design, development, and machine learning, and start building those skills together. It's not complicated; right now, data science is not that complicated any more. . . . If it's becoming so easy, then we should be understanding how to step up and be better at what we're doing ethically outside of that. And also making judgement calls, like personal judgement calls. . . . We should inherently be questioning ourselves all of the time, in order to build a product.*

The suggestion of augmented responsibilities for machine learning engineers does not

[8] E.g., fast.ai (www.fast.ai) and Google AI Education (ai.google/education)

seem unreasonable: The specialists from other disciplines can hardly replace the data scientist's understanding of machine learning systems and their capabilities and limitations. On the other hand, rules and regulations are not likely to replace the personal understanding of creating robust and responsible systems. However, as demonstrated above, more systematised machine learning development practices seem to be necessary for building such robust and responsible systems efficiently.

> *So yeah, discipline, I think. That's probably the best word. Machine learning data scientists need to be better disciplined.*

5.3 Summary

Below, the findings presented in Sections 5.1 and 5.2 are summarised with respect to the research questions formulated in the introduction of this thesis.

RQ1: What quality issues are faced in the real-world machine learning development work? For the most part, the quality issues encountered in machine learning work in the industry seem to be from the categories that were described in Chapter 3. However, the exact problems within these categories do not reflect the prominent body of the literature. The most notable highlights from these differences are the question of service validity, which has gained almost no attention from the academia before, and the call for ethical responsibility from developers and organisations, which is an issue for the developers of today, in comparison to the artificial moral agents of tomorrow. Other interesting remarks from the findings include the need for a better distinction between the security of privacy and data privacy, and the apparent lack of long-term data quality considerations in machine learning research. The quality issues that are prominent in the industry are summarised in Table 5.3.

RQ2: Which types of quality assurance have the biggest impact on the quality of machine learning systems? Apart from verification and validation, the findings highlight four other important practices: Firstly, data and domain understanding seems to be a prerequisite for successful quality assurance, and especially the systematic collection of metadata was a new perspective in this practice. Another activity was designing of the quality; designing of the prevention of unwanted behaviour of the system in particular. A practice that has not been considered as quality assurance before was documentation, addressing the opaqueness of machine learning and data. Finally, although not a novel approach, the engineering practices seemed to have a significant impact on the overall quality of machine learning development. These quality assurance practices are described briefly in Table 5.4.

Table 5.3. *Summary of the quality issues faced in industry*

Service Quality 5.1.1	While the literature's focus on machine learning performance is often purely technical, in a real-world machine learning system the quality of the model's predictions is quite dependent on every other part of the system, drawing attention to the need for a more comprehensive understanding of the system and its environment. Moreover, the validity of the machine learning service, i.e., how well the model predicts the true business objective, is a notable problem for machine learning development in the industry, but this is rarely discussed in the literature. Whereas the model correctness is not necessarily the most important machine learning related quality attribute in the system, service validity has a larger impact on the perceived quality.
Technical Quality 5.1.2	Currently, the most prominent real-world issues related to the technical quality of a machine learning system concern the infrastructure and deployment setup of the system. Today, there are no shared practices for the development and operations of machine learning systems. In addition, technical debt, where the literature is focused, is still a notable problem for development, but the cause might be the limited resources available for machine learning development.
Data Quality 5.1.3	Although the data quality issues that are studied in academia (i.e., skew and stability) are also present in the industry, the literature discusses little the problems in data quality that appear after deployment, during operation of the system. In the light of the findings, the most notable runtime problems with data quality are broken data and semantic evolution of data (concept drift), the latter of these being also difficult to recognise. In addition, data quality management, especially in the long term, is not discussed in machine learning literature, but according to the participants' experiences, such efforts do have an impact for the development and runtime stability.
Security & Privacy 5.1.4	In security threats, the adaptiveness of the real-world adversaries was a considerable source of difficulties for implementing defences for a machine learning system, but the academia has not so far given much consideration for the adaptiveness of the defence mechanisms. However, the machine learning specific security risks are not currently relevant for many systems. Regarding privacy, the threat of deducing adversaries appears to be similarly low in comparison to traditional privacy leaks. On the other hand, data privacy risks were seen important for machine learning, but the lack of organisational support reflected on the developers. In general, privacy risks were not well known, and the ambiguous use of the word *privacy* might worsen the situation.
Interpretability 5.1.5	In industry, interpretability is most often used for debugging and improving usability of the system. Based on the material, the value of interpretability in real-world systems is not *intrinsic* but *instrumental*, but this difference is not made in the literature. In addition, there seems to be almost no critical examination over *when* and *how* interpretability should be applied, although these questions largely define the usability of the interpretations.
Fairness 5.1.6	In the real-world development work for fairness-aware machine learning, three problems emerged: Firstly, a policy is often needed for deciding over fairness interventions, because fairness is dependent on the context. Secondly, in reality, it can be difficult to evaluate fairness, because there might not be data of the protected groups. Thirdly, it is not intuitive for non-specialists that improving fairness comes with the cost of the machine learning service performance. However, fairness problems are perhaps not as common as could be thought from the public discussion.
Ethics 5.1.7	In comparison to the literature, the ethics that were discussed did not concern the artificial moral agents but the development responsibility and impact of the built system on the society. In this perspective, the personal sensitivity and subjectivity of ethical questions were seen problematic. In addition, the lack of organisational support seemed to discourage ethical and responsible development practices.

Table 5.4. *Summary of the quality assurance practices*

Data & application domain understanding 5.2.1	Based on the material, it seems that data and domain understanding has perhaps the most notable impact on achieving a high service validity and stable service. Although as an activity, *understanding* might seem vague, more deliberate actions, such as metadata collection, were recommended for reaching a sufficient level of knowledge of the data and domain. Notably, the understanding is also likely to increase during the system development and consequently cause iterativeness to both requirements and system development.
Design 5.2.2	Two categories where design should be applied emerged: Firstly, it seems necessary to design the desired quality attributes of a system in order to reach them. Secondly, in machine learning, the importance of designing the prevention and mitigation of the unwanted behaviour of a system appears highlighted. For example, as the security defences for machine learning are not generally effective, the design should account for the behaviour of the system under attack in order to prevent failures and disasters.
Verification & Validation 5.2.3	In comparison to classical software, testing activities were seen especially important for machine learning, as the correct system functioning and robustness cannot be logically deduced from the code. In addition, it was underlined that the testing should be replicated at system level, because also the software may contain defects that break the machine learning functionality. Apart from testing, also monitoring was seen essential. Other, non-programmatic validation activities were reviews – notably for the model concept –, and security and fairness validations that were conducted by parties other than the development team. Finally, it appears that the iterativeness of machine learning development also applies to the verification and validation of the system, and it was emphasised that these measures should be done continuously during the development.
Documentation 5.2.4	Because machine learning models are generally opaque black boxes, an explicit description of their functioning is the only efficient way of transferring knowledge. The need for documentation also applies to data, because there are implicit meanings and expectations that concern data, having a significant impact on the system. Similarly to verification and validation, also documenting should be done regularly, as the understanding of the system and data increase and evolve.
Engineering practices 5.2.5	The engineering practices seem to have a big impact on the system quality and development efficiency. Most importantly, being aware of the quality characteristics in machine learning systems, as well as having the knowledge of how to encounter the related problems, is a prerequisite for taking the characteristics into consideration in the development. Also software engineering skills were called for, as custom software and process architectures need to be created for machine learning systems. Finally, the machine learning developers' personal responsibility for developing robust and ethical systems was underlined.

6 DISCUSSION

The following sections will discuss the implications for the findings of this work. In addition, a brief look into the related work is provided, and the reliability and validity of this study are assessed.

6.1 Implications

During the recent years, the machine learning community has started to realise that as machine learning is increasingly applied in real-world systems, benchmarking the machine learning model correctness is no longer the main concern for those systems (e.g., Ma et al. 2018). Consequently, a growing body of machine learning literature has been focusing on the other central characteristics of machine learning models and systems, as well as on the evaluation and testing of those properties (e.g., J. M. Zhang et al. 2019). The academia has, for example, proposed algorithms for testing the prediction quality (J. M. Zhang et al. 2019), creating and testing for optimal attacks against the models (Carlini, Athalye, et al. 2019), and evaluating the level of unfairness in the model behaviour given a test data set (Mehrabi et al. 2019). While the course of the development seems positive from the perspective of the industry, the research is still somewhat disconnected from the real-world development work. Although all these research tracks could arguably benefit also the real-world systems, I suggest that currently the machine learning practitioners in the industry are mainly struggling with a different set of problems, causing the research advancements to be left unapplied.

Implications for machine learning development

It was mentioned in Section 5.1.2 that the software in machine learning systems did not present the participants any unsolvable issues in technical sense. In the light of the findings, this claim can be generalised to all of the reported quality perspectives: almost none of the problems described by the interviewees presented such algorithmic problems that the academia would not already have addressed (or that would not be under research, as in case of the security defences). Even most of the production time incidents that the interviewees described would have been rather easily prevented, had they been considered in advance

In fact, the origin of many of the described problems was in the lack of a systematic

consideration, assessment, and management of the machine learning characteristics. I argue that this is further caused by the absence of common engineering practices and methodology for machine learning development – without such an organised approach, the quality work becomes more difficult and time-consuming than is necessary, and quality is less likely to be considered pervasively. Overall, the results draw attention to the need for a stronger machine learning development discipline that comprises the awareness of the different characteristics of machine learning systems as well as the methodologies that support developing the machine learning systems efficiently and responsibly.

This claim has support also in the literature. For example, Sculley et al. has called for the software skills of machine learning developers already in 2015. Ma et al. (2018) pointed out the lack of generally accepted engineering practices in the context of secure and robust deep learning. Moreover, the ethical responsibility of individuals has gained attention also in literature in recent years. For example, Russell, Dewey, and Tegmark (2015) mention professional ethics of machine learning development as one of the future research priorities. B. Smith and Shum (2018, pp. 8–9) propose a “Hippocratic Oath” for developers to accentuate their role in creating responsible machine learning systems.

One could argue that the existing software methodologies could be applied in machine learning development as well. While the software methodologies can indeed support building machine learning systems, they lack in a few important areas: Firstly, they do not consider the machine learning specific characteristics of the systems and consequently leave the responsibility of obtaining the awareness of the potential issues and how to address them onto the machine learning developer, as has been until to date. Secondly, the software engineering methodologies do not take into account the indeterminism and uncertainty caused by the data-driven logic or the invisibility caused by the complexity and opaqueness of machine learning, and therefore, the software methodologies need to be adapted when used for machine learning development. Thirdly, the software engineering methodologies are not adequately prepared for the experimental nature of machine learning development, although the iterative software engineering methodologies, such as agile, do facilitate this to some extent. However, the experimental nature of data science can result even in revising the business requirements, as the understanding of the data and the system capabilities increase (Chapman et al. 2000). For these reasons, this work proposes establishing methodologies specifically for machine learning development.

Implications for education and organisations

This discussion also has an important relation to the university education of machine learning. For example, currently in the field of software engineering, the engineering practices and methodologies – not only the theory of programming – are taught at university level. Similarly, also the real-world appliance of data science and machine learning ought to be considered in the education. Although it cannot be expected that the education covers all of the machine learning characteristics thoroughly, the findings here

suggest that a more comprehensive look into them is needed to have the machine learning developers better prepared for the industry work. While it is arguably necessary to deliver the knowledge of the statistical theory behind machine learning in the education, this is not a sufficiently comprehensive proficiency in machine learning when building other than research systems aiming at new benchmarks. As demonstrated in this thesis, real-world machine learning development work encounters issues that are not (directly) related to the correctness of the machine learning model, but resolving these issues can be difficult and the means remain insufficient, if the machine learning developers do not have prior competence in addressing them.

However, in the light of the findings, it seems necessary that the facilitation for quality comes not only from education and developers but also from the companies investing into machine learning. Indeed, it appears necessary that the organisations understand the cost of building robust, production-quality systems. If an organisation values high quality, the developers should be given the necessary resources to build the systems accordingly. This facilitation appears all the more important given that – based on the interview material – it is currently difficult to motivate the organisations to invest into even the lowest necessary level of quality. Although not explicitly discussed in Chapter 5, the most pointed criticism from the participants towards the organisational side was the absence of concrete support for building and improving quality.

> *It would be really great if companies treat machine learning projects as they treat other software products. It has to have a lifespan and a process, and you have to go through iterations. It's not just proof of concept and this garbage. And you are given this really small short time to prove something to them, and if you didn't really meet their expectations, they kill it. It's not like that, it should not be like that.*

The need for organisational support does not limit to resources only. As mentioned in Section 5.1.7, the ethics of an organisation also have an impact on the ethical decision-making of individuals. However, in the light of the interviews, this phenomenon seems to generalise to the other discussed characteristics as well: the organisation culture appeared to have a significant impact on how sensitive the participants were to the different characteristics of quality. In this perspective, most of the interviewees reflected the mindset of the companies that they were or had been working for. Notably, it is likely that companies that value quality have also been investing into it, and therefore the organisational values and culture cannot be fully separated from the willingness to invest into quality.

In conclusion, it seems necessary that the machine learning developers, organisations, and universities co-operate in creating a common set of practices and methodologies for developing reliable, fair, and safe machine learning systems. The machine learning developers are encouraged to document and share the best found development practices and methodologies, but organisational support is needed to facilitate this. In addition, providing this increasing knowledge for the universities would allow widening the scope

of the machine learning education and potentially discovering new directions for research over machine learning engineering methodologies.

6.2 Related work

Based on the thorough review over related literature, this work is one of the first to look into the quality and quality assurance of machine learning systems at a general level, and one of the few to bring up the differences between academia and industry in the development work. To demonstrate how this work and its results contribute to the literature, the following gives an overview of the previous works that have overlapped this study as of certain parts.

J. M. Zhang et al. (2019) conducted an extensive survey on machine learning testing. In the work, the authors create a comprehensive taxonomy of machine learning system "testing properties" – properties that in this work are called "quality perspectives" or "machine learning characteristics". The testing properties enumerated in the paper are correctness, overfitting, robustness and security, efficiency, fairness, and interpretability. As the testing properties were drawn from literature over machine learning testing, the origin of those properties are different from this work. In comparison, this thesis gives minimal emphasis on machine learning prediction quality and related properties, as the objective of this work was to discuss characteristics exactly other than those. In addition, the authors were discussing testing only, and consequently the other quality assurance activities, data quality, or technical quality of the system were not considered. The work is undeniably an excellent overview of the efforts towards verifying the expected behaviour of machine learning systems, but it only provides theoretical information over a subset of possible quality assurance practices and consequently does not specifically support designing and building quality systems, which this work aims at addressing.

Ma et al. (2018) created an engineering process model for secure deep learning (Secure Deep Learning Engineering Life Cycle, SDLC). In the model, the software engineering process was augmented with the necessary concepts to facilitate creating a robust and secure deep learning system. While the authors did mention other quality characteristics, such as interpretability, in the work, they did not consider those in the created framework. The authors also discussed quality assurance in the model, but no systematic perspective into it was taken, and the proposed quality assurance activities largely – although not completely – focused on verification and validation of different machine learning system components. Consequently, while demonstrating the importance of considering security of machine learning systems during the complete lifespan of such systems, the work benefits little in development of systems where security is not a main concern, while according to the findings of this study, such systems exist plenty. However, the authors reach the same conclusion as this thesis in that the software methodologies and quality assurance approaches are not suitable for machine learning systems as such, and more consideration to machine learning specific issues must be given.

Baier, Jöhren, and Seebacher (2019) studied the challenges faced by industry practitioners in deployment and operation of machine learning systems. Similarly to this work, the authors conducted semi-structured interviews to machine learning practitioners to answer the research question. The authors categorised the findings of the literature review into *pre-deployment*, *deployment*, and *non-technical* challenges, and this classification was used both for structuring the interviews and the findings. However, as opposed to this work, the challenges that the authors identified from the literature were mostly related to the machine learning model, technical implementation, or data, making the initial scope of the study narrow. Moreover, although the non-technical perspectives were discussed to some extent, the technical emphasis of the work reflected onto the findings: the authors conclude that more research is needed especially around the challenges faced during deployment, such as infrastructure setup and model concept drift, and that better tooling for communicating the machine learning results would be valuable. In comparison, this work reveals mostly non-technical issues in machine learning development, as the literature review and interview structure were constructed around machine learning characteristics rather than operations. Therefore, despite the very similar research questions, the findings between these papers are quite different.

As already mentioned earlier in this thesis, there have been works related to "quality" or "quality assurance" of machine learning systems specifically. However, in these these papers, "quality" has generally meant the quality of the software (e.g., Villalobos, Ferrer, and Alba 2018), while "quality assurance" has denoted testing (e.g., Nakajima 2018). There also exist papers that discuss quality assurance in a broader meaning of the expression, but the applicability of these works seems low because of the lack of sufficient descriptions of the concepts (e.g., Hamada et al. 2020; Nishi et al. 2018).

6.3 Reliability and validity

It is often said that reliability of a qualitative research is difficult to assess, as the researcher is an "instrument" in the study, and it is impossible to remove the researcher's subjective experiences and knowledge from the empiria and results (e.g., Eskola and Suoranta 1998, ch. 5). However, suggestions for important considerations exist. Here, the reliability and validity of the used methodology and the results are assessed as proposed by Tuomi and Sarajärvi (2018, ch. 6). In addition, the work is discussed in the light of the perspectives described by Charmaz (2006, pp. 181–183).

Chapter 4 describes the methodology and data used in this study as objectively and as explicitly as possible, while protecting the anonymity of the informants. The chapter also justifies the rationale for the design of the empirical part of this study, i.e., the decision to collect data with interviews, selection criteria for the interviewees, and the chosen type and course of the interviews.

It was assumed that by using snowball sampling to recruit the interviewees, the most relevant persons for this study would be found. The interviewees also had experience

from companies of different sizes and from different continents. As the participants' experiences started to saturate as of majority of the interview themes, the number of participants seems reasonable. However, the demographic and cultural similarity of the interviewees is likely to have affected the discussed issues. Moreover, as mentioned in Chapter 5, fewer of the participants had experiences regarding security and fairness of machine learning systems, and it is possible that more prominent insights from these characteristics can be found. For example, the industry practitioners' views into fairness-aware machine learning development have been studied before (Holstein et al. 2019).

All of the interviews were recorded and transcribed, and as mentioned in Chapter 4, there were no issues in the recording quality, and the transcriptions were made verbatim first. In addition, there is no reason to suspect the authenticity of the experiences of the participants. Consequently, the material that was analysed was technically of high quality. However, the interview protocol was quite simple, and majority of the course of the interviews depended on the experiences of the interviewees. As the researcher was inexperienced in interviewing, this may have affected the depth, or information content, of the collected material (e.g., Hirsjärvi and Hurme 2008, ch. 6).

The analysis was reserved the necessary time to avoid rushed conceptions. The process of constant comparison was carried as pedantically as possible, but the lack of experience of the researcher might have affected the depth of the results (e.g., Hirsjärvi and Hurme 2008). After the analysis, all interview material was compared to the final concepts to verify the results. The assumption prior the interviews was that the results would highlight the need of breaching the gap between the academia and industry for improving the quality of work in both. The findings, however, indicate that problem is more complicated than just a lack of communication, so it seems safe to claim that the presumptions of the researcher did not reflect on the results of the study. Finally, the validity of the analysis was tried to ensure by allowing the participants to review the findings before publishing the work.

Next, this work is assessed according to the four categories proposed by Charmaz (2006, pp. 181–183): credibility, resonance, originality, and usefulness. Firstly, the *credibility* of this study has been tried to ensure by reporting the findings with excerpts from the interviews, in order to provide rationale for the conclusions of the researcher and to allow the reader to independently evaluate the results. In addition, by having completed the constant comparison carefully during the analysis, there should be strong evidence for the presented claims. By allowing the participants to view the results before publishing the work, the credibility of this study is hopefully further enhanced.

The *resonance* of the results seems to be supported by the identified, wider image of machine learning development and difficulties in it: based on the findings, it has been possible to draw links between different phenomena and institutions and suggest actions on the grounds of them. However, the generalisability of the results of this work are limited in a few ways. Firstly, due to the nature of the discussed systems, the quality problems appearing in other types of data, such as text, images, and sound, got less

attention. It is likely that the versatility of the issues increases along with the richness of the data because of the higher information content of such data. Secondly, the high variance in the efforts for quality assurance might threaten the validity of the results, although some support from prior work (e.g., Ma et al. 2018) can be found. Thirdly, due to the demographics of the participants, the results may reflect primarily the experiences of developers working at consultancies, where the systems that are developed are often young. In comparison, systems developed in product companies might introduce issues that are specific for mature systems exactly. For example, Sculley et al. (2015) have reported technical issues faced in the development of such, mature products.

Finally, although the categories of quality and quality assurance in this work were for the most part not unheard-of, the *originality* of this study perhaps lies in the effort of discussing “quality” and “quality assurance” of machine learning systems in the wider meanings of the expressions. Consequently, this work intends to prove its *usefulness* in the future research and practical work at machine learning by encouraging more rigorous quality considerations and development of practices for ensuring quality in machine learning work.

7 CONCLUSION

The research over machine learning has considered the quality of machine learning systems for a long time from a very narrow perspective, leading into the nonchalant application of the technology, even in high-risk domains (Ma et al. 2018). The purpose of this study was to increase the knowledge around quality and its assurance in the context of machine learning, and consequently help the practitioners in implementing robust and safe machine learning systems. By using semi-structured interviews, the study aimed at clarifying the important quality characteristics in machine learning by investigating the problems appearing in the practical work in the domain. The second goal of this work was to identify the types of quality assurance approaches that have the biggest positive impact on the quality of machine learning systems.

Seven important machine learning system characteristics were ascertained from the interviews: service quality, technical quality, data quality, security and privacy, interpretability, fairness, and ethics. Although there exist literature in all of these domains, the findings reveal that there are several issues in practical machine learning work that differ or are missing from the main body of the research: For example, data quality presents problems both during system development and maintenance, and discussion over technical quality was largely on processes and operations instead of software. Moreover, the ambiguity of the term *privacy* seemed to reflect onto the practical considerations related to it. The most notable perspectives in machine learning development that are not discussed widely in the literature were the difficulty of achieving a high machine learning service validity and the call for higher ethical responsibility from both developers and organisations.

Apart from the quality characteristics, five meaningful practices for quality assurance could be identified from the material: data and domain understanding, design, verification and validation, documentation, and engineering practices. Of these, especially the engineering practices seemed to have a significant impact on the quality of the machine learning development work in general. However, these practices were not deliberately performed by all of the interviewees, and the variance in the efforts for ensuring quality was high. Given this heterogeneity, these results are limited to rather abstract suggestions for the practical work.

Finally, an important observation from the findings is that currently the problems in machine learning development work are mostly methodological and not theoretical or technical. In fact, the findings have revealed an important gap in the field of machine learning research and education: there is nearly no effort towards the development and establish-

ing of a methodological discipline of machine learning engineering. Consequently, the quality work today can be overly resource demanding, and the necessary activities to ensure quality are learned by production incidents at worst. Moreover, the role of organisations in machine learning work has been long neglected: both resources and cultural support from organisations are needed for facilitating the development robust and safe real-world systems. In conclusion, it seems imperative that the machine learning developers, organisations, and academia collaborate in development, validation, and education of a systematic methodology for machine learning engineering.

REFERENCES

Aïvodji, U., Arai, H., Fortineau, O., Gambs, S., Hara, S., and Tapp, A. (2019). Fairwashing: the risk of rationalization. arXiv: 1901.09749 [cs.LG].

Alshuqayran, N., Ali, N., and Evans, R. (2016). A Systematic Mapping Study in Microservice Architecture. *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)*, 44–51.

Andress, J. (2014). *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*. Syngress Media Incorporated. ISBN: 9780128007440.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (May 23, 2016). Machine Bias. *ProPublica*. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (visited on 06/22/2020).

Bacchelli, A. and Bird, C. (2013). Expectations, outcomes, and challenges of modern code review. IEEE Press, 712–721.

Baeza-Yates, R. (2018). Bias on the Web. eng. *Association for Computing Machinery. Communications of the ACM* 61.6, 54–61. ISSN: 00010782. URL: http://search.proquest.com/docview/2070923291/.

Baier, L., Jöhren, F., and Seebacher, S. (2019). Challenges In The Deployment And Operation Of Machine Learning In Practice. Proceedings of the 27th European Conference on Information Systems (ECIS). ISBN: 978-1-7336325-0-8. URL: https://aisel.aisnet.org/ecis2019_rp/163.

Barocas, S., Hardt, M., and Narayanan, A. (2019). *Fairness and Machine Learning*. http://www.fairmlbook.org. fairmlbook.org. (Visited on 06/22/2020).

Batini, C. and Scannapieca, M. (2006). *Data Quality: Concepts, Methodologies and Techniques*. Springer-Verlag. ISBN: 3540331727. DOI: 10.1007/3-540-33173-5.

Bordeleau, F., Cabot, J., Dingel, J., Rabil, B. S., and Renaud, P. (2019). Towards modeling framework for DevOps: requirements derived from industry use case. *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*. Springer, 139–151.

Borkar, S. (2005). Designing reliable systems from unreliable components: the challenges of transistor variability and degradation. *Ieee Micro* 25.6, 10–16.

Breck, E., Cai, S., Nielsen, E., Salib, M., and Sculley, D. (2016). What's your ML Test Score? A rubric for ML production systems. URL: https://research.google/pubs/pub45742/.

Breck, E., Polyzotis, N., Roy, S., Whang, S., and Zinkevich, M. (2019). Data Validation for Machine Learning. Proceedings of the 2nd SysML Conference. URL: https://mlsys.org/Conferences/2019/doc/2019/167.pdf.

Buneman, P., Khanna, S., and Wang-Chiew, T. (2001). Why and where: A characterization of data provenance. *International conference on database theory*. Springer, 316–330.

Buolamwini, J. and Gebru, T. (23–24 Feb 2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by S. A. Friedler and C. Wilson. Vol. 81. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, 77–91. URL: http://proceedings.mlr.press/v81/buolamwini18a.html.

California Legislative Information (2020). *California Consumer Privacy Act of 2018*. URL: http://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?lawCode=CIV&division=3.&title=1.81.5.&part=4. (visited on 01/18/2020).

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. (2019). On Evaluating Adversarial Robustness. arXiv: 1902.06705v2 [cs.LG].

Carlini, N. and Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. Proceedings - IEEE Symposium on Security and Privacy, 39–57. DOI: 10.1109/SP.2017.49.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics 8.8, 832. ISSN: 2079-9292. DOI: 10.3390/electronics8080832.

Cervantes, J.-A., López, S., Rodríguez, L.-F., Cervantes, S., Cervantes, F., and Ramos, F. (2020). Artificial Moral Agents: A Survey of the Current Status. *Science and Engineering Ethics*, 1–32.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Tech. rep. SPSS Inc. CRISPWP-0800.

Charmaz, K. (2006). *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*. eng. Introducing qualitative methods. London: Sage. ISBN: 0-7619-7352-4.

Chen, X., Li, B., and Vorobeychik, Y. (2016). Evaluation of defensive methods for DNNs against multiple adversarial evasion models. URL: https://openreview.net/pdf?id=ByToKu9ll.

Chouldechova, A., Benavides-Prado, D., Fialko, O., and Vaithianathan, R. (23–24 Feb 2018). A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by S. A. Friedler and C. Wilson. Vol. 81. Proceedings of Machine Learning Research. New York, NY, USA: PMLR, 134–148. URL: http://proceedings.mlr.press/v81/chouldechova18a.html.

Cihon, P. (2019). Standards for AI Governance: International Standards to Enable Global Coordination in AI Research & Development. University of Oxford. URL: https://www.fhi.ox.ac.uk/wp-content/uploads/Standards_-FHI-Technical-Report.pdf.

Collinge, G., Lupu, E., and Muñoz-González, L. (June 2019). Defending against Poisoning Attacks in Online Learning Settings. European Symposium on Artificial Neural Networks (Bruges, Belgium).

Dalvi, N., Domingos, P., Mausam, Sanghai, S., and Verma, D. (2004). Adversarial classification. Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 99–108. DOI: 10.1145/1014052.1014066.

DeBrusk, C. (2018). The Risk of Machine-Learning Bias (and How to Prevent It). MIT Sloan Blogs.

Dinga, R., Penninx, B. W., Veltman, D. J., Schmaal, L., and Marquand, A. F. (2019). Beyond accuracy: measures for assessing machine learning models, pitfalls and guidelines. *bioRxiv*. DOI: 10.1101/743138.

Doshi-Velez, F. and Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. arXiv: 1702.08608 [stat.ML].

Dwork, C. (2006). *Differential Privacy*. Vol. 4052. Springer Berlin Heidelberg, 1–12. ISBN: 0302-9743. DOI: 10.1007/11787006_1.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2011). Fairness Through Awareness. arXiv: 1104.3913v2 [cs.CC].

Ehrlinger, L., Rusz, E., and Wöß, W. (2019). A Survey of Data Quality Measurement and Monitoring Tools. arXiv: 1907.08138 [cs.DB].

Elango, B., Paul, K., Kundu, S. K., and Paudel, S. K. (2010). Organizational ethics, individual ethics, and ethical intentions in international decision-making. *Journal of Business Ethics* 97.4, 543–561.

Erich, F. M. A., Amrit, C., and Daneva, M. (2017). A qualitative study of DevOps usage in practice. Journal of software: Evolution and Process 29.6, e1885–n/a. ISSN: 2047-7481. DOI: 10.1002/smr.1885.

Eriksson, P. and Kovalainen, A. (2008). *Qualitative methods in business research. E-book*. SAGE. ISBN: 0857028049.

Eskola, J. and Suoranta, J. (1998). *Johdatus laadulliseen tutkimukseen. E-book*. Vastapaino. ISBN: 9517685041.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust Physical-World Attacks on Deep Learning Visual Classification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1625–1634. DOI: 10.1109/CVPR.2018.00175.

Federal Highway Administration (2004). *Speed limit 80 sign*. Public domain. URL: https://commons.wikimedia.org/wiki/Speed_limit_road_signs#/media/File:Speed_limit_80_sign.svg (visited on 07/05/2020).

Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. (Aug. 2014). Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, 17–32. ISBN: 978-1-931971-15-7. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/fredrikson_matthew.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*. DOI: 10.1145/3287560.3287589. URL: http://dx.doi.org/10.1145/3287560.3287589.

Galin, D. (2004). *Software Quality Assurance. From Theory To Implementation*. Pearson Education Limited. ISBN: 9783766331182.

Garvin, D. A. (1984). What Does "Product Quality" Really Mean?: Sloan Management Review (pre-1986) 26.1, 25–43. ISSN: 0019-848X.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., III, H. D., and Crawford, K. (2018). Datasheets for Datasets. arXiv: 1803.09010v7 [cs.DB].

Gillham, B. (2000). *Case study research methods*. Continuum. ISBN: 0826447961.

Glaser, B. and Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Observations (Chicago, Ill.) Aldine. ISBN: 9780202302607.

Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., and Sculley, D. (2017). Google vizier: A service for black-box optimization. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1487–1495.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. arXiv: 1412.6572 [stat.ML].

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2019). A Survey of Deep Learning Techniques for Autonomous Driving. arXiv: 1910.07738 [cs.LG].

Gudivada, V., Apon, A., and Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software* 10.1, 1–20.

Guest, G., Bunce, A., and Johnson, L. (2006). How Many Interviews Are Enough?: An Experiment with Data Saturation and Variability. Field Methods 18.1, 59–82. ISSN: 1525-822X. DOI: 10.1177/1525822X05279903.

Hacker, P. (2018). Teaching fairness to artificial intelligence: Existing and novel strategies against algorithmic discrimination under EU law. *Common Market Law Review* 55, 1143–1186. URL: https://ssrn.com/abstract=3164973.

Hamada, K., Ishikawa, F., Masuda, S., Matsuya, M., Myojin, T., Nishi, Y., Ogawa, H., Toku, T., Tokumoto, S., Tsuchiya, K., et al. (2020). Guidelines for Quality Assurance of Machine Learning-based Artificial Intelligence. The 32nd International Conference on Software Engineering & Knowledge Engineering. DOI: 10.18293/SEKE2020-094. URL: http://ksiresearch.org/seke/seke20paper/paper094.pdf.

Hanif, M. A., Khalid, F., Putra, R. V. W., Rehman, S., and Shafique, M. (2018). Robust Machine Learning Systems: Reliability and Security for Deep Neural Networks. 2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS), 257–260. DOI: 10.1109/IOLTS.2018.8474192.

Hirsjärvi, S. and Hurme, H. (2008). *Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö. E-book*. Finnish. Helsinki: Gaudeamus Helsinki University Press. ISBN: 9524958864;9789524958868;

Holstein, K., Wortman Vaughan, J., Daumé, H., Dudik, M., and Wallach, H. (2019). Improving Fairness in Machine Learning Systems. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. DOI: 10.1145/3290605.3300830. URL: http://dx.doi.org/10.1145/3290605.3300830.

Hong, S., Zhou, Y., Shang, J., Xiao, C., and Sun, J. (July 2020). Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. *Computers in Biology and Medicine* 122, 103801. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2020.103801. URL: http://dx.doi.org/10.1016/j.compbiomed.2020.103801.

IEEE Std 730-2014 (2014). IEEE Standard for Software Quality Assurance Processes (Revision of IEEE Std 730-2002), 1–138. DOI: 10.1109/IEEESTD.2014.6835311.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial Examples Are Not Bugs, They Are Features. arXiv: 1905.02175 [stat.ML].

ISO (2020). *ISO/IEC JTC 1/SC 42 – Artificial intelligence*. URL: https://www.iso.org/committee/6794475.html (visited on 07/05/2020).

ISO/IEC 25010 (2011). System and software quality models – Systems and software Quality Requirements and Evaluation (SQuaRE).

ISO/IEC 25012 (2008). Data quality model – Systems and software Quality Requirements and Evaluation (SQuaRE).

ISO/IEC/IEEE 24765 (2017). ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary, 1–541. DOI: 10.1109/IEEESTD.2017.8016712.

Japkowicz, N. (2006). Why question machine learning evaluation methods?: AAAI workshop on evaluation methods for machine learning, 6–11.

Ji, Z., Lipton, Z. C., and Elkan, C. (2014). Differential Privacy and Machine Learning: a Survey and Review. arXiv: 1412.7584v1 [cs.LG].

Kaiser, K. (2009). Protecting Respondent Confidentiality in Qualitative Research. Qualitative health research 19.11, 1632–1641. ISSN: 1049-7323. DOI: 10.1177/1049732309350879.

Kearns, M. and Li, M. (1993). Learning in the Presence of Malicious Errors. SIAM Journal on Computing 22.4, 807–837. ISSN: 0097-5397. DOI: 10.1137/0222052.

Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent Trade-Offs in the Fair Determination of Risk Scores. arXiv: 1609.05807 [cs.LG].

Koh, P. W., Steinhardt, J., and Liang, P. (2018). Stronger Data Poisoning Attacks Break Data Sanitization Defenses. arXiv: 1811.00741v1 [stat.ML].

Laporte, C. Y. and April, A. (2018). *Software Quality Assurance*. 1st ed. Wiley. ISBN: 1118501829. DOI: 10.1002/9781119312451.

Lee, D. (2016). *Google self-driving car hits a bus*. URL: https://www.bbc.com/news/technology-35692845 (visited on 01/02/2020).

Leite, L., Rocha, C., Kon, F., Milojicic, D., and Meirelles, P. (2019). A Survey of DevOps Concepts and Challenges. ACM Computing Surveys (CSUR) 52.6, 1–35. ISSN: 0360-0300. DOI: 10.1145/3359981.

Lewis, J. (Jan. 29, 2018). Fitness-Tracker App Exposes Security Flaw at Taiwan's Missile Command Center. *The Daily Beast*. URL: https://www.thedailybeast.com/strava-fitness-tracker-app-exposes-taiwans-missile-command-center (visited on 02/15/2020).

Li, B., Vorobeychik, Y., and Chen, X. (2016). A General Retraining Framework for Scalable Adversarial Classification. arXiv: 1604.02606v2 [cs.GT].

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2019). Federated learning: Challenges, methods, and future directions. arXiv: 1908.07873v1 [cs.LG].

Lin, Y.-C., Liu, M.-Y., Sun, M., and Huang, J.-B. (2017). Detecting Adversarial Attacks on Neural Network Policies with Visual Foresight. arXiv: 1710.00814 [cs.CV].

Lundberg, S. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. arXiv: 1705.07874 [cs.AI].

Ma, L., Juefei-Xu, F., Xue, M., Hu, Q., Chen, S., Li, B., Liu, Y., Zhao, J., Yin, J., and See, S. (2018). Secure Deep Learning Engineering: A Software Quality Assurance Perspective. arXiv: 1810.04538 [cs.SE].

Marijan, D., Gotlieb, A., and Kumar Ahuja, M. (2019). Challenges of Testing Machine Learning Based Systems. English. IEEE, 101–102.

Martínez-Plumed, F., Ferri, C., Nieves, D., and Hernández-Orallo, J. (2019). Fairness and Missing Values. arXiv: 1905.12728 [cs.LG].

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2019). A Survey on Bias and Fairness in Machine Learning. arXiv: 1908.09635v2 [cs.LG].

Miltenburg, E. van (2016). Stereotyping and Bias in the Flickr30K Dataset. arXiv: 1605.06083 [cs.CL].

Molnar, C. (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/.

Morstatter, F., Shu, K., Wang, S., and Liu, H. (2017). Cross-Platform Emoji Interpretation: Analysis, a Solution, and Applications. arXiv: 1709.04969v1 [cs.CL].

Nakajima, S. (2018). [Invited] Quality Assurance of Machine Learning Software. 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), 601–604. DOI: 10.1109/GCCE.2018.8574766.

Nickerson, R. C., Varshney, U., and Muntermann, J. (2013). A method for taxonomy development and its application in information systems. European Journal of Information Systems 22, 336–359. ISSN: 0960-085X. DOI: 10.1057/ejis.2012.26.

Nishi, Y., Masuda, S., Ogawa, H., and Uetsuki, K. (2018). A Test Architecture for Machine Learning Product. 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 273–278.

O.Nyumba, T., Wilson, K., Derrick, C. J., and Mukherjee, N. (2018). The use of focus group discussion methodology: Insights from two decades of application in conservation. Methods in Ecology and Evolution 9.1, 20–32. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12860.

Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science* 366.6464, 447–453. URL: https://escholarship.org/content/qt6h92v832/qt6h92v832.pdf.

ONNX Project Contributors (2019). *ONNX – Open Neural Network Exchange*. URL: https://onnx.ai (visited on 11/05/2019).

Oxford University Press (2019). *Definition of Privacy*. URL: https://www.lexico.com/en/definition/privacy (visited on 11/08/2019).

Ozbayoglu, A. M., Gudelek, M. U., and Sezer, O. B. (2020). Deep Learning for Financial Applications : A Survey. arXiv: 2002.05786 [q-fin.ST].

Packer, M. J. (2010). *The science of qualitative research*. Cambridge University Press. ISBN: 0521148812. DOI: 10.1017/CBO9780511779947.

Papernot, N., McDaniel, P., Sinha, A., and Wellman, M. (2016). Towards the Science of Security and Privacy in Machine Learning. arXiv: 1611.03814 [cs.CR].

Paudice, A., Muñoz-González, L., Gyorgy, A., and Lupu, E. C. (2018). Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. arXiv: 1802.03041v1 [stat.ML].

Pérez-Peña, R. and Rosenberg, M. (Jan. 29, 2018). Strava Fitness App Can Reveal Military Sites, Analysts Say. *The New York Times*. URL: https://www.nytimes.com/2018/01/29/world/middleeast/strava-heat-map.html (visited on 02/15/2020).

Publications Office of the European Union (May 4, 2016). General Data Protection Regulation. *Official Journal of the European Union* L 119, 1–88.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv: 1602.04938 [cs.LG].

Al-Rubaie, M. and Chang, J. M. (2018). Privacy Preserving Machine Learning: Threats and Solutions. arXiv: 1804.11238 [cs.CR].

Russell, S., Dewey, D., and Tegmark, M. (2015). Research priorities for robust and beneficial artificial intelligence. AI Magazine 36.4, 105–114. ISSN: 0738-4602. DOI: 10.1609/aimag.v36i4.2577.

Saucedo, A. (2019). *The state of MLOps in 2019*. URL: https://www.youtube.com/watch?v=Ynb6X0KZKxY (visited on 12/31/2019).

Schminke, M. (2001). Considering the business in business ethics: An exploratory study of the influence of organizational size and structure on individual ethical predispositions. *Journal of Business Ethics* 30.4, 375–390.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. (2018). Towards the first adversarially robust neural network model on MNIST. arXiv: 1805.09190v3 [cs.CV].

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden Technical Debt in Machine Learning Systems. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Montreal, Canada: MIT Press, 2503–2511. URL: https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf.

Sherin, S., Khan, M. U., and Iqbal, M. Z. (2019). A Systematic Mapping Study on Testing of Machine Learning Programs. arXiv: 1907.09427v1 [cs.LG].

Shostack, A. (2014). *Threat modeling: designing for security*. John Wiley and Sons. ISBN: 9781118822692.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2019). Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. arXiv: 1911.02508 [cs.LG].

Smith, B. and Shum, H. (2018). *The future computed: artificial Intelligence and its role in society*. Microsoft Corporation. URL: https://blogs.microsoft.com/wp-content/uploads/2018/02/The-Future-Computed_2.8.18.pdf (visited on 12/30/2019).

Smith, C. S. (2018). *Alexa and Siri Can Hear This Hidden Command. You Can't.* URL: https://www.nytimes.com/2018/05/10/technology/alexa-siri-hidden-command-audio-attacks.html (visited on 07/03/2020).

Sokol, K. and Flach, P. (Jan. 2020). Explainability fact sheets. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. DOI: 10.1145/3351095.3372870. URL: http://dx.doi.org/10.1145/3351095.3372870.

Sze, V., Chen, Y.-H., Emer, J., Suleiman, A., and Zhang, Z. (Apr. 2017). Hardware for machine learning: Challenges and opportunities. *2017 IEEE Custom Integrated Circuits Conference (CICC)*. DOI: 10.1109/cicc.2017.7993626. URL: http://dx.doi.org/10.1109/CICC.2017.7993626.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. arXiv: 1312.6199 [cs.CV].

Taeihagh, A. and Lim, H. S. M. (2018). Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. DOI: 10.1080/01441647.2018.1494640.

TensorFlow Data Validation: Checking and analyzing your data (2019). URL: https://www.tensorflow.org/tfx/guide/tfdv (visited on 11/06/2019).

The Guardian (2018). *Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian*. URL: https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe (visited on 07/03/2020).

The Institute for Ethical AI & Machine Learning (2020). *Awesome production machine learning*. URL: https://github.com/EthicalML/awesome-production-machine-learning (visited on 07/01/2020).

The Khronos Group (2019). *Neural Network Exchange Format (NNEF)*. URL: https://www.khronos.org/nnef/ (visited on 11/05/2019).

The National Archives (2020). *Equality Act 2010*. URL: http://www.legislation.gov.uk/ukpga/2010/15/section/4 (visited on 06/22/2020).

Tian, J. (2001). Quality assurance alternatives and techniques: A defect-based survey and analysis. Software Quality Professional 3.3, 6–18.

Tolmeijer, S., Kneer, M., Sarasua, C., Christen, M., and Bernstein, A. (2020). Implementations in Machine Ethics: A Survey. arXiv: 2001.07573 [cs.AI].

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., and Fox, E. A. (2020). Natural Language Processing Advancements By Deep Learning: A Survey. arXiv: 2003.01200 [cs.CL].

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing Machine Learning Models via Prediction APIs. arXiv: 1609.02943v2 [cs.CR].

Tuomi, J. and Sarajärvi, A. (2018). *Laadullinen tutkimus ja sisällönanalyysi: Uudistettu laitos. E-book*. Tammi. ISBN: 9789520400118.

Verma, S. and Rubin, J. (2018). Fairness definitions explained. Proceedings of the International Workshop on software fairness. ACM, 1–7. DOI: 10.1145/3194770.3194776.

Victor, D. (Mar. 24, 2016). Microsoft Created a Twitter Bot to Learn From Users. It Quickly Became a Racist Jerk. *The New York Times*. URL: https://www.nytimes.com/2016/03/25/technology/microsoft-created-a-twitter-bot-to-learn-from-users-it-quickly-became-a-racist-jerk.html (visited on 10/03/2020).

Villalobos, I., Ferrer, J., and Alba, E. (2018). Measuring the quality of machine learning and optimization frameworks. Vol. 11160. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 128–139. DOI: 10.1007/978-3-030-00374-6_13.

Wagner, S. (2013). *Software Product Quality Control*. Springer. ISBN: 9783642385704. DOI: 10.1007/978-3-642-38571-1.

Wagstaff, K. (2012). Machine Learning that Matters. arXiv: 1206.4656 [cs.LG].

Wang, J., Crawl, D., Purawat, S., Nguyen, M., and Altintas, I. (2015). Big data provenance: Challenges, state of the art and opportunities. 2015 IEEE International Conference on Big Data (Big Data), 2509–2516. DOI: 10.1109/BigData.2015.7364047.

Wang, R. Y. and Strong, D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of Management Information Systems 12.4, 5–33. ISSN: 0742-1222. DOI: 10.1080/07421222.1996.11518099.

Wang, Y., Jha, S., and Chaudhuri, K. (2019). An Investigation of Data Poisoning Defenses for Online Learning. arXiv: 1905.12121v2 [cs.LG].

Washizaki, H., Ogata, S., Hazeyama, A., Okubo, T., Fernandez, E. B., and Yoshioka, N. (2020). Landscape of Architecture and Design Patterns for IoT Systems. *IEEE Internet of Things Journal*, 1–1.

Washizaki, H., Uchida, H., Khomh, F., and Gueheneuc, Y.-G. (Dec. 2019). Studying Software Engineering Patterns for Designing Machine Learning Systems. *2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP)*. DOI: 10.1109/iwesep49350.2019.00017. URL: http://dx.doi.org/10.1109/IWESEP49350.2019.00017.

Widmer, G. (1996). Recognition and exploitation of contextual clues via incremental meta-learning (Extended version). *The 13th International Conference on Machine Learning (ML-96), Morgan Kaufmann, San Francisco*.

Wilson, R. J., Zhang, C. Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., and Gipson, B. (2019). Differentially Private SQL with Bounded User Contribution. arXiv: 1909.01917v3 [cs.CR].

Wiyatno, R. R., Xu, A., Dia, O., and de Berker, A. (2019). Adversarial Examples in Modern Machine Learning: A Review. arXiv: 1911.05268v2 [cs.LG].

Yu, H., Shen, Z., Miao, C., Leung, C., Lesser, V. R., and Yang, Q. (2018). Building Ethics into Artificial Intelligence. arXiv: 1812.02953v1 [cs.AI].

Yu, L., Liu, L., Pu, C., Gursoy, M. E., and Truex, S. (2019). Differentially private model publishing for deep learning. arXiv: 1904.02200v5 [cs.CR].

Zhang, C., Tan, K. C., Li, H., and Hong, G. S. (2019). A Cost-Sensitive Deep Belief Network for Imbalanced Classification. IEEE Transactions on Neural Networks and Learning Systems 30.1, 109–122. DOI: 10.1109/TNNLS.2018.2832648.

Zhang, J. M., Harman, M., Ma, L., and Liu, Y. (2019). Machine Learning Testing: Survey, Landscapes and Horizons. arXiv: 1906.10742v2 [cs.LG].

Zhang, S., Yao, L., Sun, A., and Tay, Y. (Feb. 2019). Deep Learning Based Recommender System. *ACM Computing Surveys* 52.1, 1–38. ISSN: 1557-7341. DOI: 10.1145/3285029. URL: http://dx.doi.org/10.1145/3285029.

A INTERVIEW PROTOCOL

1. Can you tell me about yourself? Your background, such as education and work.
2. How has *quality* been part of your education or work?
3. How would you define quality, in general?
4. How would you define quality in the context of machine learning? (Can you tell me an example?) *Discuss as long as the interviewee stays within the scope of the work.*
5. *Introduce the quality axis of the theme matrix (Table A.1).*
6. *Continue the discussion. For example:* Do you have experiences of some of these characteristics? / How do these characteristics show in your work? / You already discussed about *X* and *Y* – do you have experiences related to explainability?
7. *If necessary, introduce the quality assurance axis of the interview theme matrix. Ask experiences from quality assurance categories that did not appear so far.*
8. Do you have something else, related to quality or quality work, in mind?

Table A.1. *Interview theme matrix*

	Constructive QA	Analytical QA	Failure control
Technical quality			
Data quality & Ethical bias			
Security & Privacy			
Explainability			

B INTERVIEW CONSENT FORM

INTERVIEW CONSENT FORM

TOWARDS QUALITY ASSURANCE IN MACHINE LEARNING SYSTEMS

Thank you for agreeing to be interviewed as part of the above master's thesis project. I kindly ask you to read this consent form to ensure that you understand the purpose of your involvement and agree to the conditions of your participation.

Participation in this study is voluntary, and you can withdraw at any time or refuse to answer any question without consequences of any kind. The information you provide for this study will be treated confidentially, and all identities will remain anonymous in any report of the results. The information you provide will be used for this study only. The final version of the thesis will be public by law.

At your consent, the interview will be audio-recorded, and the recording will be transcribed so that all identifying information has been removed. The recording, transcript, and all other material will be destroyed after the thesis has been approved. You are entitled to access the information you have provided at any time while it is in storage.

I give the permission to audio-record the interview: yes / no

Signature(s) of participant(s)	Signature of researcher
Date	Date