Empirical Software Engineering (2022) 27: 66
https://doi.org/10.1007/s10664-021-10105-2

Systematic literature review on software quality for AI-based software

Bahar Gezici[1] · **Ayça Kolukısa Tarhan**[1]

Accepted: 9 December 2021 / Published online: 17 March 2022

Abstract
There is a widespread demand for Artificial Intelligence (AI) software, specifically Machine Learning (ML). It is getting increasingly popular and being adopted in various applications we use daily. AI-based software quality is different from traditional software quality because it generally addresses distinct and more complex kinds of problems. With the fast advance of AI technologies and related techniques, how to build high-quality AI-based software becomes a very prominent subject. This paper aims at investigating the state of the art on software quality (SQ) for AI-based systems and identifying quality attributes, applied models, challenges, and practices that are reported in the literature. We carried out a systematic literature review (SLR) from 1988 to 2020 to (i) analyze and understand related primary studies and (ii) synthesize limitations and open challenges to drive future research. Our study provides a road map for researchers to understand quality challenges, attributes, and practices in the context of software quality for AI-based software better. From the empirical evidence that we have gathered by this SLR, we suggest future work on this topic be structured under three categories which are Definition/Specification, Design/Evaluation, and Process/Socio-technical.

Keywords Artificial intelligence · Machine learning · Software quality · Quality attributes · Quality metrics · Measurement · Product quality model

1 Introduction

AI-based software consistently displays distinctive characteristics since this type of software is built with training data in an inductive manner. The components obtained are naturally

Communicated by: Paolo Tonella

✉ Bahar Gezici
bahargezici@cs.hacettepe.edu.tr

Ayça Kolukısa Tarhan
atarhan@hacettepe.edu.tr

[1] Institute of Science, Computer Engineering Department, Hacettepe University, Ankara, Turkey

imperfect, i.e., having no possibility of 100% accuracy, they are black-box, and since the learned behavior is derived from training data, it is not understandable as traditional software in which the system behavior depends on the logical design built by engineers. Since existing approaches/guidelines do not work for AI-based software, one survey pointed out that more than 40% of engineers feel the difficulties of achieving quality for AI-based software (Ishikawa and Yoshioka 2019). So, it is crucial to supply clear guidance for providing insight and handling the challenges for ensuring high-quality AI software.

The necessity for paying attention to the quality of AI-based systems was investigated more than 30 years ago (Rushby 1988). In his study, Rushby suggested the initiation of a powerful program that should pay attention to both the adaptation of existing techniques from traditional software engineering and the development of the new approaches/techniques that consider the nature and characteristics of AI-based software. There are frameworks/quality standards that have been searched for traditional software and have worked for years for such software systems and, there is also a need for similar frameworks/standards as guides for engineers when developing AI-based software. The increased interest in software engineering for AI-based software has resulted in a growing number of publications in this area over the last four years (Nascimento et al. 2020). Although there are existing secondary studies on the topic of quality for big data systems (Zhang et al. 2020; Rahman and Reza 2020), surveys on quality of ML software (Masuda et al. 2018) or data quality for big data software (Lakshen et al. 2016; Taleb et al. 2018); to the best of our knowledge, there is no systematic literature review on software product quality for AI-based software specifically. This has motivated us to conduct a systematic literature review (SLR) in a broader way to provide an up-to-date, holistic, and comprehensive view of the state-of-the-art of software quality for AI-based software research, especially from the product quality perspective, to analyze existing quality challenges, models, attributes, metrics or techniques that are concerning quality for AI-based software. In this way, we have gathered related primary studies published between 1988 and 2020 with the goal of (i) understanding and summarizing scientific literature and (ii) searching for the open and addressed challenges necessary to drive future research.

In this SLR, we proposed to answer the following research goal: *"How is quality defined or investigated for the AI-based software?"*. The SLR is divided into specific research questions to answer this research goal, as described in Section 4.1. The results we have identified could help practitioners understand challenges and approaches about how to develop AI-based software by considering software quality. Our contributions in this paper are described below:

1- The study makes a bottom-up analysis and contribution. It focuses on metrics/measurements at the bottom and relates them to quality characteristics and concepts at the top.

2- The study synthesizes the scientific literature in the context of quality for AI-based software and, investigates the gap between the current challenges and their solutions, in terms of measures and quality characteristics,to suggest areas for further research.

The remainder of this article is organized as follows: Background and related work are presented in Section 2. The methodology followed in the study is described in Section 4. Section 5 reports the study results concerning the research questions. In Section 6, we present the discussion regarding the results obtained and the validity threads. Finally, Section 7 provides overall conclusions.

2 Background

In this part, we give the general terms of Artificial Intelligence and Machine Learning software. Hopgood (2005) defined Artificial Intelligence (AI) as the science of mimicking human mental faculties on a computer. Hence, AI is used to describe machines that mimic "cognitive" functions such as "learning" and "problem-solving" that people associate with other human minds (Russel and Norvig 2009). Bosch et al. (2021) claimed that the prominent area of AI engineering is an extension of software engineering, and it contains new technologies and processes required while developing such systems. Therefore, researchers have studied the difficulties, processes, and approaches related to developing AI software systems.

There are several studies that focus on the taxonomy or classification of AI software systems (Forward and Lethbridge 2008; Geske et al. 2021; Samoili et al. 2020). Although this is far beyond the scope of our purposes in this SLR study, we summarize several closely related terms under AI in the following. AI has evolved into new waves as Machine Learning (ML), and more specifically Deep Learning (DL) (Deng 2018), in the last decades. AI is actually a broad concept involving machines making decisions based on ML algorithms. As a branch of AI, ML makes progressively better decisions or predictions by using data (Aggarwal et al. 2019). DL is a subset of ML, which requires little or no guidance of humans to improve its function. DL is the study of artificial neural networks and machine learning algorithms, and is recently used for many software applications (Ongsulee 2017). Software systems which include one or more AI-components such as autonomous driving, image recognition, etc. are called AI-based software systems (Martínez-Fernández et al. 2021).

Within the scope of this SLR study, the term "AI-based software" is used to refer to the type of software targeted basically for AI or ML, including sub-fields as deep learning and computer vision, wherever reported in the primary studies. In addition, while the term 'model' is used to express an aspect of software design in Software Engineering, the same term is used to refer to a sub-component of an AI-based software. In order to avoid ambiguities that may arise from using this term for "quality model" and "AI model", we used the term 'model (component)' to indicate AI model and used 'model' to indicate SQ model.

Software quality is defined starting with the requirements stage as a basis of objective and systematic evaluation. Requirements in AI-based systems are vaguer in comparison to the ones in traditional software systems, because such systems are much more data-driven. Different data causes different requirements to be defined and when the data changes, the requirements also change (Alamin and Uddin 2021; Wan et al. 2019). More specifically, randomness in ML methods (e.g., randomness in data, initialization of weights, and optimizations) causes uncertainty and complicates software testing (Wan et al. 2019). There are no expected results predefined for software systems using such ML methods, and it may not be possible to generate reliable test oracles (Braiek and Khomh 2020; Wan et al. 2019). In addition, when developing traditional software, there is usually a clear development method to be followed step-by-step. Consequently, the problem definition and solving processes have more uncertainty in AI-based software systems compared to the traditional ones.

Nevertheless, there is an increasing demand for AI-based software in industry, and a high rate of attempts has been done to understand the processes followed and difficulties faced by researchers and professionals (Byrne 2017; Rahman and Reza 2020). Like any traditional software, AI-based software also requires quality assurance. There are quality models and/or standards that include different quality attributes for ensuring the intended quality of a software system. For instance, ISO/IEC 25010 (ISO/IEC 2011) defines a product quality model for software.

However, there is not enough empirical evidence that these software quality models can be applied directly for specifying and evaluating the quality of AI-based software due to its different nature. For traditional software systems, there are various quality metrics that are used to measure quality attributes. For example, Gezici et al. [23] analyzed 61 versions of the three open-source software (OSS), and measured the Portability quality attribute with Depth of Inheritance Tree (DIT) metric, and measured the Understandability quality attribute with Coupling Between Object Classes (CBO), Commented Lines of Code (CLOC), Lack of Cohesion in Methods (LCOM) metrics which are widely-used object-oriented design metrics. In another study, Paulson et al. [53] focused on measuring software complexity by using McCabe's Cyclomatic Complexity metric over three open-source and three commercial software projects. However, for AI-based software systems, these metrics cannot be directly used since the performance of the model (component) depends on the learning/training data, and it is a challenging task to use these metrics or transfer them to measure the quality of AI-based components (Poth et al. 2020). Therefore, it is necessary to find new quality models or metrics specific to AI-based software due to the different nature of such software systems (Nakamichi et al. 2020). In spite of this necessity, there is a lack of knowledge and awareness about the quality of AI-based software. According to a study of a systematic literature review (Nascimento et al. 2020) about software engineering (SE) for artificial intelligence, it has been found that there was no comprehensive study in the field of SE for AI-based systems until 2016 and in 2019, publications had a high growth peak, i.e., there were 21 studies published this year.

One area with the lowest number of studies on software engineering for AI-based software was "Software Quality" (only 4% of the studies, which are Aggarwal et al. 2019; Ishikawa and Yoshioka 2019) as shown in Fig. 1.

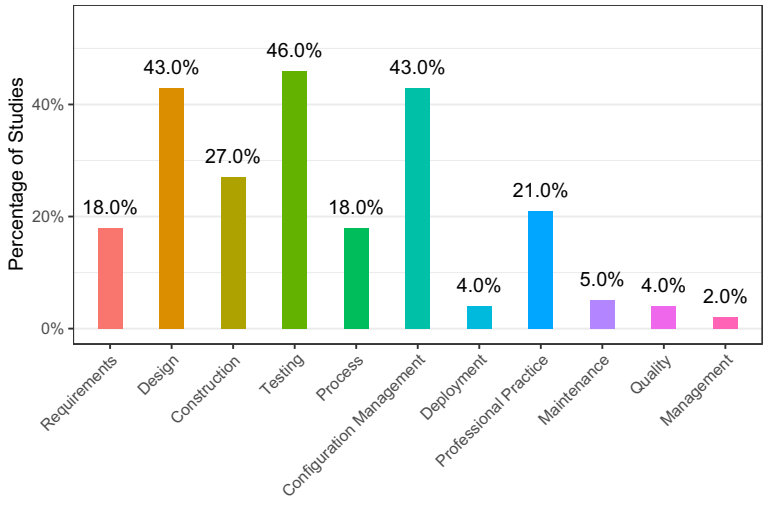

Fig. 1 SWEBOK knowledge areas associated with primary studies on SE for AI software (Nascimento et al. 2020)

3 Related Work

In this section, we focus and list the secondary studies which have proposals on quality for AI-based software.

To emphasize our contribution to this study, we list the existing secondary studies related to our context within the different scope as shown in Table 1. When we have started this SLR study (in Fall 2020), there were already literature reviews on the quality for AI/ML software. However, their scope, objective, and types of outcomes are different from our purposes in this SLR.

Borg et al. (2018) surveyed the state of the art in the context of verification and validation of ML systems, more specifically Deep Neural Network (DNN). They reviewed 64 primary studies within the periods of 2002-2007 and 2013-2016. They focused on challenges in V&V for safety-critical automotive systems. At the end of the study, the authors reported gaps between ISO 26262 standard and ML practice and a need for further empirical studies to encourage safety approaches, including "safety cage architectures" and "simulated test cases." Differently from our study, while we focus on AI-based software systems and quality metrics at the bottom, the study by Borg et al. focused only on reliability, robustness, and safety challenges for automotive systems.

A survey conducted by Masuda et al. (2018) reported on the quality of ML applications. They recommended the need for software engineering research in this context. Through a survey, they collected problems with ML applications and explored software engineering approaches and software testing research areas to solve these problems. They focused on problems that have not been solved in the literature yet: "How to verify the answer that ML-as-a-services (MLaaS) returns to unknown data?", "How to verify insufficient or biased data because problems because data determine the logic of ML applications?", "How to verify the quality of end-to-end systems?", "How to notify users confidence of answer correctness from the system?". The authors firstly summarized the top seven software quality techniques/methods for ML applications such as "deep learning," "fault localization," "MLaaS," "search-based," and "model checking." Then, they found papers that have correspondence between difficulties and these methods/techniques. However, they concluded that each paper focused on some difficulties, and there remained difficulties that have not been covered. While this survey study examines machine learning software, as similar to our study, it examines software types in a narrower scope (e.g., software testing research area). As different from us, the study focused on software testing problems only for the quality of ML software.

Braiek and Khomh (2020) reviewed the current testing practices for ML-based systems. They searched on Engineering village, Google Scholar, and Microsoft Academic Search databases and found 37 related primary studies within the years of 2007-2019. To ensure the reliability of ML programs, the authors searched for and identified challenges when testing such software systems, presented solutions recommended by studies in literature, and made suggestions in the light of their findings. This study aims to help practitioners to improve the quality of ML programs by learning testing techniques for such systems. Unlike their study, in our SLR study, we focus on different challenge categories, and testing is just one of those categories.

Zhang et al. (2020) conducted a comprehensive survey study to analyze the testing trends, challenges, and solutions for ML software systems. They collected 138 articles within the years of 2007-2019 and focused on correctness, robustness, efficiency, privacy, and fairness as testing properties. They defined and showed existing ML testing approaches and

Table 1 List of existing literature and our proposed contribution in this study

Study	Domain	Contribution
Borg et al. (2018)	Deep Learning	They reviewed the state of the art in safety-critical ML systems in the context of verification and validation. They focused on challenges for safety evaluations for such software systems.
Masuda et al. (2018)	Machine Learning	They proposed a survey of software quality for discovering techniques to evaluate and improve the software quality for ML software. They considered the quality issues in ML applications and aimed to find solutions with traditional SE approaches and software testing study fields.
Braiek and Khomh (2020)	ML-based Systems	They focused on the challenges and solutions for testing approaches.
Zhang et al. (2020)	Machine Learning	They investigated the literature in the context of "testing properties" such as fairness, robustness, etc., "testing components" such as data, framework, etc., "testing workflow" such as test generation, test evaluation, and "application scenario" such as autonomous driving, etc.
Riccio et al. (2020)	ML-based System	They conducted a systematic mapping study in the context of testing techniques. They reviewed 70 papers, identified problems, solutions to the problems, and investigated different testing approaches in their study.
Nascimento et al. (2020)	AI/ML	They focused on the challenges in software engineering practices within AI/ML development process. They focused on different software engineering practices such as project management, testing, architecture design, education, software quality, operation support, etc.
Lwakatare et al. (2020)	Machine Learning	They aimed to conduct a systematic literature review to survey the literature for the development and maintenance of ML systems. They identified some challenges and solutions in the industrial context. They focused on more research and synthesized challenges in SE by considering "adaptability," "scalability," "privacy," and "safety" quality attributes..

Table 1 (continued)

Study	Domain	Contribution
Tsintzira et al. (2020)	Machine Learning	They conducted a systematic literature review that is limited by five high-quality SE journals. They investigated the literature for the current status, opportunities, and challenges for technical debt management in machine learning.
Our study (2021)	AI-based software	We conduct a Systematic Literature Review (SLR) by analyzing the measures proposed/used at the bottom to the quality characteristics and dimensions at the top. We propose to find and synthesize the primary studies in the context of quality for AI-based software in the literature, and identify the gap between the current challenges and their solutions, in terms of measures, quality characteristics, and quality models, with a purpose to recommend areas for further research.

workflows. Similar to our study, they investigated the quality attributes such as correctness, robustness, etc., as specific to testing of ML software systems. However, their context and domain are different from ours since we aim to study product quality not only for software testing, and our scope covers AI-based software that includes AI/ML software, deep learning, etc.

Riccio et al. (2020) performed a systematic mapping study by reviewing 70 papers in the context of functional testing for ML systems (MLS). While we focus on AI-based software, the authors of this study focused on MLS and specifically on MLS testing, challenges, and solutions. While the research questions they defined are specific to testing, the ones in our study are about product quality for AI-based software. Accordingly, their contributions and outcomes are quite different from ours.

Nascimento et al. (2020) conducted a systematic literature review of the studies in the time period between 1990-2019. They extracted and analyzed 57 primary studies in the context of software engineering for AI. They focused on the challenges and categorized them under five dimensions which are Test, Software Quality, Data Management, Model Development, and Project Management. In future work, the authors stated that the focus will be on certain research topics such as maintenance, or software quality and management, which are the least studied subjects of research in the literature and will be the main topics to be focused on in the future.

Lwakatare et al. (2020) performed a systematic literature review of 72 papers on the development and maintenance of ML-based software systems, their challenges, and solutions. This study does not specifically focus on the product quality, and the type of software analyzed by it is ML-based software, while in our study it is AI-based software.

Another systematic literature review conducted by Tsintzira et al. (2020) focuses on technical debt management (TDM) for ML software with 90 primary studies. Since the

authors of this study investigated current challenges and solutions in the context of TDM and ML, their outcomes, scope, and purpose are very different from ours.

Consequently, the shortcomings mentioned below have motivated us to conduct a broader and more functional systematic literature review.

- From an analysis of the related literature, it can be concluded that there is still a big gap in addressing the quality problems of AI-based software systems.
- None of the former studies have explored the relationship between challenges, deficiencies, and practices of SE with the quality for AI-based software, or specified which contexts or domains have been mostly concerned, or which deficiencies have been met.
- Although the prior studies have made an understanding of particular cases and context of AI-based software systems, they have not provided a clearer picture of AI-based software quality and related challenges in several experimental, academic or industrial contexts.

Therefore, we claim that ours is the first research that provides a comprehensive review of scientific literature in the context of software quality for AI-based software systems, by making an analysis of the primary studies for eliciting the quality measures proposed/used at the bottom and relating them to the quality characteristics and dimensions at the top as well as a synthesis on the quality characteristics and the approaches adopted for their assurance together with the challenges and proposed solutions.

4 Methodology

By following Kitchenham's guideline (Kitchenham and Charters 2007), we performed a systematic literature review to provide a holistic picture of the state of the art in quality for AI-based software.

Systematic literature studies can include reviews and synthesizing existing work in a manner that is fair for sustaining researchers in realizing the current state of a research field in software engineering (Wohlin 2014). In this SLR study, the results are expected to provide a more general sight of the gaps and evidence for software quality in the context of AI-based software systems.

According to Peterson's guideline, main stages such as study selection, string-search, data extraction strategies are processed in the following sections (Petersen et al. 2015).

4.1 Research Goal and Questions

This SLR is aimed to answer the following research goal: "How is quality defined or investigated for the AI-based software?". To answer this research goal, the SLR was aimed to answer research questions and was divided into sub-questions, as described in Fig. 2.

In order to address the Software Development Lifecycle (SDLC) for the quality of AI-based software systems, we adopt the software development practices of machine learning-driven systems in Wan et al. (2019). While the authors found significant differences between ML systems and non-ML systems in eight dimensions (1-Requirements, 2-Design, 3-Testing and Quality, 4-Process and Management, 5-Skill Variety, 6-Job Complexity, and Problem Solving 7-Task Identity, 8-Interaction), we focus on the first three dimensions with an additional "Background and State-of-the-Art Understanding" dimension following

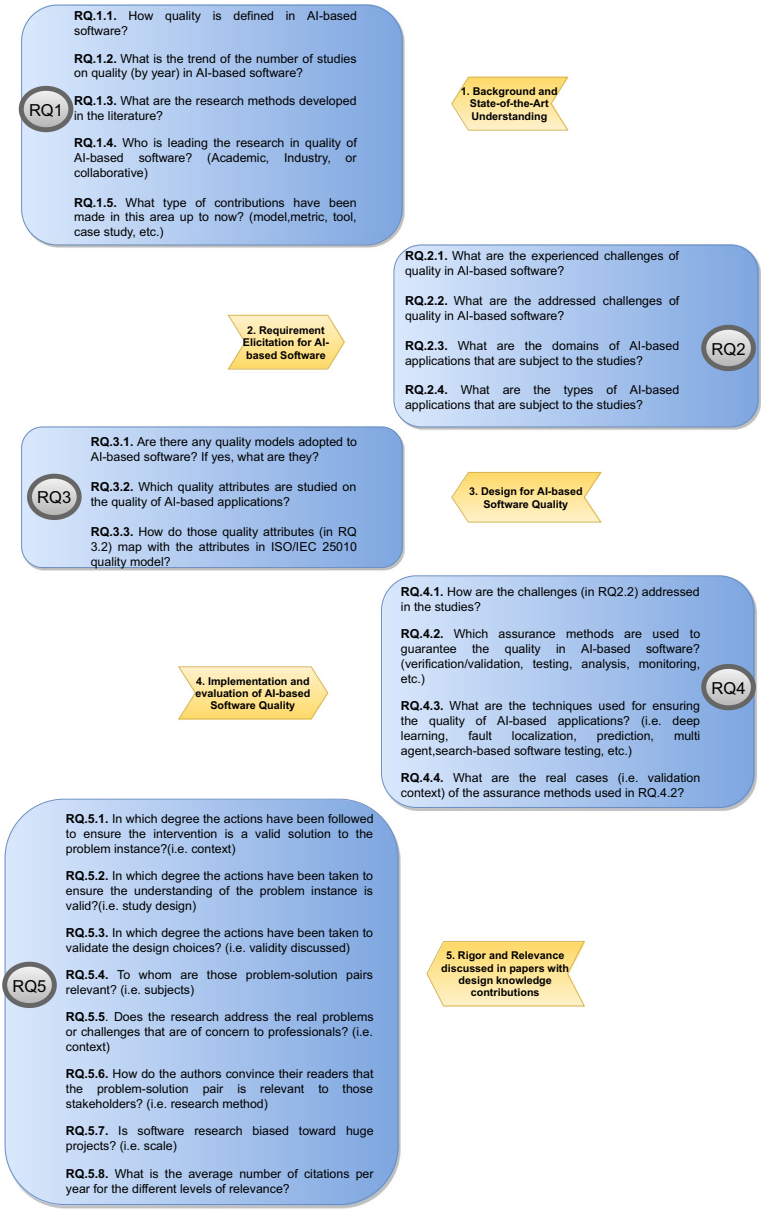

Fig. 2 Research Questions analyzed in this SLR

four dimensions and also, as specific to our study, we add Rigor and Relevance as a fifth dimension in order to discuss design knowledge contributions of the primary papers.

In the dimension of "Background and State-of-the-Art Understanding", in order to develop an AI-based software by considering quality, firstly, we have to know what is "software quality". So, in RQ 1.1, we propose to learn "How quality is defined in AI-based

software". After obtaining the main idea behind the AI-based software quality, we propose to analyze the state of the art in this field by defining the remaining sub-RQs of RQ 1.

After we understand the background and state of the art, we intend to learn the system development requirements by considering possible challenges, solutions, or the domains of applications (RQ 2.1, RQ 2.2, and RQ 2.3) that are subject to the studies in "Requirements elicitation for AI-based software quality" dimension.

After the second dimension with the requirements stage for AI-based software quality, understanding how to design for AI-based software quality is another important dimension for practitioners. In this stage, we focus on quality models, quality attributes, and more specifically quality metrics. Hence, we derived sub-questions under RQ 3.

After the design phase is defined as the third dimension, in the "Implementation and evaluation of AI-based software quality" dimension, when a system is being implemented, we focus on identifying different system attributes such as methods and techniques that help for the qualitative and quantitative evaluations on these systems. We also need to focus and identify how the challenges are addressed by considering approaches followed in the literature. Hence, we derived four sub-RQs under RQ 4 in order to obtain this information.

We also included the "Rigor and Relevance discussed in papers with design knowledge contributions" as the fifth dimension in order to see the maturity of the field. Therefore, there are eight sub-questions defined under this dimension regarding RQ 5.

After describing our SLR protocol, we now focus on sourcing relevant studies from the literature. The procedure followed in this SLR is visualized as shown in Fig. 3.

4.2 Initial Search with Databases

In this systematic literature review, we used string-based search and snowballing techniques to source and select the relevant studies in the literature.

Before starting the research, to get insights and be closer to the context of quality for AI-based software, we made some exploratory readings from the literature, which is defined

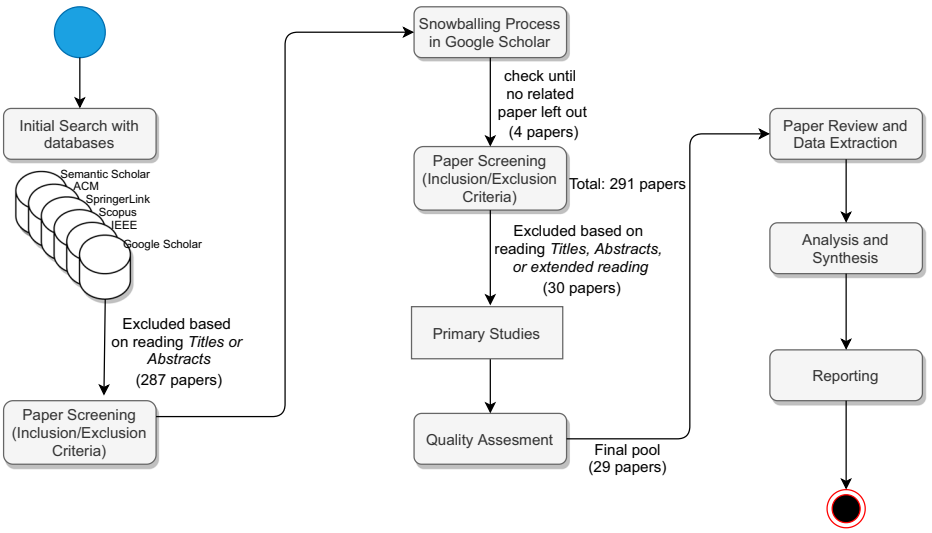

Fig. 3 Systematic literature review procedure applied in this study

as free-style search, and performed a selective review process that covered reviewing some papers from these readings to identify control papers to check the suitability of search strings. For string-based search, we carried out both free-style search and selective review processes. Before constructing search strings, we performed a free search to find articles in the context of quality for AI-based software. We read the full texts of the papers and identified some control papers.

In string-based search, firstly, we defined search keys by including synonyms, abbreviations as supplementary terms to increase the search results. To advance the qualification of the search process, search strings were slightly modified. We checked the search keys via making a selective review on the control papers list. For identifying control papers, please see pages (10-11) in the following de Almeida Biolchini et al. (2007). We checked ten articles (in Table 2) that we obtained with our free-style work and can be accepted as a base, to see if we can obtain them with these keywords.

After ensuring correctness and availability of search terms, the final search string was composed of the terms that represent **Population**- Artificial Intelligence, Machine Learning, AI Software, ML Software and **Intervention**- Software Engineering, Software Quality of AI/ML software by following the guidelines of Kitchenham (2004), and we finalized our search terms as below.

Search String-1:

```
(''artificial intelligence software'' OR ''AI software'' OR ''AI-based
```

Table 2 List of control papers

No	Article name	Reference
1	Quality Measures and Assurance for AI Software	Rushby (1988)
2	Software Quality in Artificial Intelligence Systems	Vinayagasundaram and Srivatsa (2007)
3	Quality Assurance of Machine Learning Software	Nakajima (2018)
4	Adapting SQuaRE for Quality Assessment of Artificial Intelligence Systems	Kuwajima and Ishikawa (2019)
5	Guidelines for Quality Assurance of Machine Learning-based Artificial Intelligence	Hamada et al. (2020)
6	Software Quality for AI Where we are now?	Lenarduzzi et al. (2021)
7	Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation	Nakamichi et al. (2020)
8	Towards Guidelines for Assessing Qualities of Machine Learning Systems	Siebert et al. (2020)
9	Priority Quality Attributes for Engineering AI-enabled Systems	Pons and Ozkaya (2019)
10	Quality Assurance for Machine Learning–an approach to function and system safeguarding	Poth et al. (2020)

```
software'' OR ''machine learning software'' OR ''ML software'')
AND (''quality assessment'' OR ''quality evaluation'' OR ''quality
assurance'' OR ''software testing'' OR ''software verification'' OR
''software validation'') AND (metric OR measure OR measurement)
```

Search String-2:

```
(''artificial intelligence software'' OR ''AI software'' OR ''AI-based
software'' OR ''machine learning software'' OR ''ML software'')
AND (''quality model'' OR ''quality attribute'' OR ''quality
characteristic'' OR ''nonfunctional requirement'' OR ''quality
requirement'') AND (metric OR measure OR measurement)
```

Specific to this study, the population constitutes academic studies that discuss software, and intervention being the artificial intelligence or machine learning and any knowledge area of software engineering as described in SWEBOK (Bourque et al. 2004). By considering this roadmap, we constructed two different search strings and used them on the date of Nov. 1, 2020, on six different databases, namely Google Scholar, IEEE, Scopus, SpringerLink, ACM Digital Library, and Semantic Scholar. We chose these six databases because they are the most comprehensive scientific databases that involve computer science papers (Zhang et al. 2020). Since Google Scholar covers several significant databases such as Springer, Elsevier, ACM, IEEE, we chose the Google Scholar database for searching related articles. In addition, we pick the Scopus database because it lists papers from several popular and well-known publishers such as ACM, IEEE, Springer, and Elsevier (Nascimento et al. 2020). We also included the ACM, Springer, and IEEE databases to be sure that we don't miss any publications in these databases, even if Scopus indexes them. ACM also indexes some papers from Springer Link (Nascimento et al. 2020). In addition, we used the Semantic Scholar for searching scientific literature because it is a free and AI-based tool that supports researchers clearly recognize related studies (Hannousse 2021). By using each search string, we searched in all databases separately and listed the number of selected papers in Table 3. For each paper found in each database, we applied the inclusion/exclusion criteria. We first read the titles and then abstracts, or if the necessary introduction and/or conclusion parts of the studies. After applying the selection procedure and removing duplicates, in total, we obtained 287 studies for the second stage of the SLR for extended reading.

4.3 Snowballing Process in Google Scholar

In this step, we again performed a search to assure that there is not any related study left out from the initial search. The snowballing technique is recommended and used in place of database searches in systematic literature reviews (Wohlin 2014), which includes backward and forward snowballing iteration processes. In the snowballing stage, the first author performed the backward snowballing process in which the reference list in a primary study is analyzed to find related papers to be included in. From this part, we found four more related papers, and the initial set with the relevant studies reached 291.

4.4 Inclusion and Exclusion Criteria

To decide whether a relevant study must be in the final pool or not, it needs to satisfy both inclusion and exclusion criteria as defined below. If all inclusion criteria are satisfied and

Table 3 Search strings and quantity of retrievals in databases

Results Quantity	Search String-1	Search String-2
Google Scholar	3160	303
IEEE	49	9
Scopus	7	2
SpringerLink	175	50
ACM	65	8
Semantic Scholar	17000	7450
Selected Papers	265	22

none of the exclusion criteria is satisfied, the study is selected; otherwise, it is discarded.

Inclusion Criteria (IC)

- IC1. The selected papers must focus on quality for AI-based software.
- IC2. The selected papers must be in the software engineering research area.
- IC3. One or more research questions constructed in this SLR study must be directly answered.
- IC4. The selected papers must be written in English.
- IC5. The literature must consist of journal papers or papers published as a part of conference or workshop proceedings or book chapters.

Exclusion Criteria (EC)

- EC1. The study is related to AI-based software but not related to the quality of AI-based software. Although data quality can affect resulting product quality, we should note that our purpose is to study the product quality for AI-based software, not the quality of data as input to such software systems. Data quality is a related and self-contained topic that requires further attention, as also addressed by ISO/IEC 25012 (25012:2008 2008).
- EC2. The study does not discuss the quality of AI-based software or quality factors, models, attributes of these software systems.
- EC3. Duplicate research papers. Some papers are found in different databases more than once and, they may be published in both journals or conferences, or workshops. We selected journal papers.
- EC4. Ph.D. dissertations, secondary studies (SLR/SM), posters, editorials, magazines.

In the paper screening stage, which is about applying inclusion/exclusion criteria to the articles in the pool, the two authors considered the inclusion and exclusion criteria together as complementary. The paper screening process was carried out in the following steps. Firstly, the authors discussed and decided on inclusion/exclusion criteria by reading Titles or Abstract of the papers, and selected the related papers in the context of quality for AI-based software separately. Within the initial pool, the authors got 287 papers selected in this step. Next, by applying the snowballing process, the corresponding author got four more papers, and a total of 291 papers were included in the pool. Then, both authors applied the same inclusion/exclusion criteria to these 291 papers by reading Titles, Abstract, and

the full texts in more detail and voted each paper separately. During this process, detailed discussions were held in case of disagreements when finalizing paper selection.

There is a certain level of difficulty when analyzing all the literature. Developing research questions and search strategies were conducted by the two authors together. After making decisions according to inclusion/exclusion criteria, the final pool reached 30 papers from 291 initial sets of papers. After deciding included papers as a final pool, all of the papers were embedded in an Excel sheet and, the data extraction processes were made on this sheet. In one iteration, the first author extracted all of the attributes to find answers to the RQs. Then, the second author checked the extracted attributes made by the first author. Any conflict in this stage was discussed by the authors, and a final decision was made together. A detailed procedure followed during the study is shown in Fig. 3.

4.5 Quality Assessment

After inclusion/exclusion criteria were applied to the papers, we screened 30 papers in the Excel sheet. From here on, we proposed to evaluate the papers' quality. Guidelines of Kitchenham and Charter on determining the quality criteria were followed in this study (Kitchenham and Charters 2007). Therefore, we set four quality criteria (QC), each QC has 0.5 points and put papers in the final pool if they reach 1.0 as a score. If the score was lower than 1, we removed them from the final pool. In this study, we also give detailed information about the quality assessment of relevant papers in the final pool with RQ 5 and its sub-questions. Hence, we set the QC as followed.

Quality Criteria (QC)

- QC1. The number of citations. We gathered the citation numbers of each paper from Google Scholar and checked this criterion by identifying the citation numbers. A paper's citation number was checked, and if the paper had more than 0 citations, then it got 0.5. Otherwise, it got 0 points as the score for this quality criteria.
- QC2. Contribution of methodology. We checked the papers with this criterion by looking for the paper's relevance with software quality and finding a methodology followed in the study.
- QC3. Clear outcomes as results. We evaluated the findings if they are presented sufficiently or not, the validity of these findings, and the quality of results.
- QC4. Availability of conclusion. We specifically checked the availability of future works and discussion of validity threats in the papers.

For each quality criteria, the scores were boolean (if the study met the related quality criteria, then it got 0.5; otherwise, it got 0 points). As an initial step, the first author assigned a score for each quality criteria for each paper, and then the second author checked the correctness of the scores. If there were any disagreements with the scores of a paper, the authors discussed the related criteria together and made a consensus about the paper's scores.

After considering the QC for each paper, we decided to exclude one more paper from the 30 papers in the final pool. Hence, as a final pool, we got 29 relevant papers (see in Table 18 in Appendix) for the data extraction and synthesis stage. The study (Mannarswamy et al. 2020) was excluded because it did not provide appropriate information for our purpose, there was no method contribution and, there was no discussion of validity threats. Since this study did not pass the quality assessment as defined above, it was excluded from the final pool.

4.6 Paper Review and Data Extraction

In this step, we proposed to gather data to answer RQs as defined in the study. For making a detailed investigation, we applied the data extraction process to all of the 29 papers in the final pool. Firstly, we downloaded all the papers from the databases. Then, an Excel sheet was created to record all of the necessary and beneficial information that we collected. To avoid bias with the data extraction, two authors conducted the data extraction process one after the other author. When there was an unsolved dispute between the two researchers, they first discussed the reasons they support, and then they met an agreement for the decision. Also, all the papers and data extracted from the papers were stored in the Google sheet (Gezici and Tarhan 2019) for analysis by independent researchers.

5 Results

The goal of this section is to summarize and analyze the data extraction results by considering defined research questions (RQs). We provided all of the findings through the following sections.

5.1 RQ 1. Background and State-of-the-Art Understanding

In this part, we proposed to see the state of the art through RQ 1.2-RQ 1.5 in the field of software quality for AI-based software after obtaining the sights from researchers about what the software quality is with RQ 1.1.

5.1.1 RQ 1.1. How quality is defined in AI-based software?

In this part, we propose to understand the researchers' focus on quality in AI-based software. To do this, we recorded the definition of "quality" reported in each study. Our motivation with this RQ is to guide the audience to comprehend 1) what AI-based software quality is and 2) the researchers' perceptions on quality for AI-based software. Figure 4 represents a word cloud that shows the frequency of the words that appear in the original definition

Fig. 4 Word cloud for RQ 1.1. How is quality defined in AI-based software?

of quality in each study, which can roughly reflect the perception of researchers on quality for AI-based software. Although we focus on software quality in this SLR, some studies focused on different quality perspectives for AI-based software. The most frequently appeared words include: system quality, software quality, product quality, external quality, data quality, service quality, complexity, safety, process, etc.

According to our observations, there is no clear definition of quality for AI-based software in the literature. In general, we could not extract information from the paper directly, and we read all the text in the papers and made a classification. Therefore, to answer RQ 1.1 more in-depth, we provide detailed information by categorizing the definitions of each study as shown in Table 4.

These results show there are multiple but not uniform definitions of quality in the literature, and researchers tend to research system quality in addition to software/product quality in general. The lack of a common view about quality perception and definition indicates how inadequate the studies in this area are.

5.1.2 RQ 1.2. What is the trend of the number of studies on quality (by year/ by type) in AI-based software?

In this part, we present the publication frequency of primary studies from 1988-2020. Figure 5 shows the number of studies published by years between 1988–2020 concerning quality for AI-based software, which constitutes a total of 29 papers.

Observations from Fig. 5 show that there were only one or two studies per year in the field of quality for AI-based software until 2018, in which there was a peak with 6 studies. However, in 2019 and 2020, we can observe a significant growing curve with 8 studies each year. According to findings, we can conclude that in the last three years (2018, 2019, and 2020), the software engineering society has realized the necessity of research focused on quality for AI-based software.

Table 4 Classification of the quality definitions for AI-based software

Category	Perspective	Literature
Software/Product Quality	concerns with development-time of a system such as functional suitability, maintainability, etc.	Kuwajima et al. (2020), Nakamichi et al. (2020), Nguyen-Duc and Abrahamsson (2020), Liu et al. (2019), Nakajima (2019), Kuwajima and Ishikawa (2019), Arpteg et al. (2018), Nakajima (2018), Nishi et al. (2018), and Hyun Park et al. (2017)
Service Quality	provides a broad approximation for supplying dynamic consumer requirements	Hyun Park et al. (2017) and Nakajima (2018)
System Quality	concerns with quality perspectives of the system and its components (data, process, model, etc.)	Siebert et al. (2020), Nakamichi et al. (2020), Poth et al. (2020), Hamada et al. (2020), Kuwajima et al. (2020), Tao et al. (2019), Vogelsang and Borg (2019), Kuwajima et al. (2018), Tao et al. (2017), and Chen et al. (1995)

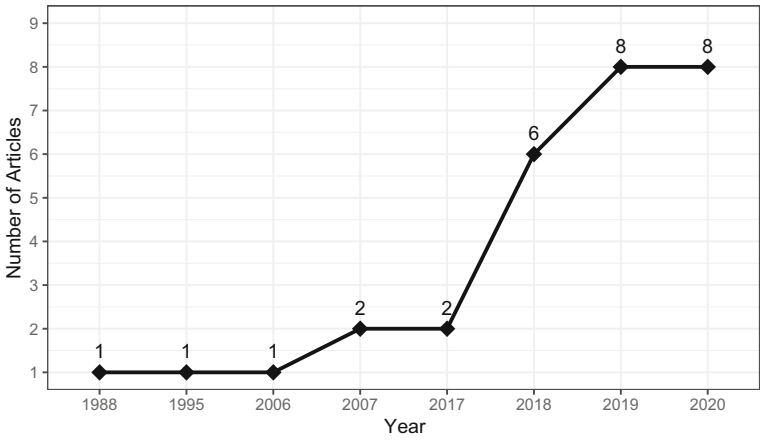

Fig. 5 Publication frequency, 1988–2020

We evaluated the distribution of studies by considering publication type as shown in Fig. 6. The most common publication type is conference papers (C) with 14 studies, followed by journal papers (J) with 9, and, finally, symposium or workshop (S/W) papers or book chapters (Book C.). We observed that in 2020, there were widespread conferences over journals. These results indicate that this area is perceived as emerging by researchers in both academia and industry.

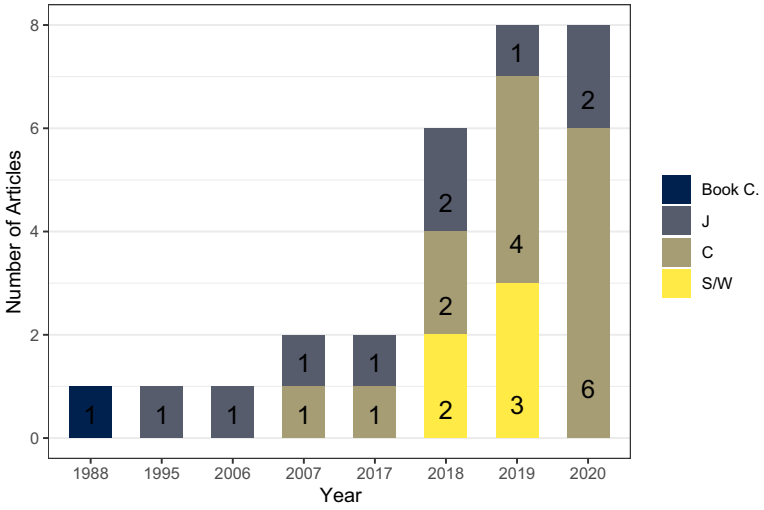

Fig. 6 Publication frequency by type

5.1.3 RQ 1.3. What are the research methods employed in the literature?

In this part, we investigated the types of research methods that are used by the authors of the articles. We adopted the following research methods from Wieringa (2014) in this SLR study:

- Solution Proposal: The solution that may be an important extension of a new or existing technique for a problem is proposed. The possible advantages and viability of the solutions are illustrated via a small case or a good discussion thread.
- Weak Empirical Study: The techniques examined are novel and not yet applied in practice. Papers with no hypotheses or research questions were classified as Weak Empirical Study.
- Strong Empirical Study: The techniques are applied in practice and, the assessment of the implemented technique is carried out. In other words, solution implementation and implementation evaluation that is related to discussing threats and drawbacks are shown in the papers.
- Experience Papers: They describes how to do something in practice, taking into account the personal experience of the authors.
- Other: Research methods that can not be put under any categorization proposed by Wieringa (2014).

Figure 7 shows the distribution of articles according to the type of research method. According to the figure, the most applied research method is the weak empirical study by 18 studies.

Figure 7 shows the cumulative distribution of the same articles by year concerning the type of research method. While no solution proposal study employed a strong empirical method and experience paper as a research method in the early years, it is observed that there is a trend towards the recent years for the papers employing these research methods. Also, there is an increasing trend with weak empirical studies throughout the years. Since more than half of the studies used weak empirical methods, it can be said that the contribution of the papers is far from being mature enough in this field. Uncertainty in the way about how to develop AI-based software with high quality highlights the necessity of studies that guides the practitioners. Especially starting from 2017, it is observed that there are attempts to guide researchers by defining research road maps to be followed in the context of quality for AI-based software.

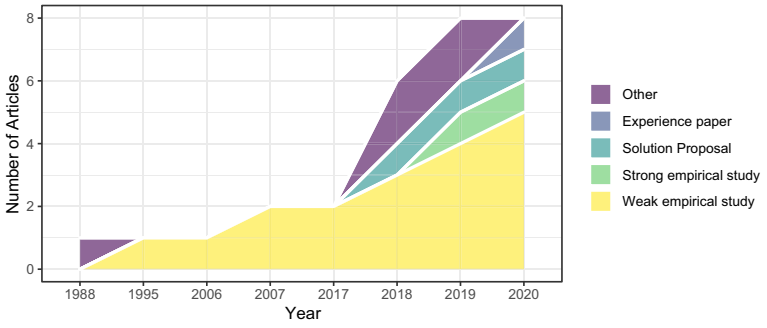

Fig. 7 Types of Research Methods employed in primary studies by Years

5.1.4 RQ 1.4. Who is leading the research in the quality of AI-based software? (Academic, Industry, or Collaborative)

We determined whether an article is academic, industrial, or collaboration between these two by considering the affiliation of the authors in each study. If the affiliations of all authors were with a university, then we determined that the article was academic, if the affiliations of all authors related to the industry, then we determined it as industrial and, if the affiliation of the authors were mixed, i.e., they were from university or industry, then we determined the article as collaborative. According to the analysis of the academic, industry, or business associations of the articles, it is found that 15 articles were carried out with academia, 9 studies within the industry, and 5 studies with academia-industry collaboration. These results indicate that there is a room for further collaborative studies in this emerging research area.

5.1.5 RQ 1.5. What type of contributions has been made in this area up to now? (model, metric, tool, case study, etc.)

This question maps the studies by contribution type. We searched for the number of articles that presented a new method, metric, tool, etc. Table 5 presents the distribution of the 29 primary articles included in the article pool by type of contribution.

As shown in Table 5, 18 articles contributed a new method/technique/ approach, followed by 7 articles with the contribution of a model. Some articles are classified under more than one contribution. For example, the study (P3), contributed method, metric, and process. The author of the study presented a requirement-driven method with the ML software quality requirements development process, extended the ISO25010 quality model (ISO/IEC 2011) with additional attributes, and provided metrics. Since there is no tool contribution, and there is a limited number of contributions of metric (16%) and model (17%), these results indicate that there is a critical need for such contributions in the context of quality for AI-based software. In addition, it is observed that when the studies proposed a new model, they generally focused on the current quality models and tried to modify or add novel quality aspects to them. They also specified different views of software systems such as model view, system view, data view, etc. and categorized several quality attributes and corresponding metrics with respect to these views.

Table 5 Types of contributions made by primary studies

Contribution Type	#of Articles	References to Articles
Method/Technique/Approach	18	P1, P3, P4, P7, P8, P9, P11, P13, P14, P15, P16, P18, P19, P20, P23, P27, P28, P29
Model	5	P1, P7, P22, P25, P28
Process	4	P3, P4, P6, P10
Empirical (Case) Study	4	P12, P17, P24, P26
Metric	4	P1, P3, P6, P14
Tool	0	-
Other	5	P2, P4, P5, P21, P29

5.2 RQ 2. Requirement elicitation for AI-based software quality

The motivation of RQ 2 is to identify the challenges of software quality that are experienced or addressed by the researchers, and the application domain/type which the developed system must meet.

5.2.1 RQ 2.1. What are the experienced challenges of quality in AI-based software?

By answering this RQ, we provided a detailed understanding of the popular research areas being investigated up to now in the scientific literature. As shown in Fig. 8, we classified each challenge under a specific topic.

1. Software Quality There are traditional quality models in the software engineering world. Since AI-based software is different from traditional software, there is no well-defined guideline, framework, road map, or model on measuring software quality for AI-based software. Therefore, inadequate process definitions to be followed, quality metrics, attributes, and assurance techniques cause new challenges in the academia or industry.

2. Software Development AI-based software systems consistently display unique characteristics (e.g. being black-box) in engineering because models (components) are constructed by training with data in an inductive manner. Also, it is possible to encounter unexpected outcomes. Hence, it might cause new problems with development processes when trying to evaluate the quality of these systems.

3. Design This category is about designing systems that imply ML models (components) by considering and performing the characteristics of "Change Anything Change Everything" (CACE) (Kuwajima and Ishikawa 2019). A slight change in training data may affect learning results, thus on the functional behavior of such systems.

4. Social Aspect This category is about the communication between developers, development skills, the combination of data scientists, software engineers, and different kinds of branches when developing such systems.

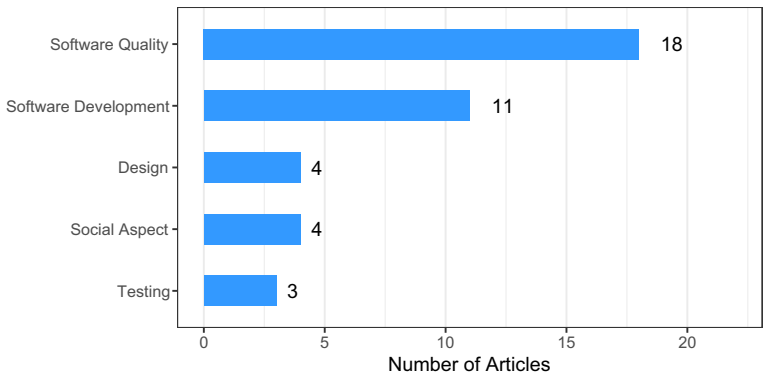

Fig. 8 Distribution of experienced challenges in the primary studies

5. Testing Testing ML systems pose difficulties that appear from the different nature of ML systems, compared to relatively more deterministic conventional software systems (Zhang et al. 2020). Some machine learning applications are intended to learn the properties of data sets in situations where the correct answers are not known to human users. Testing and debugging such machine learning software is difficult because there is no reliable test oracle. Murphy et al. (2006).

According to the results, the "Software Quality" challenge is the most experienced challenge faced by the researchers with 18 studies, followed by 11 studies with the "Software Development" challenge. We summarize each category from a software engineering perspective, and provided the detailed information about the classification in Table 6.

The first challenge of the "Software Quality" category is about the lack of guidelines to be followed when developing high-quality software for AI-based software. There is a limited number of studies that investigate the relationships between quality attributes and their measurements for AI-based software. Other challenges in this category include the difficulties caused due to the lack of a standard or quality model established for such systems. While there are several traditional quality standards followed for the traditional software systems, there are no such guidelines yet for AI-based software and it therefore becomes critical to select the convenient quality attributes for such systems.

The list of challenges in the "Software Development" category is about the difficulties due to the different nature of AI-based software from traditional ones. The necessity of new measurements has emerged to understand and explain the quality characteristics required for AI-based software since developing such software is inherently challenging and not transparent. Unlike traditional software, it solves more complex problems, its behavior depends on training data, and it is dependent on data and processes.

The first challenge in the "Design" category is specific to AI-software, especially in cases of critical applications, e.g., in medical or self-driving car fields. For such systems, it is important to get accurate, reliable, and secure ML responses fast. Therefore, designing such systems by defining algorithm, data, system/infrastructure to meet system requirements (reliability, adaptability, maintainability, etc.) as specific to the cases and deciding which quality attributes to be considered beforehand are among the important problems.

The challenges of the "Social Aspect" category are related to the difficulties about developers' skills, lack of training, communication problems between developers from different fields due to the differences in terminology, etc. The importance of multidisciplinary work is incontrovertible. When developing high-quality AI-based software, experts who have different kinds of knowledge must come together and complete the development of software by considering the specific parts of the software system with their knowledge separately.

The last challenge category of "Testing" brings new and unprecedented difficulties since the outcome of such systems depends on both code and training data. Repetitions of the training phase may lead to different behaviors. So, there is no exact output and deterministic oracles for the learning process.

While eliciting and gathering answers to RQ 2.1, the most important point that drew our attention is that regardless of the study being academic or industrial, before making a recommendation for software quality for AI-based software, most studies firstly mention challenges/risks such as the applicability of existing models for AI-based software, the lack of a standard that can be followed, and the inaccuracy of the measures used; and then propose solutions by adopting these challenges/risks for the benefit of the AI-based software. In addition, industry-based studies often talk about the difficulties or risks they face in the

Table 6 Challenges identified by the researchers for AI-based software quality

Challenge category	List of challenges	References to articles
Software Quality	(1) "The combination of and the relationship between quality attributes and related metrics have not been sufficiently investigated yet and it is not clear whether they can be satisfied altogether. Comprehensive development guidelines for quality-aware ML systems are largely missing or not made explicit." Siebert et al. (2020). (2) "The model (component) and characteristics of ML software quality are neither established nor standardized." Nakamichi et al. (2020). (3) "The practice of software engineering for systems that incorporate an AI component is at present an open research topic." Pons and Ozkaya (2019). (4) "There are so many quality models but it is necessary to select appropriate quality attributes for AI systems." Vinayagasundaram and Srivatsa (2007).	P1, P2, P3, P4, P6, P7, P8, P10, P11, P13, P14, P15, P16, P19, P20, P22, P23, P24, P25, P26, P29
Software Development	(1) "The outcome of an ML system is highly intractable and non-transparent due to its mathematical complexity, as well as the dependency of its behavior on both the training and in-use data and processes." Poth et al. (2020). (2) "Scientific-based development instead of engineering-based development. Uncertainty in system outputs, responses, and decision makings." Tao et al. (2019).	P1, P2, P3, P4, P5, P6, P9, P12, P17, P21, P29
Design	(1) "Creating mature high-quality products and services is very hard. Quality assurance (QA) principles should be pro-actively incorporated by design beforehand, instead of reacting on quality claims during product/service use." Poth et al. (2020). (2) "An open engineering problem at the system level of ML solutions is designing systems that include machine learning models (components) by considering and applying the characteristics of Change Anything Change Everything (CACE)." Kuwajima et al. (2020).	P1, P4, P6, P18

Table 6 (continued)

Challenge category	List of challenges	References to articles
Social Aspect	(1) "Developers' skills and training." Lenarduzzi et al. (2021). (2) "Communication between AI developers and other developers." Cummaudo et al. (2019) and Lenarduzzi et al. (2021). (3) "Organizational challenges (e.g., effort estimation, privacy and data safety." Arpteg et al. (2018). (4) "Necessity of combining quality experts with data scientists." Hyun Park et al. (2017).	P2, P12, P17, P23
Testing	"Some machine learning applications are intended to learn properties of data sets where the correct answers are not already known to human users. It is challenging to test and debug such ML software, because there is no reliable test oracle." Murphy et al. (2006) and Poth et al. (2020).	P2, P4, P27

industry and then include their solutions to overcome these difficulties. In academic studies, on the other hand, it is frequently aimed to make investigations of challenges from the literature and to propose solutions for them. In general, regardless of the affiliation type of the authors, the studies focus primarily on adapting challenges/risks to the characteristics of the AI-based software, based on previous experiences, risks or challenges encountered.

5.2.2 RQ 2.2. What are the addressed challenges of quality in AI-based software?

After looking at the challenges identified in the papers, we now focus on addressed challenges (RQ 2.2) with proposed solutions (RQ 4.1) in the papers specifically. In Fig. 9, we presented all the addressed challenges in the papers as a word cloud. Since each study focused on one or more specific challenges, we give the details for the addressed challenges per study in Table 19 (Column B) of Appendix in Section 1. If we evaluate the extracted data in Figure and Column C in the table, we see that all of the studies attempt to address the challenges in the "software quality" category, while 9 studies tried to address the challenges in the "software development" category. There are also addressed challenges in the "testing" and "social aspect" category.

5.2.3 RQ 2.3. What are the domains of AI-based applications that are subject to the studies?

In this part, we focus on the domains of AI-based systems that are investigated in the primary studies. "Domain" is about the business area that the developed system must meet, such as finance, medicine. Table 7 shows the application domains. According to the findings, while most of the authors (in 15 studies) did not specify the application domain in their studies,

Fig. 9 Word cloud for addressed challenges of quality in AI-based software

the authors in 5 studies used "Automotive" systems when experimenting with the studies, and there is a domain diversity in the experiments.

5.2.4 RQ 2.4. What are the types of AI-based applications that are subject to the studies?

In this part, we focus on the types of AI-based systems that are investigated in the primary studies. "Type" is about application types such as ML, AI, etc. Figure 10 shows the application types. According to the findings, for the application types of the studies, while 13 studies considered ML software, 11 studies used AI software as a subject, and 6 studies investigated their research by considering other types of software such as deep learning (DL), deep neural network (DNN), and computer vision, etc. The "Other" category is also related to AI or ML software, but some studies shrink their research area, and we aimed to enable readers to see it from a more narrowed perspective by categorizing them specifically.

Table 7 Application domains investigated in the primary studies

Automotive	Healtcare	Aircraft	Scientific software	Others	NA
P4, P6, P19, P20, P23	P11	P28	P9, P24, P26	P1, P3, P5, P17	P2, P7, P8, P10 P12, P13, P14, P15, P16, P18, P21, P22, P25, P27, P29
5	**1**	**1**	**3**	**4**	**15**

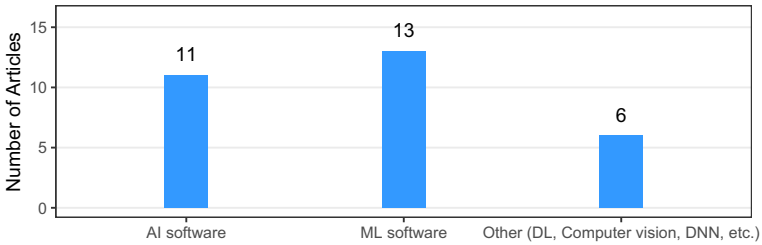

Fig. 10 Application types investigated in the primary studies

5.3 RQ 3. Design for AI-based software quality

The motivation of RQ 3 is to identify quality attributes that are subject to the studies and quality models contributed or adopted in the studies.

5.3.1 RQ 3.1. Are there any quality models adopted to AI-based software? If yes, what are they?

In the previous sections, we have mentioned that there are quality models such as ISO/IEC 25010 (ISO/IEC 2011) proposed for traditional software, but it is not always possible to use these models for AI-based software due to the different nature of these products. Therefore, in this part, we investigate whether there is any traditional quality model adopted for AI-based software or not. Before that, we give a summary of traditional quality models investigated in this SLR study, in the following paragraphs.

ISO/IEC 9126: It has been developed to evaluate software product quality in 1991. While it defines 6 quality characteristics (Functionality, Reliability, Usability, Efficiency, Maintainability, Portability), it is also subdivided into 27 sub characteristics. The measurement in this model covers internal, external, and quality-in-use.

ISO/IEC 25000: The model provides the details on the transition of the ISO/IEC 9126 series to SQuaRE (Software product Quality Requirements and Evaluation). The main objective of this model is to define requirement specifications and evaluate software quality by supporting the measurement process. It consists of six main divisions which define different parts of the quality such as Quality Management Division (The ISO/IEC 2500n), Data Quality (The ISO/IEC 2501n), Data Quality Measures (ISO/IEC 2502n), Division of quality evaluation (ISO/IEC 2504n), Quality Requirements (ISO/IEC 2503n), Square extension standards (ISO/IEC 25050–25099).

ISO/IEC 25010: This quality model defines software product quality as in which degree customer's needs and expectations are achieved for the system. The model identifies these needs under quality attributes and sub-attributes in a hierarchic structure. When recommending this hierarchic model, it refers to the ISO/IEC 9126 quality model and revised

this model. To evaluate the software product quality internally and externally, there are 8 quality characteristics which are Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, Portability, and the characteristics are evaluated based on 31 sub-characteristics in this model.

ISO/IEC 29119: This model covers a set of standards related to software testing. It is a multi-layered model; at the top of the layer there is an organizational test process, under this layer, there are test management processes which are test planning, test monitoring & control, and test completion processes, and at the bottom of the layer there are processes which related with test management such as test design & implementation, test environment set-up & maintenance, test execution, and test incident reporting processes.

ISO/IEC 26262: This standard is adapted from "IEC 61508: Functional Safety" and complies with road vehicles' functional safety. It defines six phases which are management, development, production, operation, service, and decommission through the automotive life cycle.

Figure 11 shows the number of studies that adopt any quality models. According to results, adopting a quality model for AI-based software has been increasing through the year 2020. Since there is an increasing trend starting with 2018, the researchers attempt to investigate whether any traditional quality models can be adopted to AI-based software or not. Just 7 of the 29 primary studies (%24) focus on adopting any quality models, and it remains an open and hot topic in the literature. In Table 8, we provide the adopted quality models per study. We can observe from the findings that the most focused quality models are ISO 25000 and one of its division quality models, ISO 25010 standard.

While some studies attempt adopting traditional quality models such as ISO 25010 (ISO/IEC 2011), we can say that the studies of academy-industry collaboration generally focus on extending the ISO 25010 quality standard or adapting it to the characteristics of AI/ML software. However, there is not yet a consensus on whether ISO 25010 is appropriate to use for AI-based software or which characteristics of AI-based software may be mapped to the attributes of traditional quality models.

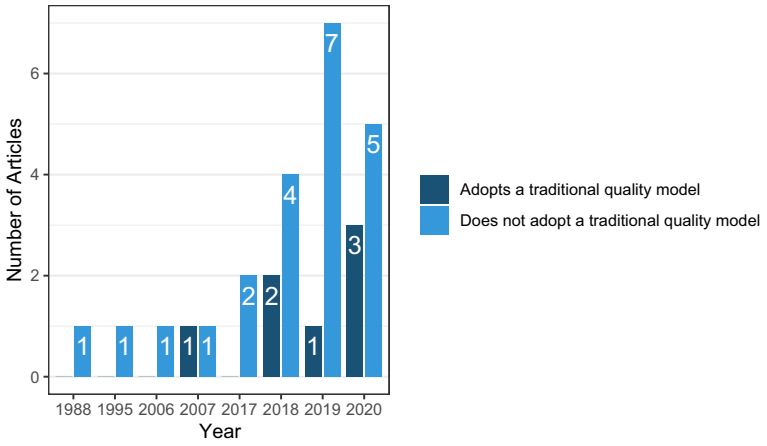

Fig. 11 Primary studies adopting traditional quality models

Table 8 Adopted quality models per study

Quality models	ISO 25010 (ISO/IEC 2011)	ISO 25000 (25000 2005)	ISO 26262 (26262-1:2018 2018)	ISO29119 (29119-1:2013 2013)	ISO 9126 (9126-1:2001 2001)
References to Articles	P3, P4	P6, P14	P20	P22	P25

5.3.2 RQ 3.2. Which quality attributes are studied on the quality of AI-based applications?

In addition to specifying the quality attributes focused in the primary studies, we also explore the quality metrics to measure these attributes so that the metrics can be taken into consideration for the quality of AI-based software.

Since it is the latest, most used, and updated model for software quality, we use the ISO/IEC 25010:2011 (ISO/IEC 2011) to extract the quality attributes of AI-based software. There are 8 categories with ISO25010 and, one "Others" category was added by us for the remaining attributes in the studies. Here, we give a summary of each category.

Table 9 shows the statistical distribution of the quality attributes. Some articles may involve more than one quality attribute.

From Table 9, we can see that while fewer studies focused on ISO 25010 quality attributes, most of the studies proposed to investigate the other quality attributes that are not defined with ISO 25010 quality standard, and this situation indicates that there is an inadequate search on ISO25010 quality standard for AI-based software. It is observed that "Reliability" and "Security" are the quality attributes the most investigated (in 8 studies), and "Usability," "Portability," and "Compatibility" attributes are the least studied ones in the literature. Now, we give some examples from the primary studies for summarizing each quality attribute.

- Functional Suitability: "Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions" (ISO/IEC 2011). It has three sub-attributes as: "Functional Correctness", "Functional Completeness", and "Functional Appropriateness". For the AI-based software perspective, the study (Siebert et al. 2020) investigated the "Appropriateness" quality attribute and defined this term as "Degree to which the model type is appropriate for the current task (e.g., classification) and can deal with the current data type (e.g., numerical, categorical)." As a quality metric or checklist, "Prerequisites for model type" is specified in the study.
- Reliability: "Degree to which a system, product or component performs specified functions under specified conditions for a specified period" (ISO/IEC 2011). Horkoff studied challenges and new directions for the NFRs of ML software (Horkoff 2019). He emphasized that software metrics recommended for the traditional SE can be applied to

Table 9 Distribution of the quality attributes in primary studies

Attributes	References to Articles	Frequency
Reliability	P1, P3, P4, P6, P14, P16, P27, P28, P29	9
Security	P3, P6, P8, P10, P11, P14, P16, P19	8
Maintainability	P3, P4, P6, P7, P14, P25	6
Functional Suitability	P1, P3, P4, P6, P14	5
Performance Efficiency	P1, P3, P4, P6, P7	5
Usability	P6, P14, P25	3
Portability	P4, P6	2
Compatibility	P4, P6	2
Others (robustness, safety, etc.)	P1, P3, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15, P16, P17, P19, P20, P22, P24, P25, P26	21

ML software, such as "number of runtime errors" for measuring "Reliability." However, he concluded that understanding how NFRs can be measured in practice for ML-based solutions is necessary for this domain.

- Usability: "Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO/IEC 2011). In (Kuwajima and Ishikawa 2019), while the researchers recommended: "Monitoring Capacity" measurement for "Operability" quality sub-attribute that is under "Usability" quality attribute, they also added a new quality sub-attribute by extending "Usability" with "Collaboratability" to reflect autonomous roles of AI systems.
- Performance Efficiency: "Degree of the performance relative to the number of resources used under stated conditions" (ISO/IEC 2011). It is composed of "Time Behaviour," "Resource Utilization," and "Capacity" quality sub-attributes. In Nakamichi et al. (2020), while "Time Behavior" sub-attribute was mapped to "Sufficiency of Temporal Performance" MLS quality sub-attribute, "Resource Utilization" was mapped to "Sufficiency of Capacity of Data Storage" and "Appropriateness of Resource Utilization" MLS quality sub-attributes.
- Portability: "Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another" (ISO/IEC 2011). In (Kuwajima et al. 2020), the researchers focused on two sub-attributes of "Portability" namely, "Adaptability" and "Replaceability," and recommended the metrics of "Number of functions which were tested in different operational environments" and "Number of functions which produce similar results as before" for measuring these sub-attributes.
- Maintainability: "Degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in the environment, and in requirements" (ISO/IEC 2011). Kuwajima and Ishikawa (2019) mentioned the difficulties of measuring the "Modularity" sub-attribute under the "Maintainability" and extended "Modifiability" sub-attribute for evaluating the "expected training time" of ML models (components) built-in ML-based software systems.
- Compatibility: "Degree to which a product, system or component can exchange information with other products, systems or components, and perform its required functions while sharing the same hardware or software environment" (ISO/IEC 2011). It is composed of the "Co-existence" and "Interoperability" sub-attributes. Poth and his colleagues (Poth et al. 2020) mapped each ISO25010 quality attribute with a questionnaire to present the methodical improvement of the standardized characteristics for the AI and ML domain.
- Security: "Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization" (ISO/IEC 2011). Pons and Ozkaya (2019) emphasized the difficulty of securing AI systems since the models (components) and intermediate stages of the artificial intelligence system are not understood by humans and, they need machine assistance. Therefore, in the paper, they focused on ensuring the safety and security of these systems.
- Others: Quality attributes that have not been included in ISO 25010 quality model were investigated in the studies, such as readability, robustness, safety, etc.

When we look at the studies that investigate the usability of the ISO 25010 quality attributes for AI-based software, we observe that the distributions of the studies with authors

having academic or industrial affiliations are very close, i.e., 53% of the studies were based on industry or academia-industry collaboration, while 47% of the studies were based on academy only. This shows us the focusing on ISO 25010 for AI-based software is getting important for both academia and industry.

In addition to this, since we employ a bottom-up approach in this SLR study, we provide details of the quality attributes, related metrics, metric definitions and formulae used in the primary studies in Appendix Table 20.

5.3.3 RQ 3.3. How does AI-based software quality attribute map with the attributes in ISO/IEC 25010:2011 quality model?

Our motivation with this RQ is to list the papers which mapped the AI-based software quality attributes with the ISO/IEC 25010 quality attributes that is the latest system and software quality model (ISO/IEC 2011).

By surveying the papers, we observed so many metrics/checklists/ characteristics of AI-based software, and to shorten and systematize these, firstly, we defined them as AI-based software quality attributes, then mapped those with eight main characteristics of ISO/IEC 25010:2011 quality model. In Table 10, we list the papers that mapped quality attributes specific to AI-based software with ISO 25010 quality attributes in the first column. In the second column, we group and list all metrics/checklist/characteristics as quality attributes of AI-based software which we get from this SLR, and then in the third column of this table, we write only those characteristics for which at least one attribute can be found in the second column. For example, in P1, the researchers consider "precision, recall, f-score, and goodness of fit" as "Functional suitability." This is how we show each attribute in Table 10.

5.4 RQ 4. Implementation and evaluation of AI-based software quality

In this section, our motivation is looking for the distribution of assurance methods or approaches/techniques used in the primary studies to ensure the quality of AI-based software, and for the real cases which provide empirical evidence on the use of these methods, approaches, or techniques.

5.4.1 RQ 4.1. How are the challenges (in 2.2) addressed in the studies?

In this part, we give details about empirical methods that the papers followed when addressing the challenges. To see the methods that the papers followed during their investigations, we made some categorization over the studies. By reviewing the papers, we defined three categories of empirical methods which are (i) interview/survey, (ii) literature analysis, and (iii) experience, for answering this RQ. The first category is about interview/survey. In this category, we included the papers whose authors performed interviews/surveys with the appropriate stakeholder, developer, expert, etc. Those authors defined some questions and used a questionnaire, interview, or survey in their studies. In the second category, we grouped the papers that made their contributions theoretically based on the previous studies in the literature. In the third category, we included the papers whose authors used their own experiences in AI-based software development or from the experiences of the experts in this field. In Table 11, you can see the matching of each approach to the papers that employed it. It is observed that 20 papers considered the literature analysis, 7 papers used the experience of their own authors or other experts, and 6 papers conducted interviews or surveys while conducting their studies. Some papers are matched with more than one category. For

Table 10 List of the papers that mapped the quality attributes in the primary studies with the ones in ISO25010

Paper	List of quality attributes	Mapping with ISO/IEC 25010
P1	(1) "Goodness of fit, Appropriateness, Infrastructure suitability" (2) "Resource utilization, Supervision overhead/efficiency, Training efficiency, Execution efficiency" (3) "Robustness"	(1) Functional Suitability (2) Performance Efficiency (3) Reliability
P3	(1) "Sufficiency of accuracy of trained model, Appropriateness of model training process, Appropriateness of means of creating training dataset, Appropriateness of quality maintenance means for training" (2) "Suitability of input data, Robustness against change of input data, Robustness against noise data, Understandability, Suitability of input data quality maintenance means" (3) "Sufficiency of temporal performance, Sufficiency of capacity of data storage, Appropriateness of resource utilization" (4) "Easiness of resource update, Easiness of software update, Easiness of system status analysis" (5) "Appropriateness of security and privacy assurance means"	(1) Functional Suitability (2) Reliability (3) Performance Efficiency (4) Maintainability (5) Security
P4	(1) "Adequacy, Bias, Completeness, Compliance" (2) "Execution environment, Worst cases (compute power, memory size)" (3) "Third party reliability" (4) "Completeness, Process-chain, Regulations/compliance, Training and testing chain, Model transparency, Model adequateness, Model robustness, Model completeness, Configuration, Execution environment, Monitoring, Validation" (5) "Training and testing chain, Third party, Model fitting" (6) "Model fitting"	(1) Functional Suitability (2) Performance Efficiency (3) Compatibility (4) Reliability (5) Maintainability (6) Portability
P6	(1) "Number of functions that are incorrect, Number of functions missing or incorrect among those that are required for achieving a specific usage objective" (2) "Number of functions which were tested in different operational environments, Number of functions which produce similar results as before" (3) "Number of system/software failures actually occurred, Number of failures detected during observation time" (4) "Number of test functions required"	(1) Functional Suitability (2) Portability (3) Reliability (4) Maintainability
P11	(1) "Adversarial inputs and data poisoning"	(1) Security
P14	(1) "Functional coverage" (2) "Test function completeness, Expected training time, Structural complexity and Behavioral complexity (robustness)" (3) "Integrity, Adversarial examples, Privacy" (4) "Monitoring capacity, Collaboration effectiveness"	(1) Functional Suitability (2) Maintainability (3) Security (4) Usability

Table 11 The empirical methods papers that the primary studies followed to address the challenges

Paper ID	Interview/ Survey	Literature	Experience
P1	✓	✓	✓
P2			✓
P3	✓		✓
P4	✓	✓	✓
P5	✓		
P6		✓	
P7			
P8		✓	
P9		✓	
P10		✓	
P11		✓	
P12			✓
P13			
P14		✓	
P15	✓		
P16		✓	
P17			✓
P18		✓	
P19		✓	
P20	✓		✓
P21		✓	
P22		✓	
P23		✓	
P24		✓	
P25		✓	
P26		✓	
P27		✓	
P28		✓	
P29		✓	

example, in P1, the authors considered the results from the literature analysis, discussions with experts (i.e. interview), and their experience together while categorizing the quality attributes, and contributed a model by considering various views of ML software systems. In addition, in Appendix in Section 1, we provide the details about the challenges and proposed solutions for all the papers. Since the studies proposed solutions for different challenges by following different road maps, we could not further categorize the solutions contributed by the studies. Column D in Table 19 shows the solutions proposed by each study.

5.4.2 RQ 4.2. Which assurance methods are used to guarantee the quality of AI-based software? (verification/validation, testing, analysis, monitoring, etc.)

To answer this RQ, we focus on extracting the quality assurance methods used in the primary studies. Figure 12 shows the distribution of studies that used assurance methods. By

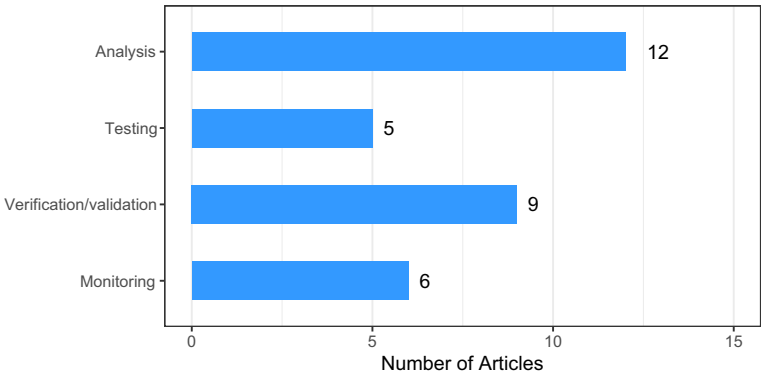

Fig. 12 Distribution of different quality assurance methods in primary studies

reviewing each paper in the final pool, we specified four methods in our study, i.e., analysis, testing, verification/validation, monitoring.

Since the quality assurance of software concerns the entire AI-based software development process, we divided assurance methods into four categories (i.e., analysis, testing, verification/validation, monitoring) that were observed with the design-time and run-time. In Table 12, it is shown a list of assurance methods with their definitions and related papers.

Findings from RQ 4.2 showed that the most used method for assuring AI-based software quality is the Analysis methods with 13 primary studies followed by Verification/Validation methods with 9 primary studies. Since the most used quality assurance methods are applied at design time, it can be said that it is hard to use quality assurance methods at run-time due to the challenges that arise from the different (e.g., black-box and complex) nature of AI-based software systems. In order to see the challenges in detail, please see Table 6 in RQ 2.1.

5.4.3 RQ 4.3. What are the approaches/techniques used for ensuring the quality of AI-based applications? (deep learning, supervised learning, classification, risk-based approach, etc.)

Our motivation with this RQ is to summarize the software quality approaches or techniques applied for AI-based applications from different perspectives.

ML-based techniques in which machine learning techniques such as neuro-fuzzy, supervised machine learning classifier, image processing, etc., are used for ensuring the AI-based software quality in the primary studies (P5, P8, P12-P14, P26-P28). For example, Kuwajima and Ishikawa (2019), to measure quality measure elements, implemented a large fuzzy function trained in a data-driven manner. By this technique, they focused on modifying the existing quality attributes in SQUARE.

Categorization-based approach in which various quality attributes or test categories are organized in the papers (P1, P22). In P1, from five different views (Model, Data, Environment, Infrastructure, System), the authors categorized quality attributes and derived a quality model.

Table 12 Identified quality assurance methods in primary studies

Assurance method	Stage	Description	References to articles
Analysis	Design-time	This method is to analyze the quality attributes of AI-based software to specify the primary factors that have an effect on their quality.	P1, P6, P8, P9, P12, P13, P14, P15, P23, P24, P25, P26, P28
Testing	Run-time	It is about besides approving application levels of performance, correctness, and other quality attributes but additionally checking the testability of AI-based software.	P5, P9, P13, P22, P27
Verification/Validation	Design-time	The goal of verification/validation is to confirm that the design of the output guarantees that the design stage of the input requirements is met.	P3, P4, P5, P7, P9, P15, P17, P19, P24
Monitoring	Run-time	In order to ensure the AI-based software quality, monitoring is used to find failures or potential anomalies, or see the performance, or the other quality attributes' distribution at runtime.	P8, P9, P12, P24, P28

Requirements-based approach in which the needs are determined and created as detailed functional documents for the AI-based software are investigated (P3, P6). In P3, to determine quality characteristics for AI-based software, the researchers proposed a requirement-driven method. In the paper, the requirement analyst extracted important issues, and the authors mapped these issues with ISO25010 quality characteristics.

Risk-based approach in which possible quality risks for AI-based software, and provide recommendations on these risks are identified (P4, P8). While P4 made a questionnaire and identified possible quality risks, P8 proposed a new "Neuro-Fuzzy workflow" process for reducing identified software risks to observe quantitative performance analysis of the competing products. The authors made an analytical contribution to the approaches of managing risks using risk management strategies for ensuring the quality of AI applications.

Rule-based approach in which pre-defined expert-based rules are established and used for ensuring the quality of AI-based software. One of the approaches of P5 is constructing a rule-based AI or traditional software to evaluate the target quality attributes.

Model-based approach in which a holistic view of AI-based quality model is constructed and quality attributes are specified. In P7, the researchers proposed modeling AI software quality and specifying quality attributes through their approach. They proposed to model "AI software quality and ethical attributes" by combining business values and ethical principles.

Test-based approach in which a property-based software testing technique is used as an effective approach for addressing the test oracle problem. In P9, the researchers provided the general concept of AI testing and performed an empirical study for quality validation of an image recognition system by applying metamorphic testing.

Findings from RQ 4.3 showed that the most used approach/technique for ensuring AI-based software quality is ML-based techniques with 8 primary studies followed by categorization-based and requirements-based approaches. At this point, the most critical observation that caught our attention is that the researchers, who carried out studies in industry or in cooperation, used the questionnaires or interviews to validate the completeness and practical relevance of their approaches followed in their studies. Taking out risks or requirements, conducting questionnaires/interviews by experienced people, and matching the answers to these questionnaires/interviews with the quality attributes can be cited as examples.

5.4.4 RQ 4.4. What are the real cases (i.e., validation context) of the assurance methods used in 4.2?

Our motivation behind this RQ is to elicit empirical evidence on the use of assurance methods used in RQ 4.2. In Table 13, we give details about cases with assurance methods. According to results, there is a limited number of studies that made an experiment for ensuring the quality of AI-based software. However, most of the experiments are based on realistic systems, and as a domain, there are different domains such as automotive, enterprise, face recognition systems, etc. Real-time systems process data as it comes in, and real-world and realistic systems utilize real-world data to support the querying and retrieval of specific content. As a result, further empirical work to include real case studies should be performed to get more insight into each domain/type and to clarify the appropriateness

Table 13 Experimental summary in primary studies

Method	Ref	Case Type	Domain	#of examples under studies
Analysis	P1	Lab experiment	Fujitsu's Purchase Order Request	3
	P8	Real-time system	OpenPose	1
	P9	Realistic system	Image Recognition System	1
	P23	Real-world	Automotive	7
	P24	Realistic experimental study	Face Recognition Systems	1
	P26	Real-time system	Software	9
	P28	Real-time system	Aircraft	2
Testing	P5	Real-world	(1) Generative Systems, (2) Operational Data in Process Systems, (3) Voice User Interface, (4) Autonomous Driving	4
	P9	Realistic system	Image Recognition System	1
	P27	Real-world	(1) CMarti, (2) PerlMarti, (3) FastCMarti	3
Verification/Validation	P3	Real-world	Enterprise Systems	3
	P4	Real-world	Enterprise Systems	10
	P5	Real-world	(1) Generative Systems, (2) Operational Data in Process Systems, (3) Voice User Interface, (4) Autonomous Driving	4
	P9	Realistic system	Image Recognition System	1
	P19	Real-world	Automative	3
	P24	Realistic experimental study	Face Recognition Systems	1
Monitoring	P8	Real-time system	OpenPose	1
	P9	Realistic system	Image Recognition System	1
	P24	Realistic experimental study	Face Recognition Systems	1
	P28	Real-time systems	Aircraft	2

of the quality models, attributes, and metrics in the context of AI-based software systems. Case types such as "Real-time system", "Real-world" etc. were generally transferred from the articles as written in them.

5.5 RQ 5. Rigor and Relevance discussed in papers with design knowledge contributions

The motivation of RQ 5 is to evaluate the rigor and industrial relevance of technology evaluations in software engineering. One of the indicators that this field is yet immature can be understood from the rigor of the studies. Evaluation of the rigor and relevance of

the studies has been performed by using the related criteria from the DSR method (Ivarsson and Gorschek 2011). While rigor of a design science study refers to the "strength of the added support for the technological rule and may be assessed concerning all of the three knowledge-creating activities: problem conceptualization, solution design, and empirical validation;" relevance of a study refers to the "three aspects of evaluations that are considered in evaluating the realism of evaluations: subjects, scale, and context" (Ivarsson and Gorschek 2011). In the following subsections, we provide RQs related to Rigor and Relevance separately.

In this part, we provide details about RQ 5.1, 5.2, and 5.3 to make an assessment about the rigor degree of primary studies by considering three aspects shown in Table 14. Firstly, by referencing the study of Ivarsson and Gorschek (2011), we scored all three aspects of each primary study by classifying them with the same three score levels as a strong, medium, and weak as shown in Fig. 13.

After classifying, by assigning scores, we quantified each rigor degree and found a final score (rigor degree) for each primary study with the following formula (Ivarsson and Gorschek 2011):

Table 14 Extracted scoring for evaluating rigor (Ivarsson and Gorschek 2011)

RQs	Strong description (1)	Medium description (0.5)	Weak description (0)
RQ 5.1 In which degree the actions have been followed to ensure the intervention is a valid solution to the problem instance? (Context Described)	"The context is described to the degree where a reader can understand and compare it to another context."	"The context in which the study is performed, is mentioned or presented in brief but not described to the degree to which a reader can understand and compare it to another context."	"There appears to be no description of the context in which the evaluation is performed."
RQ 5.2 In which degree the actions have been taken to ensure the understanding of the problem instance is valid? (Study Design Described)	"The study design is described to the degree where a reader can understand, e.g., the variables measured, the control used, the treatments, the selection/sampling used etc."	"The study design is briefly described, e.g., ten students did step1, step2, and step3."	"There appears to be no description of the design of the presented evaluation."
RQ 5.3. In which degree the actions have been taken to validate the design choices? (Validity Discussed)	"The validity of the evaluation is discussed in detail where threats are described and measures to limit them are detailed."	"The validity of the study is mentioned but not described in detail."	"There appears to be no description of any threats to validity of the evaluation."

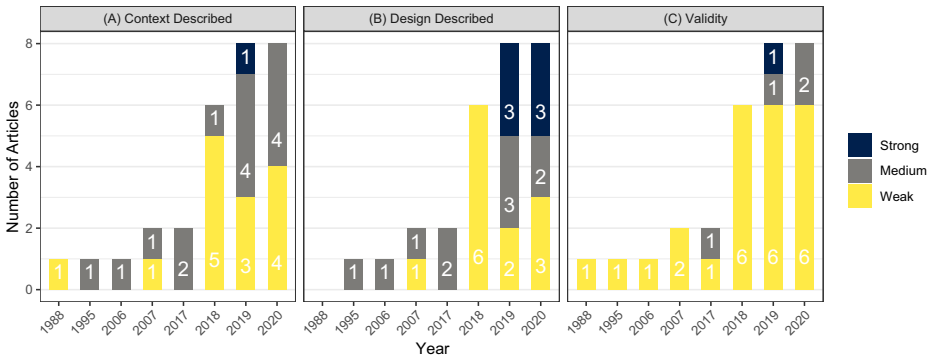

Fig. 13 Rigor degree of three aspects (Context, Design, Validity) in primary studies

`Rigor = C + D + V`
where (C) is Context described related with RQ 5.1, (D) is Study design related with RQ 5.2, and (V) is Validity described related with RQ 5.3.

In this part, we analyze the remaining sub-RQs of RQ 5 by investigating the industrial relevance degree of the primary studies. According to four aspects ("Subjects" for RQ 5.4, "Context" for RQ 5.5, "Scale" for RQ 5.6, and "Research method" for RQ 5.7), we evaluated each primary study by assigning them 0 or 1 for scoring. The scoring rubric that was used to classify each aspect is detailed in Table 15 by referencing the study of Ivarsson and Gorschek (2011). Each RQ (aspect) was evaluated according to descriptions for scoring degree 0 or 1. Aspects are scored as 1 if contributing to relevance, 0 otherwise. To quantify the relevance of each primary study, we used the following formula:
`Relevance = C + RM + U + S`
where (C) is Context, (RM) is Research method, (U) is User/Subject, (S) is Scale.

To analyze the resulting classifications to present an abstract view of the state of the technology evaluation, the classification is converted into numerical values. In the analysis, we used the average of these variables. The average for rigor should be interpreted as the average number of aspects per year that are described in the studies. The average of relevance should be interpreted as how many aspects contributing to relevance are included on average per year in the studies. Figure 14 shows how the variables of rigor have changed over time together with the number of papers included each year.

According to results, although the number of studies is increasing towards 2020, the primary studies do have not high rigor degrees, and the average rigor remains low. In addition, regarding the relevance of research carried out, it is observed that in two points (1995 and 2017), it was reached at the maximum degree of relevance which is four, and although the research field has not shown a big improvement in terms of average industrial relevance over time, it is also possible to say that from 2017 to 2020, there is an attempt in industry at that field.

We also assessed the pair of rigor and relevance assessments for each study, similar to what was done in Ivarsson and Gorschek (2011). The pair-wise comparison of rigor and relevance in this SLR is shown in Fig. 15. Only 7 of the 29 primary studies had zero rigor and relevance. This means that 24% of the studies were experiments in which aspects related to rigor were not described and were of application of a technology done by either students or researchers in academia in toy examples. On the contrary, just 1 study had high degrees of rigor and relevance.

Table 15 Extracted scoring for evaluating relevance degree (Ivarsson and Gorschek 2011)

RQs	Contribute to relevance (1)	Do not contribute to relevance(0.5)
RQ 5.4 To whom are those problem-solution pairs relevant? (Subjects)	"The subjects used in the evaluation are representative of the intended users of the technology"	"The subjects used in the evaluation are not representative of the envisioned users of the technology (practitioners). Subjects included on this level (Students, Researchers, Not mentioned)"
RQ 5.5 Does the research address the real problems or challenges that are of concern to professionals ? (Context)	"The evaluation is performed in a setting representative of the intended usage setting."	"The evaluation is performed in a laboratory situation or other setting not representative of a real usage situation."
RQ 5.6 How do the authors convince their readers that the problem-solution pair is relevant to those stakeholders? (Research method)	"The research method mentioned to be used in the evaluation is one that facilitates investigating real situations."	"The research method mentioned to be used in the evaluation does not lend itself to investigate real situations."
RQ 5.7 Is software research is biased toward huge projects? (Scale)	"The scale of the applications used in the evaluation is of realistic size, i.e., the applications are of industrial scale."	"The evaluation is performed using applications of unrealistic size such as Toy examples."

In general, to characterize the research carried out in this field, we analyzed the rigor and relevance degrees of each study, and we can say that by considering the relation of rigor and relevance degrees, both academia and industry have carried out immature-based research, and there is a necessity of future work that may set the relation between the two and also may impact both industry and academy in this field.

5.5.1 RQ 5.8. What is the average number of citations per year for the different levels of relevance?

In this part, for different levels of relevance, we focus on the average number of citations per year as shown in Table 16. While the degree of relevance increases, the number of citations first decreases and then increases. Since the relevance degrees and number of citations show

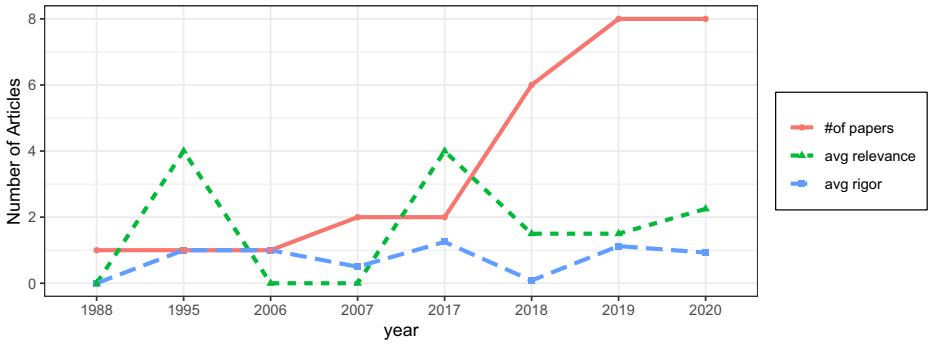

Fig. 14 Evolution of research for average rigor/relevance degree per year

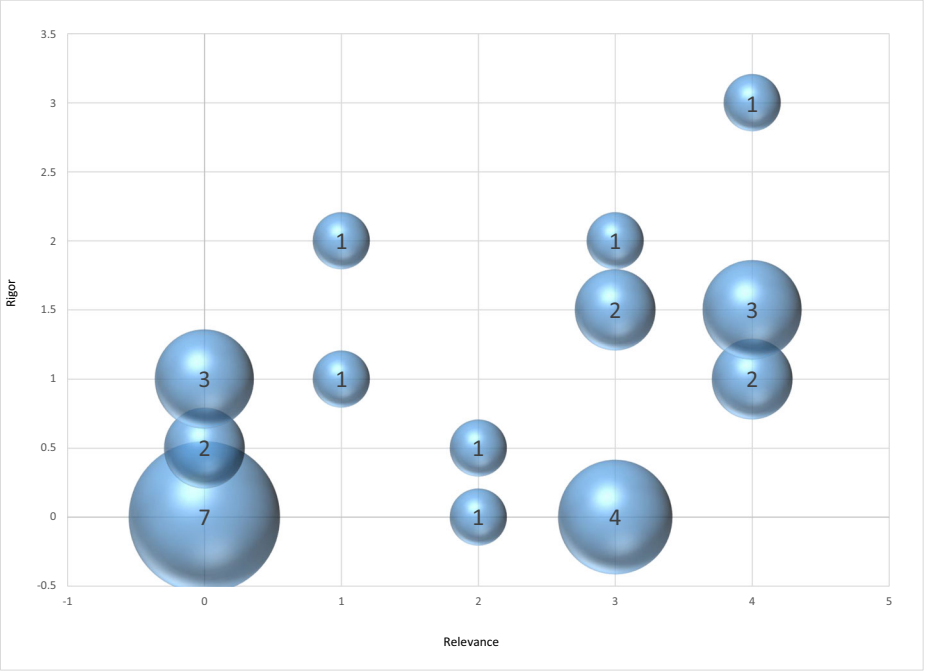

Fig. 15 Rigor versus relevance of the primary studies in this SLR

different distributions, we can conclude that there is no indication that the level of industry relevance presented in Section 5.5 has influenced academic relevance.

6 Discussion

6.1 Summary of Findings and Suggestions

In this section, we summarize the general findings and overview our suggestions related to the results.

Table 16 Average number of citations per year for different levels of relevance

Relevance degree	# of studies	Average # of citations to the studies
0	12	12.8
1	2	10.0
2	2	1.0
3	7	9.3
4	6	17.5

6.1.1 Findings from RQ 1

- Since the main purpose of this study was investigating software quality for AI-based systems, we firstly proposed to understand the perceptions of researchers on quality with RQ 1.1. By presenting a word cloud, we showed the frequency of each word that appeared in the primary studies. We created three categories which were "Software/Product Quality," "Service Quality," and "System Quality" in the context of this study because the most of the studies did not provide a clear definition of quality. Only 41% of the studies attempted to define quality. More specifically, while 34% of the primary studies investigated the quality as "Software/Product Quality" and "System Quality," just 7% of the studies was concerned with "Service Quality," and some studies covered more than one category. Consequently, we observed that there is no consensus on the perception and definition of quality for AI-based software. Further research is required to elicit and clarify related terminology on this topic.
- RQ 1.2 - RQ 1.4 were about the trends and contributions of the studies. With RQ 1.2, we proposed to see the trend of researchers' interests on software quality of AI-based systems. When we examined the studies carried out between 1988 and 2020, we observed that there was an increase in academia on this subject towards 2020. Although the interest towards software quality for AI-based software was more on the side of the academia (52%) compared to the industry (31%), we can say that there were attempts in both academia and industry; however, there were limited attempts for collaboration of academia and industry (RQ 1.4). In order to observe the maturity of the studies in this field, we examined the research methods employed in the studies with RQ 1.3. The fact that 62% of the studies applied weak empirical methods and that only 7% of the studies employed strong empirical research indicates that the field is not mature enough. In addition, 17% of the studies contributed a guideline, road map, etc., and these contributions were put under the "Other" category in this SLR. To conclude, creating technically reasonable drafts, proposing quality assurance processes, and setting standardization trends of AI-based software is critical. Since the maturity in the research methods applied is low, the research base also needs to be strengthened.
- Regarding the papers' contributions we designed RQ 1.5, and its results showed that the most of the studies (62%) contributed a method/technique/approach, following with model contribution in five studies (17%). Since only 14% of the studies contributed metric, the number of studies that made detailed measurements and had common metric recommendations was very limited. Therefore, for quality assessment, focus in future studies may be on taking bottom-up approach and metric-based research. Moreover, designing quantitative models for evaluating the quality of AI-based software by measuring quality attributes with software metrics should be a priority.

6.1.2 Findings from RQ 2

- RQ 2.1 is about experienced challenges of quality in AI-based software. A broad set of challenges exists due to the novelty of the subject. In general, standardized approaches for eliciting the quality of AI-based software are lacking. Considering quality attributes as specific to AI-based software characteristics is required to ensure the quality of these software systems. Many attempts have been made to fill the gap. However, understanding the problem is still deficient. We categorized the challenges faced by researchers by observing commonalities among several primary studies in this SLR. We

examined these challenges under five categories which are Software Quality, Software Development, Design, Social Aspect, and Testing.

- The "Software Quality" category, which had the highest rate (72%) among the categories we examined, included the difficulties in application of the quality attributes of the existing software quality models directly to the AI-based software. The corresponding studies mentioned that there is a lack of common processes that can be followed to evaluate the quality of AI-based software and that there is a need for adaptation of various traditional software processes/models for this type of software. Since the difficulties arising from the nature of AI-based software were also considered in the "Software Development" category, 38% of the studies mentioned the related challenges under this category. AI-based software produces models which are opaque and difficult to understand for people, so called black-box. This nature brings challenges especially for the data scientists in order to understand and interpret AI-based software systems. These challenges result in an inability to perform quality assurance or testing activities due to the lack of specifications that provide the knowledge necessary to understand, build, and test such systems. Therefore, understanding the nature of AI-based software and identifying how it differs from traditional software may help in eliciting related challenges and developing suitable definitions of software quality for these systems.
- There are also challenges we categorized under "Social Aspect", which were related to the communication challenges between AI-based developers and others, terminology differences that caused misunderstandings between developers, etc. Some studies mentioned the necessity of interdisciplinary work involving, e.g., requirements engineering, data science, and software engineering. The authors of these studies argued that the disciplines from many different technical fields might discuss these challenges in-depth and develop solutions in much more efficient ways. In conclusion, it seems logical to suggest that AI-based software development projects should search for SE experts, and SE researchers should search for real projects as the basis of their research designs. Moreover, developing guidelines on the collaboration of AI and SE experts may be beneficial.
- In the response to RQ 2.2, we also included the challenges addressed in the primary studies. Since this SLR study deals with the articles in the context of software quality for AI-based software, it was inevitable that the most addressed difficulty was again related to the "Software Quality" category. From the results, we observed that there were some attempts to address these challenges. However, the most of them examined specific cases and perspectives, and it is necessary to perform further empirical studies to validate the different quality aspects in more detail.
- According to the results of RQ 2.3 and RQ 2.4, we observed there were various domains and types of software included in the studies while performing empirical investigations. Since the domains and sub-types of AI-based software are various, define-your-own model approach might work better than a fixed-model approach in product quality modeling and evaluation, especially for the initial efforts. In addition, application domains investigated in the primary studies showed that the domains had diversity, so domain experts might also be included in the definition/specification of the quality attributes. Moreover, identifying and considering sub-types of AI-based software might be useful in eliciting unique requirements of product quality for each sub-type and in developing related quality specifications.

6.1.3 Findings from RQ 3

- In the previous sections, we have reported from the studies that it was difficult to use the traditional quality models for AI-based software due to the problems stemmed from the different nature of these systems. Therefore, we searched for the studies which examined the conventional models for AI-based software within RQ 3.1. According to the results, there was a limited number of studies that investigated traditional models such as ISO 25010 with 7 studies, and 22 studies did not have an attempt on adopting any traditional models. However, we also observed (from Fig. 11) that there was an increasing trend for adopting traditional quality models between 2018 and 2020. Since the number of primary studies adopting existing quality models was low, and the quality attributes not in ISO 25010 were widely used in the studies; it seems that the research base needs to be strengthened. In addition, there was a lack of evidence that the existing software quality models/standards could be applied for specifying AI-based software quality. Therefore, individual and context-specific efforts may be more helpful than the efforts spent for using/adopting the existing quality models for traditional software.
- With RQ 3.2, we focused on the quality attributes that the primary studies investigated for AI-based software quality. According to ISO 25010, which is the latest model and has eight characteristics of software product quality (ISO/IEC 2011), we categorized the attributes under these eight characteristics and in addition to these, we defined "Other" category that consisted of the quality attributes except the ones under ISO 25010. According to the results, 73% of the studies examined the quality attributes of the "Other" category, while 28% of the studies investigated the ISO 25010's quality attributes of "Reliability" and "Security." Since the most of the studies investigated the quality attributes such as robustness, safety, etc. under the "Other" category, we can conclude that there is a gap in the relation of ISO 25010's quality attributes with AI-based software quality characteristics. Therefore, while measuring the quality of AI-based software, considering AI-specific quality attributes is required.
- Lastly, with RQ 3.3, we investigated the studies that matched AI-based software characteristics with the ISO 25010's quality attributes. After mapping all quality attributes into their relevant AI-based software characteristics, we got numeric values. According to these values, we found that 83% of the studies that matched the AI-based software quality attributes with ISO 25010's quality attributes discussed "Functional Suitability", 27% of the studies discussed "Reliability" and "Maintainability", and 20% of the studies "Performance Efficiency" and "Security." According to our observations, these five quality attributes were the most studied quality attributes for AI-based software.

6.1.4 Findings from RQ 4

- With RQ 4, we focused on the assurance methods/approaches/techniques that were used in the primary studies for ensuring the quality of AI-based software. Firstly, we investigated the approaches in the primary studies for how they addressed the challenges, but we could not categorize them because each study dealt with distinct challenges it highlighted in a different context. As an assurance method, 42% of the studies used "Analysis", followed by 31% of the studies using "Verification/Validation". Since each study followed different approaches/techniques when ensuring the quality of AI-based software, we grouped them under seven categories (i.e., ML-based techniques, Categorization-based approach, Requirements-based approach, Risk-based approach, Rule-based approach, Model-based approach, Test-based approach). According to the

results, we observed that eight studies used ML-based techniques such as neuro-fuzzy and image processing for ensuring the quality of AI-based software. As a suggestion, evaluating the approaches/techniques explored in this SLR study with software professionals who have experience in the development of AI/ML projects may be rewarding for both researchers and practitioners who concern AI-based software quality.

6.1.5 Findings from RQ 5

– Finally with RQ 5, we proposed to evaluate the rigor and industrial relevance of the primary studies. By observing rigor, we focused on eliciting to what extent and detail the context of AI-based software quality evaluation was presented by researchers. Because failing to report aspects related to rigor makes it difficult to understand and use the results and limits the possibility for other researchers to replicate or reproduce the study (Wohlin et al. 2012). We calculated the average rigor degree for each primary study and observed that although the number of studies was increasing towards 2020, the primary studies did not have high rigor degrees, and the average rigor value remained low. Relevance, on the other hand, is about evaluating the potential to impact the industry. While the results showed an increasing number of potentially relevant studies between 2017-2020, there was just a slight change and the average relevance of primary studies remained about the same. As a result, the rigor-relevance relationship in the studies and the lack of a linear pattern in their related citations show the significance of further studies based on industrial problems. In this direction, the necessity of more empirical work on real cases has arisen.

6.1.6 Suggestions

In line with an overview of the findings that we derived from the answers to the RQs of this SLR study, we also included our suggestions in the above sub-sections. In Table 17, we gather all the suggestions to be considered for future work on AI-based software quality and group them under three categories, which are Definition/Specification, Design/Evaluation, and Process/Socio-technical. We also provide the links between the suggestions and the RQs in the rightmost column of the table. Accordingly, there are four items for "Definition/specification", four items for "Design/evaluation", and six items for "Process/socio-technical approach" as the suggestions for future work on AI-based software quality.

6.2 Threats to Validity

In this section, we identified several threads that might affect the validity of the outcomes in our study.

External Validity Since the data collection process was conducted by two researchers, this might lead to incomplete data collection, as some related articles might be missing. Although the authors discussed any disagreement, there might still exist a threat. Also, in the data extraction process, each researcher might be biased and inflexible while collecting data, so at each stage of the study, to mitigate the researcher bias, we ensured that two peers reviewed the work. In addition, the selection of academic papers in English was also a threat. English is the base language in academia, but this threat is considered as minimal reasonably. Another limitation may be considered as the exclusion of grey literature. We did

Table 17 Categorization of our suggestions in relation with the RQs

Category	Suggestions	RQs
Definition/specification of AI-based software quality	1- There is no consensus on the perception and definition of quality for AI-based software. Further research is required to elicit and clarify related terminology on this topic.	RQ 1.1
	2- Understanding the nature of AI-based software and identifying how it differs from traditional software may help in eliciting related challenges and developing suitable definitions of software quality in such systems.	RQ 2.2
	3- There is a lack of evidence that existing software quality models/standards can be applied for specifying AI-based software quality. Therefore, individual context-specific efforts may be more helpful than the efforts spent for using/adopting existing quality models.	RQ 3.1
	4- Identifying and considering sub-types of AI-based software maybe useful in eliciting unique requirements of product quality for each sub-type, and in developing related quality specification.	RQ 2.4
Design/evaluation of AI-based software quality	1- The number of studies that make detailed measurements and have common metric recommendations is very limited. Therefore, for quality assessment, focus may be on taking bottom-up approach and metric-based research.	RQ 1.5
	2- Considering quality attributes as specific to AI-based software characteristics is required to ensure the quality of these software systems.	RQ 2.2, RQ 3.2
	3- Designing quantitative models for evaluating the quality of AI-based software by measuring quality attributes with software metrics should be a priority.	RQ 1.5
	4- Since the domains and sub-types of AI-based software are various, define-your-own model approach may work better than a fixed-model approach in product quality modeling and evaluation, especially for the initial efforts.	RQ 2.3, RQ 2.4

Table 17 (continued)

Category	Suggestions	RQs
Process/socio-technical approach for AI-based software quality	1- Developing guidelines on the collaborations of AI and SE experts may be beneficial. Creating technically reasonable drafts, proposing quality assurance processes, and setting standardization trends of AI-based software is critical.	RQ 1.3, RQ 2.2
	2- Evaluating the approaches/techniques explored in this SLR study in collaboration with software professionals who have experience in the development of AI/ML projects may be rewarding for both researchers and practitioners who concern AI-based software quality.	RQ 4.3
	3- AI-based software development projects should search for SE experts, and SE researchers should search for real projects as the basis of their research designs.	RQ 2.2
	4- Application domains investigated in the primary studies shows that the domains have diversity, so domain experts may also be included in the definition/specification of the quality attributes.	RQ 2.3, RQ 2.4
	5- Since maturity in research methods applied is low, number of studies adopting existing quality models is low, and quality attributes not in ISO25010 are widely used in the primary studies; the research base needs to be strengthened.	RQ 1.3, RQ 3.1
	6- The rigor-relevance relationship and the lack of a linear pattern in related citations show the significance of further studies based on industrial problems. In this direction, the necessity of more empirical work on real cases has arisen.	RQ 5

not include in the review the non-academic 'grey' sources of information despite that these sources are known to provide insights into the state of the practice (Garousi et al. 2016). However, due to reliability and validity issues from the grey literature, multivocal literature reviews are suggested when there is low volume and quality of evidence from the academic literature (Garousi et al. 2016). Although the subject area is immature yet, we believe that we covered the majority of the studies with not restricting period of search, constructing a different combination of search strings, and searching on six various most known databases by the two researchers. Since we covered the papers published before 2011 through this

SLR, mapping extracted quality attributes with ISO/IEC 25010:2011 in RQ 3.3 may be a threat.

Internal Validity Identification of primary studies may be a threat in this SLR. To avoid missing related novel research papers, we firstly conducted a string-based database search, and then applied snowballing techniques for selecting the primary studies, and also checked the investigation on different databases. We also used a "control paper" list to check the suitability of search strings. Still, there might be some overlooked studies because they were not on the "control paper" list, and this might have posed a threat. Since there might be a lot of noise in the string-based database search, we decreased this threat by complementing our search process with the snowballing technique. Data extraction step was not conducted mutually exclusive and was performed sequentially by the two authors on the same Excel sheet for all the papers in the pool, which might also have posed a threat to internal validity. To mitigate the bias caused by this threat, each step was discussed so many times, there was constant communication between the two authors and as the results of these discussions, changes were made where necessary.

Construct Validity With this concept, we concern about the validity of the defined research questions and the extracted data. The systematic literature review mainly addresses the selection of the primary studies and how they represent the population of the questions. In order to mitigate this risk, we took different steps, i.e., we analyzed the primary studies included in the final pool qualitatively to answer the research questions. This factor might have caused a threat to the validity of the findings, specifically regarding researcher bias. In the paper review process, several iterations were performed by the researchers to reduce this threat. In addition, a string-based database search was performed on six different electronic databases to avoid potential biases. The number of studies was too small to consider the average rigor among the papers published in a year. Nevertheless, we believe that RQ 5 provided insights to the readers about the maturity of the field.

Reliability This concept focuses on guaranteeing that this review could be repeated identically by other researchers as well. For this purpose, we prepared and examined our SLR protocol. In the data collecting and analyzing stages, there might have been a bias between the two researchers. To mitigate this threat, the researchers attended to analyze and extract required data for each primary study separately, discussed when there was a disagreement with the extractions, and finally made a consensus with the decision; so they provided a reliable degree of objectivity within the findings. However, the researchers' previous knowledge and experience might have caused threats by introducing some subjectivity in some cases. This factor might be a threat to the reliability concept in case of replication of the same research results. In addition, categorization of the quality definitions for RQ 1.1 was made manually, which means there may be quality definitions that might have been overlooked or incompletely categorized.

7 Conclusion and Future Work

In this study, we performed a systematic literature review for software quality of AI-based software. The main goal of this study is searching for "How is quality defined or investigated for the AI-based software?." To answer this research goal, we searched within six different databases for determining the furthest appropriate papers on the topic and selected

29 primary studies. After all, we analyzed the data collected from these studies to answer the 23 research questions under five main headings. This SLR is among the first exhaustive reviews that examined the scope and discussion of quality characteristics and their assurance, challenges, solutions, and existing quality models for the software quality in AI-based software in the period from 1988 to 2020.

Based on the findings, we provided important insights about the challenges of AI-based software by categorizing them with different perspectives such as "Software Quality", "Software Development", "Design", "Social Aspect", and "Testing", traditional or AI-based software quality attributes, quality models, and the domain or type of the software that are subjected to the primary studies. The results of this study are useful for future research in the quality of AI-based software. Based on our discussions, future work will focus on the nature of AI-based software and different parts of these products from the traditional ones; suitability of conventional quality models such as ISO 25010 on AI-based software; evaluating the approaches/techniques observed in this SLR study in common with software professionals; considering quality attributes with AI-based software quality characteristics together for measuring the product quality of such software systems.

Appendix

Table 18 Mapping between each primary study ID, e.g. P1, P2, and the reference to the corresponding paper

Paper ID	Reference to paper
P1	Towards Guidelines for Assessing Qualities of Machine Learning Systems (Siebert et al. 2020)
P2	Software Quality for AI: Where we are now? (Lenarduzzi et al. 2021)
P3	Requirements-Driven Method to Determine Quality Characteristics and Measurements for Machine Learning Software and Its Evaluation (Nakamichi et al. 2020)
P4	Quality Assurance for Machine Learning – an approach to function and system safeguarding (Poth et al. 2020)
P5	Guidelines for Quality Assurance of Machine Learning-based Artificial Intelligence (Hamada et al. 2020)
P6	Engineering problems in machine learning systems (Kuwajima et al. 2020)
P7	Continuous Experimentation on Artificial Intelligence Software: A Research Agenda (Nguyen-Duc and Abrahamsson 2020)
P8	Artificial intelligent environments: Risk management and quality assurance implementation (Malik and Singh 2020)
P9	Testing and Quality Validation for AI Software Perspectives, Issues, and Practices (Tao et al. 2019)
P10	Secure Deep Learning Engineering: a Road towards Quality Assurance of Intelligent Systems (Liu et al. 2019)

Table 18 (continued)

Paper ID	Reference to paper
P11	Priority Quality Attributes for Engineering AI-enabled Systems (Pons and Ozkaya 2019)
P12	Losing Confidence in Quality: Unspoken Evolution of Computer Vision Services (Cummaudo et al. 2019)
P13	Distortion and Faults in Machine Learning Software (Nakajima 2019)
P14	Adapting SQuaRE for Quality Assessment of Artificial Intelligence Systems (Kuwajima and Ishikawa 2019)
P15	Requirements Engineering for Machine Learning: Perspectives from Data Scientists (Vogelsang and Borg 2019)
P16	Non-Functional Requirements for Machine Learning: Challenges and New Directions (Horkoff 2019)
P17	Software Engineering Challenges of Deep Learning (Arpteg et al. 2018)
P18	Quality Assurance of Machine Learning Software (Nakajima 2018)
P19	Open Problems in Engineering and Quality Assurance of Safety Critical Machine Learning Systems (Kuwajima et al. 2018)
P20	Automotive safety and machine learning: Initial results from a study on how to adapt the ISO 26262 safety standard (Henriksson et al. 2018)
P21	Quality Measurement Challenges for Artificial Intelligence Software (Ali)
P22	A Test Architecture for Machine Learning Product (Nishi et al. 2018)
P23	Building a new culture for quality management in the era of the Fourth Industrial Revolution (Hyun Park et al. 2017)
P24	A Practical Study on Quality Evaluation for Age Recognition Systems (Tao et al. 2017)
P25	Software quality in artificial intelligence systems (Vinayagasundaram and Srivatsa 2007)
P26	Mining Software Data (Turhan and Kutlubay 2007)
P27	A Framework for Quality Assurance of Machine Learning Applications (Murphy et al. 2006)
P28	On the Reliability of AI Planning Software in Real-Time Applications (Chen et al. 1995)
P29	Quality Measures and Assurance for AI Software (Rushby 1988)

Table 19 Detailed information per study for RQ 2.2 (addressed challenges of quality) and RQ 4.1 (how these challenges are addressed)

(A) Ref. to Articles	(B) Addressed challenges	(C) Challenge category	(D) Solution
P1	The authors focused on different views of ML systems, and since there are no quality models that can be followed for ML software, they emphasized the need for a guideline for such software systems.	Software Development Software Quality	The authors firstly discussed the definition of the quality attributes with experts. Then, in Fujitsu Laboratories, they conducted three case studies by focusing on requirements. Finally, specific to a use case, they contributed a quality model.
P2	The authors identified the most frequent problems in quality for AI code applying existing SE methodologies, a collection of testing methods feasible to AI-based software systems, and integrated into Machine Learning and Software Engineering courses.	Software Quality Testing Social Aspect	By considering the experience of their groups, they revealed different code quality problems commonly met by all of the stakeholders.
P3	The authors proposed to derive quality characteristics based on requirements specifications for the ML systems.	Software Quality	The authors proposed a quality model, quality characteristics, and a measurement method of ML software adopted into enterprise systems by extending the traditional quality model, ISO 25010.
P4	The authors focused on the way for specifying and estimating quality risks, defining suitable quality assurance activities to mitigate quality risks, ensuring being on the "right track" to generate client trust in the AI-based software at release period, dealing with the non-deterministic manner of ML algorithms.	Software Quality Software Development	The authors mapped ISO25010 with issues taken by experiences/questionnaires from enterprise systems based on literature, and on the ML experts' experience who have worked the development of the evAIa approach.
P5	The authors discussed the quality assurance issues in specific application domains and addressed organizing the general core views of the quality assurance for ML-based systems.	Software Quality, Software Development	The authors examined four popular domains in ML software, and validate the results with a questionnaire survey.

Table 19 (continued)

(A) Ref. to Articles	(B) Addressed challenges	(C) Challenge category	(D) Solution
P6	The authors hypothesized a training process that fastens comprehensible requirements and data-driven training. Moreover, they identified that the lackness of machine learning models (components) characteristics, requirements characterization, design specification, interpretability, and robustness. In order to focus on quality models for ML-based systems, they also discussed the integration of the SQuaRE and the characteristics of machine learning models (components).	Software Quality, Software Development, Design	The authors defined, classified, and searched for the open challenges in safety-critical machine learning systems. They discussed the literature, research directions, and used automated driving vehicles as an example.
P7	1- The authors specified "Quality and Ethical Requirements of AI Software", modeled AI attributes for AI software quality and Ethical issues, traced "AI Quality Implementation", defined MVPs for an AI Model (component), and proposed a toolset "Supporting Continuous Experimentation".	Software Quality	The authors proposed a conceptual model of "continuous experimentation" for AI software, by focusing on agreement and conformity with stakeholders.
P8	The authors made an analytical contribution about managing risks and management strategies for ensuring the quality of AI applications quality.	Software Quality, Software Development	The authors proposed a framework named by Neuro-Fuzzy to avoid disadvantages of the black-box nature of neural nets.
P9	The authors provided their observations on AI software testing for quality assurance.	Software Development	The authors analyzed and discussed the fundamental types of AI software testing and validation attempts focused on the test quality evaluation challenges in AI software and conducted case studies on AI applications.

Table 19 (continued)

(A) Ref. to Articles	(B) Addressed Challenges	(C) Challenge Category	(D) Solution
P10	The authors pinpointed difficulties and future breaks to simplify paying the attention of the SE community towards industrial demand of secure intelligent systems.	Software Quality	Firstly, the authors made a literature review, then updated SDLC for DL systems by taking insights from the literature.
P11	The authors identified four dimensions on which AI engineering most strongly requires to make immediate progress: "building robust systems", "data", "human-machine interaction" and "model (component)".	Software Quality, Software Development	The authors summarized distinctive characteristics of "privacy", "security", "data centricity", "sustainability", and "explainability" since they concern with architecting AI-based software systems for the public sector.
P12	The authors assessed the "consistency", "evolution risk" and "maintenance issues" which may appear while developers using AI services.	Software Quality, Software Development	The authors evaluated the responses of three distinct AI services over 11 months by handling 3 varied data sets, verified responses against the respective documentation, and considered evolution issues.
P13	The authors defined a set of "distortion degrees" that reveal themselves as faults/failures in ML software and examined the properties by considering "neuron coverages."	Software Quality	The authors adopted a hypothesis that distortions in the trained components expose themselves as faults occurring in quality degradation.
P14	The authors presented an analysis on what quality concepts should be evaluated for AI systems.	Software Quality	The authors extended SQUARE and Ethics guidelines.
P15	The authors proposed to understand how ML experts handle "elicitation", "specification", "assurance of requirements", and "expectations".	Software Quality	The authors interviewed data scientists to investigate their perceptions on requirement engineering.
P16	The authors firstly explored and described NFRs for ML software, and used the output of 1, created a catalog of NFRs for ML.	Software Quality	The authors brought non-functional requirements (NFRs) for ML software to the foreground, facilitating "early consideration", "definition", and "trade-off analysis", raising awareness over ML performance.

Table 19 (continued)

(A) Ref. to Articles	(B) Addressed challenges	(C) Challenge category	(D) Solution
P19	The authors defined, classified, and investigated the open challenges of quality for safety-critical machine-learning systems, corresponding industry, and technological trends.	Software Quality	The authors present the open problems and corresponding research/industry trends from the viewpoint of design, verification, and operation for in-vehicle automated-driving systems, subsystems, and machine-learning models (components) using automated-driving vehicles.
P20	The authors conducted an exploratory study to represent the parts of ISO 26262 with the tightest gaps between ML and safety engineering.	Software Quality	The authors discussed the adaptable parts of ISO 26262 for safety-critical ML development in the automotive domain, interview experts to highlight necessary changes to cover "ML training", "model sensitivity", and "test case design".
P22	The authors proposed to show a "Quality Assurance Framework" for ML software.	Software Quality	The authors proposed a policy of QA for ML software, introduced the common principles of QA, and as a part of the policy, they performed ML product assessment. They are composed of A-ARAI: "Allowability", "Achievability", "Robustness", "Avoidability", and "Improvability". They also modeled the "Quality Integrity Level" for ML products. For this purpose, the authors introduced the QA activity levels for the ML product.
P23	The authors discussed the necessity of new concepts for "Quality Management" (QM) and debated the urgency of collaboration of quality experts with data scientists.	Software Quality, Social Aspect	The authors proposed the concept of strategic quality goals. Four different approaches which are "composite dimension", "team creativity", "total inspection", and "new valuation" were also presented in the implementation process for real-world applications.

Table 19 (continued)

(A) Ref. to Articles	(B) Addressed Challenges	(C) Challenge Category	(D) Solution
P24	Current recognition system evaluation techniques fundamentally concentrate on "recognition rate." Nevertheless, existing analysis sometimes focuses on the quality evaluation of "face recognition systems." They examine "accuracy" or the "quality of recognition." To discuss this point, they propose different quality factors for assessment.	Software Quality	The authors presented practical research on a "realistic recognition system" within the recommended quality evaluation approach.
P25	The authors propose to select appropriateness quality attributes for AI systems.	Software Quality	The authors measured functional and nonfunctional requirements by using metrics.
P26	The authors provided a "software quality perspective" on ML and data mining applications.	Software Quality	The authors explain the current research difficulties which are dependent on incorporating the tools established distinct metrics provided from the development processes and the software itself.
P27	The authors focused on testing and debugging ML software.	Testing	The authors described a framework supporting testing and debugging of supervised ML applications.
P28	The authors proposed a new method to evaluate the "reliability" of an AI system.	Software Quality	The authors considered intrinsic defects besides the program bugs, which exist in AI programs that bring heuristic and deadline-violation shortness.
P29	The authors adapted QA and measurements approaches of traditional software systems to the knowledge-based systems.	Software Quality, Software Development	The authors reviewed existing software quality assurance measures and techniques, paid attention to the characteristics of AI-based software, and introduced their evaluation and recommendations for further research.

Table 20 Details of bottom-up approach followed in this SLR for relations of metrics and quality attributes used in primary studies

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P1	Appropriate- ness	Prerequisites for model type.	NA	NA
	Goodness of Fit	Preision, reall, f-sore	NA	NA
	Relevane (Bias-Variane tradeoff)	Variane of ross-validation	NA	NA
	Robustness	Equalized Loss of Auray(ELA).	NA	NA
	Stability	Leave-one-out ross-validation stability.	NA	NA
	Fairness	Equalized odds.	NA	NA
	Interpretability	omplexity metris (e.g., no. of parameters, depth)	NA	NA
	Resoure Utilization	Required storage spae.	NA	NA
	Representative- ness	Statistial tests (e.g., two-sample t-test, et.).	NA	NA
	orretness	Outlier detetion metris (e.g., Z-sore).	NA	NA
	Completeness	No. of missing values.	NA	NA
	Currentness	Age of data.	NA	NA
	Intra-consisteny	Value ranges, word ounts.	NA	NA
	Train/Test Independene	Statistial tests (e.g., two-sample t-test, et.).	NA	NA
	Balanedeness	Ratio of lasses.	NA	NA
	Absene of bias	Ratios of groups.	NA	NA
	Inter-onsisteny	Value ranges, rosswise outlier detetion metris.	NA	NA
	Environmental impat	Energy onsumption.	NA	NA
	Soial impat	Impat on employees.	NA	NA
	Sope ompliane	Value ranges, novelty detetion metris	NA	NA
	Effetiveness	False positive/ negative detetion rate.	NA	NA
	Supervision Overhead / Effiieny	Time, memory used	NA	NA
	Infrastruture Suitability	omputational and storage apabilities.	NA	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P3	Functional Correctness	Model Accuracy	"A measure of the degree of accuracy of a trained model."	NA
	Functional Correctness	Achievement Of Model Accuracy (Amodel)	"A measure of the degree to which the Model Accuracy is achieved."	"Amodel=Rmodel _ Accuracy/Emodel _ Accuracy where Rmodel _ Accuracy =measured accuracy andEmodel _accuracy=target accuracy"
	Functional Correctness	MLS Accuracy	"A measure of the accuracy of the output during MLS operation. "	NA
	Functional Correctness	Achievement OfMLS Accuracy (Amls)	"A measure of achieving accuracy as the rate at which the model answers correctly"	"Amls=Rmls _Accuracy/ Emlsl _Accuracy where Rmls _Accuracy =measured accuracy and Emls _Accuracy=target accuracy"
	Functional Correctness	Consistency OfMLS OutputData Trend	"A measure of the degree to which the statistical trend of the MLS output data set is consistent with the required trend."	NA
	Functional Correctness	Achievement OfAverage ValueOf MLSOutput Data (Vmls)	"The degree to which the average value of the measured MLS output data meets the target average value of the MLS output data."	"Vmls=Rmls _Average/Emlsl _Averagewhere Rmls _Average =measured average (mean)and Emls _average=target average (mean)"
	Maturity	Sufficiency Of Input DataQuality Measurement Means	"This is a measure of the appropriateness of the quality measures of input data".	NA
	Maturity	Sufficiency OfMLS OutputData Quality Measurement- Means	"This is a measure of the appropriateness of the quality measuring functions for MLS output data."	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P3	Maturity	Runtime Model Accuracy	"This is a measure of the degreeof accuracy of a trained model during operation."	NA
	Maturity	Achievement Of Runtime Accuracy	"This is a measure of the accuracy of the runtime model."	NA
	Maturity	Consistency Of Input Data Trend	"This is a measure of the degree to which the statistical trend of the input data matches the expected trend."	NA
	Maturity	Achievement Of Average Value Of InputData	"The degree to which the average value of the measureddata matches the target average value of the input data."	NA
P6	Functional suitability	Number of functions that are incorrect	NA	NA
	Functional suitability	Number of functions missing or incorrect among those that are required for achievingMa specific usage objective	NA	NA
	Functional suitability	Number of functions that are incorrect	NA	NA
	Functional suitability	Number of functions missing or incorrect among those that are required for achieving a specific usage objective	NA	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P6	Usability	Number of functions having state monitoring capability	NA	NA
	Reliability	Number of avoided critical and serious failure occurrences based on , test cases	NA	NA
	Reliability	Number of system/software failures actually occurred	NA	NA
	Reliability	Number of failures detected during observation time	NA	NA
	Reliability	Number of system components redundantly installed	NA	NA
	Maintainability	Number of test functions required	NA	NA
	Maintainability	Number of components which are implemented with no impact on others	NA	NA
	Maintainability	Expected time for making a specific type of modification	NA	NA
	Maintainability	Number of diagnostic functions useful for causal analysis	NA	NA
	Portability	Number of functions which were tested in different operational environments	NA	NA
	Portability	Number of functions which produce similar results as before	NA	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P14	Functional suitability	Number of functions missing	NA	NA
	Functional suitability	Number of functions specified	NA	NA
	Maintainability	Number of test functions implemented as specified	NA	NA
	Maintainability	Number of test functions required	NA	NA
P24	NA	Face recognition rate	"Recognition rate is affected by a number of external factors, such as the angle of face, the size of face, and multi-faces in one image."	"R1=NP/N where NP refers to the passed recognition cases and N refers to the total cases that have faces without image quality problems."
	NA	Group recognition hit rate.	"It refers to the ratio of the recognition cases that in the correct age group."	"R2=NH/N where NH refers to the cases that hit the correct age group, and N refers to the total cases that pass the recognition system."
	NA	Recognition absolute accuracy rate.	"It describes the ratio of the cases that the recognition age equals to the true age."	"R3=NC/N where NC means the cases that the recognition age is equal to the actual age and N refers to the total cases that pass the recognition system."
	NA	The recognition rate based on absolute error.	"It refers to that the error is divided with a fixed real number."	"R4=COUNT(abs(age1-age2)¡e)/N where age1 means the real age, age2 is the recognition age, e represents the fixed real number threshold"
	NA	The recognition rate based on relative error.	"This index shows that the error is divided based on the percentage of actual age."	"R5=COUNT(abs(age1-age2)/age1¡e)/N where age1 means the real age, age2 means the recognition age, e means the percentage, and N refers to the total cases that pass the recognition system."

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
	NA	Average absolute error.	"It refers to the average absolute value of the error between real age and recognition age."	"R5= Σ abs(age1-age2)/N where age1 means the real age, age2 means the recognition age, and N refers to the total cases that pass the recognition system".
	NA	Average relative error.	NA	"R6=(Σ(abs(age1-age2)/age1))/N where age1 means the real age, age2 means the recognition age, and N refers to the total cases that pass the recognition system."
P24	NA	Age recognition variance.	"The absolute error and relative error of recognition cannot reflect the overall performance of the age recognition system."	"R7=Σ((age1-age2)*(age1-age2))/N where age1 means the real age, age2 means the recognition age, and N refers to the total cases that pass the recognition system."
P25	Usability	Number of lines (LOC)	"Counts the number of executable statements per component"	NA
	Testability, Usability, Readability	Cyclomatic complexity	"Measures the logical complexity of a program. "	NA
	Testability, Readability	Max levels of nesting	"Measures the max number of nesting in the control structure of a component"	NA
	NA	Number of paths	"Counts the number of noncyclic paths per component"	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
	NA	Unconditional jumps	"Counts the number of occurences of GOTO"	NA
	Self descriptiveness	Ratio of comment statements	"Defined as the propostion of comment lines to number of executable statements"	NA
	NA	Vocabulary frequency	"Sum of the number of the unique operands and operators that are necessary for the definition of component	NA
	Readability	Program length	"Sum of the number of occurences of the unique operands and operators"	NA

Table 20 (continued)

Paper ID	Quality attribute (QA)	Metric	Metric definition	Metric formula
P25	Usability, Readability	Average size	"Measures the average statement size of the component"	Program length/LOC
	Testability	Number of inputs/outputs	"Count the number of input and exit points of the component"	NA
P29	Reliability	Number of rules	NA	NA
	Reliability	Number of statements	NA	NA

Declarations

Conflicts of Interests/Competing interests Please find attached the paper, "Systematic Literature Review on Software Quality for AI-based Software" by Bahar Gezici and Ayça Kolukısa Tarhan, which we would like to submit for possible publication to the Empirical Software Engineering. We confirm that this work is original and has not been published elsewhere nor is it currently under consideration for publication elsewhere.

For any information concerning this manuscript, please contact me preferably by e-mail at bahargezici@cs.hacettepe.edu.tr. Thank you for your consideration of this manuscript.

References

25000 I (2005) The iso/iec 25000 series of standards. https://iso25000.com/index.php/en/iso-25000-standards

25012:2008 I (2008) software engineering — software product quality requirements and evaluation (square) — data quality model. https://www.iso.org/standard/35736.html

26262-1:2018 I (2018) Road vehicles — functional safety. https://www.iso.org/standard/68383.html

29119-1:2013 I (2013) Software and systems engineering — software testing. https://www.iso.org/standard/45142.html

9126-1:2001 I (2001) Software engineering — product quality. https://www.iso.org/standard/22749.html

Aggarwal A, Lohia P, Nagar S, Dey K, Saha D (2019) Black box fairness testing of machine learning models. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 625–635

Alamin MAA, Uddin G (2021) Quality assurance challenges for machine learning software applications during software development life cycle phases. arXiv:2105.01195

Ali Z Quality measurement challenges for artificial intelligence software

de Almeida Biolchini JC, Mian PG, Natali ACC, Conte T, Travassos GH (2007) Scientific research ontology to support systematic review in software engineering. Adv Eng Inform 21(2):133–151

Arpteg A, Brinne B, Crnkovic-Friis L, Bosch J (2018) Software engineering challenges of deep learning. In: 2018 44Th euromicro conference on software engineering and advanced applications (SEAA). IEEE, pp 50–59

Borg M, Englund C, Wnuk K, Duran B, Levandowski C, Gao S, Tan Y, Kaijser H, Lönn H, Törnqvist J (2018) Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry. arXiv:1812.05389

Bosch J, Olsson HH, Crnkovic I (2021) Engineering ai systems: a research agenda. In: Artificial intelligence paradigms for smart cyber-physical systems. IGI Global, pp 1–19

Bourque P, Dupuis R, Abran A, Moore JW, Tripp L (2004) Guide to the software engineering body of knowledge -

Braiek HB, Khomh F (2020) On testing machine learning programs. J Syst Softw 164:110542

Byrne C (2017) Development Workflows for Data Scientists. O'Reilly Media

Chen R, Bastani FB, Tsao TW (1995) On the reliability of ai planning software in real-time applications. IEEE Trans Knowl Data Eng 7(1):4–13

Cummaudo A, Vasa R, Grundy J, Abdelrazek M, Cain A (2019) Losing confidence in quality: Unspoken evolution of computer vision services. In: 2019 IEEE International conference on software maintenance and evolution (ICSME). IEEE, pp 333–342

Deng L (2018) Artificial intelligence in the rising wave of deep learning: The historical path and future outlook [perspectives]. IEEE Signal Proc Mag 35(1):180–177

Forward A, Lethbridge TC (2008) A taxonomy of software types to facilitate search and evidence-based software engineering. In: Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds, pp 179–191

Garousi V, Felderer M, Mäntylä MV (2016) The need for multivocal literature reviews in software engineering: complementing systematic literature reviews with grey literature. In: Proceedings of the 20th international conference on evaluation and assessment in software engineering, pp 1–6

Geske F, Hofmann P, Lämmermann L, Schlatt V, Urbach N (2021) Gateways to artificial intelligence: Developing a taxonomy for ai service platforms. In: Twenty-ninth european conference on information systems (ECIS)

Gezici B, Tarhan AK (2019) Final pool. https://drive.google.com/file/d/1ve6BpJTrIT sfo6auSoWKh48ajWbNb05n/view?usp=sharing
Hamada K, Ishikawa F, Masuda S, Matsuya M, Ujita Y (2020) Guidelines for quality assurance of machine learning-based artificial intelligence. In: SEKE2020: The 32nd international conference on software engineering & knowledge engineering, pp 335–341
Hannousse A (2021) Searching relevant papers for software engineering secondary studies: Semantic scholar coverage and identification role. IET Softw 15(1):126–146
Henriksson J, Borg M, Englund C (2018) Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard. In: 2018 IEEE/ACM 1St international workshop on software engineering for AI in autonomous systems (SEFAIAS). IEEE, pp 47–49
Hopgood AA (2005) The state of artificial intelligence. Adv Comput 65:1–75
Horkoff J (2019) Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27Th international requirements engineering conference (RE). IEEE, pp 386–391
Hyun Park S, Seon Shin W, Hyun Park Y, Lee Y (2017) Building a new culture for quality management in the era of the fourth industrial revolution. Total Qual Manag Bus Excell 28(9-10):934–945
Ishikawa F, Yoshioka N (2019) How do engineers perceive difficulties in engineering of machine-learning systems?-questionnaire survey. In: 2019 IEEE/ACM Joint 7th international workshop on conducting empirical studies in industry (CESI) and 6th international workshop on software engineering research and industrial practice (SER&IP). IEEE, pp 2–9
ISO/IEC (2011) Iso/iec 25010 (2011)-systems and software quality requirements and evaluation (square)-system and software quality models. International Standard ISO/IEC 25010 2(1):1–25
Ivarsson M, Gorschek T (2011) A method for evaluating rigor and industrial relevance of technology evaluations. Empir Softw Eng 16(3):365–395
Kitchenham B (2004) Procedures for performing systematic reviews. keele, UK. Keele Univ 33(2004):1–26
Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering -
Kuwajima H, Ishikawa F (2019) Adapting square for quality assessment of artificial intelligence systems. In: 2019 IEEE International symposium on software reliability engineering workshops (ISSREW). IEEE, pp 13–18
Kuwajima H, Yasuoka H, Nakae T (2018) Open problems in engineering and quality assurance of safety critical machine learning systems. arXiv:1812.03057
Kuwajima H, Yasuoka H, Nakae T (2020) Engineering problems in machine learning systems. Mach Learn 109(5):1103–1126
Lakshen GA, Vraneš S., Janev V (2016) Big data and quality: A literature review. In: 2016 24Th telecommunications forum (TELFOR). IEEE, pp 1–4
Lenarduzzi V, Lomio F, Moreschini S, Taibi D, Tamburri DA (2021) Software quality for ai: Where we are now? In: International conference on software quality. Springer, pp 43–53
Liu Y, Ma L, Zhao J (2019) Secure deep learning engineering: a road towards quality assurance of intelligent systems. In: International conference on formal engineering methods. Springer, pp 3–15
Lwakatare LE, Raj A, Crnkovic I, Bosch J, Olsson HH (2020) Large-scale machine learning systems in real-world industrial settings: a review of challenges and solutions. Inf Softw Technol 127:106368
Malik V, Singh S (2020) Artificial intelligent environments: risk management and quality assurance implementation. J Discret Math Sci Cryptogr 23(1):187–195
Mannarswamy S, Roy S, Chidambaram S (2020) Tutorial on software testing & quality assurance for machine learning applications from research bench to real world. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp 373–374
Martínez-Fernández S, Bogner J, Franch X, Oriol M, Siebert J, Trendowicz A, Vollmer AM, Wagner S (2021) Software engineering for ai-based systems: A survey. arXiv:2105.01984
Masuda S, Ono K, Yasue T, Hosokawa N (2018) A survey of software quality for machine learning applications. In: 2018 IEEE International conference on software testing, verification and validation workshops (ICSTW). IEEE, pp 279–284
Murphy C, Kaiser GE, Arias M (2006) A framework for quality assurance of machine learning applications -
Nakajima S (2018) Quality assurance of machine learning software. In: 2018 IEEE 7Th global conference on consumer electronics (GCCE). IEEE, pp 601–604
Nakajima S (2019) Distortion and faults in machine learning software. In: International workshop on structured object-oriented formal language and method. Springer, pp 29–41
Nakamichi K, Ohashi K, Namba I, Yamamoto R, Aoyama M, Joeckel L, Siebert J, Heidrich J (2020) Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In: 2020 IEEE 28Th international requirements engineering conference (RE). IEEE, pp 260–270

Nascimento E, Nguyen-Duc A, Sundbø I, Conte T (2020) Software engineering for artificial intelligence and machine learning software: A systematic literature review. arXiv:2011.03751
Nguyen-Duc A, Abrahamsson P (2020) Continuous experimentation on artificial intelligence software: a research agenda. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 1513–1516
Nishi Y, Masuda S, Ogawa H, Uetsuki K (2018) A test architecture for machine learning product. In: 2018 IEEE International conference on software testing, verification and validation workshops (ICSTW). IEEE, pp 273–278
Ongsulee P (2017) Artificial intelligence, machine learning and deep learning. In: 2017 15Th international conference on ICT and knowledge engineering (ICT&KE). IEEE, pp 1–6
Petersen K, Vakkalanka S, Kuzniarz L (2015) Guidelines for conducting systematic mapping studies in software engineering: an update. Inf Softw Technol 64:1–18
Pons L, Ozkaya I (2019) Priority quality attributes for engineering ai-enabled systems. arXiv:1911.02912
Poth A, Meyer B, Schlicht P, Riel A (2020) Quality assurance for machine learning–an approach to function and system safeguarding. In: 2020 IEEE 20Th international conference on software quality, reliability and security (QRS). IEEE, pp 22–29
Rahman MS, Reza H (2020) Systematic mapping study of non-functional requirements in big data system. In: 2020 IEEE International conference on electro information technology (EIT). IEEE, pp 025–031
Riccio V, Jahangirova G, Stocco A, Humbatova N, Weiss M, Tonella P (2020) Testing machine learning based systems: a systematic mapping. Empir Softw Eng 25(6):5193–5254
Rushby J (1988) Quality measures and assurance for AI software, vol 18. National Aeronautics and Space Administration, Scientific and Technical Information Division
Russel S, Norvig P (2009) Artificial intelligence: a modern approach, English
Samoili S, Cobo ML, Gomez E, De Prato G, Martinez-Plumed F, Delipetrev B (2020) Ai watch. defining artificial intelligence. towards an operational definition and taxonomy of artificial intelligence. In: JRC Technical reports. Joint research centre (seville site)
Siebert J, Joeckel L, Heidrich J, Nakamichi K, Ohashi K, Namba I, Yamamoto R, Aoyama M (2020) Towards guidelines for assessing qualities of machine learning systems. In: International conference on the quality of information and communications technology. Springer, pp 17–31
Taleb I, Serhani MA, Dssouli R (2018) Big data quality: a survey. In: 2018 IEEE International congress on big data (bigdata congress). IEEE, pp 166–173
Tao C, Gao J, Wang T (2019) Testing and quality validation for ai software–perspectives, issues, and practices. IEEE Access 7:120164–120175
Tao C, Hao C, Gao J, Wang T, Wen W (2017) A practical study on quality evaluation for age recognition systems. In: SEKE, pp 345–350
Tsintzira AA, Arvanitou EM, Ampatzoglou A, Chatzigeorgiou A (2020) Applying machine learning in technical debt management: Future opportunities and challenges. In: International conference on the quality of information and communications technology. Springer, pp 53–67
Turhan B, Kutlubay O (2007) Mining software data. In: 2007 IEEE 23Rd international conference on data engineering workshop. IEEE, pp 912–916
Vinayagasundaram B, Srivatsa S (2007) Software quality in artificial intelligence system. Inf Technol J 6(6):835–842
Vogelsang A, Borg M (2019) Requirements engineering for machine learning: Perspectives from data scientists. In: 2019 IEEE 27Th international requirements engineering conference workshops (REW). IEEE, pp 245–251
Wan Z, Xia X, Lo D, Murphy GC (2019) How does machine learning change software development practices? IEEE Transactions on Software Engineering
Wieringa RJ (2014) Design science methodology for information systems and software engineering. Springer
Wohlin C (2014) Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering, pp 1–10
Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer Science & Business Media
Zhang JM, Harman M, Ma L, Liu Y (2020) Machine learning testing: survey, landscapes and horizons. IEEE Transactions on Software Engineering
Zhang P, Cao W, Muccini H (2020) Quality assurance technologies of big data applications: A systematic literature review. arXiv:2002.01759