IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 September 2023, accepted 13 November 2023, date of publication 16 November 2023, date of current version 27 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333920

SURVEY

Taxonomy of Quality Assessment for Intelligent Software Systems: A Systematic Literature Review

AHROR JABBOROV[1], ARINA KHARLAMOVA[1], ZAMIRA KHOLMATOVA[1], ARTEM KRUGLOV[1], VASILY KRUGLOV[2], AND GIANCARLO SUCCI[3], (Member, IEEE)

[1]Lab of Industrializing Software Production, Innopolis University, 420500 Innopolis, Russia
[2]Faculty of Information Technology and Automatics, Institute of Radioelectronics and Information Technology, Ural Federal University, 620075 Yekaterinburg, Russia
[3]Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy

Corresponding author: Artem Kruglov (a.kruglov@innopolis.ru)

This work was supported by the Russian Science Foundation under Grant 22-21-00494.

ABSTRACT The increasing integration of AI software into various aspects of our daily lives has amplified the importance of evaluating the quality of these intelligent systems. The rapid proliferation of AI-based software projects and the growing reliance on these systems underscore the urgency of examining their quality for practical applications in both industry and academia. This systematic literature review delves into the study of quality assessment metrics and methods for AI-based systems, pinpointing key attributes and properties of intelligent software projects that are crucial for determining their quality. Furthermore, a comprehensive analysis of this domain will enable researchers to devise novel methods and metrics for effectively and efficiently evaluating the quality of such systems. Despite its importance, this area of development is still relatively nascent and evolving. This paper presents a systematic review of the current state of the taxonomy of quality assessment for AI-based software. We analyzed 271 articles from six different sources that focused on the quality assessment of intelligent software systems. The primary objective of this work is to provide an overview of the field and consolidate knowledge, which will aid researchers in identifying additional areas for future research. Moreover, our findings reveal the necessity to establish remedial strategies and develop tools to automate the process of identifying appropriate actions in response to abnormal metric values.

INDEX TERMS Artificial intelligence, machine learning, intelligent systems, AI-based software, software attributes, quality assessment, feature selection, quality models, AI system evaluation.

I. INTRODUCTION

Artificial Intelligence (AI) has rapidly emerged as one of the most prominent areas of research and development in recent years [1], [2]. AI systems are increasingly being integrated into our daily lives, playing a vital role in various applications spanning healthcare, finance, transportation, education, and entertainment [3], [4]. As an example, ChatGPT had around 1 million users within 5 days of its availability [1]. As AI systems become more ubiquitous, ensuring their high quality is crucial, as even minor flaws in AI software can lead to significant consequences [5].

Given the critical role AI systems play in our lives, thorough assessment and improvement of AI software quality are essential. However, evaluating the quality of AI software is a complex and challenging task. Unlike traditional software systems, AI systems are often designed to learn and adapt over time, making the quality of their outputs less easily quantifiable [5]. Furthermore, the social and ethical implications of AI systems must be considered when assessing their quality, as the impact of these systems on society and the environment is becoming increasingly significant.

Software engineering is now an industry experiencing tremendous growth, with the demand for software projects soaring. Has this dramatic rise in software engineering

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

affected the quality of the software code? Software quality assessment involves testing and evaluating various software components using current techniques, benchmarks, and quantifiable characteristics. This process is vital for software development and management, as it informs project managers about the product's state, enabling them to allocate resources and funds more effectively. However, quality evaluation for AI-based systems differs from traditional software quality measurement, as machine learning models often possess unique properties that cannot be checked by running these models on a test data [6]. The performance of the machine learning model and the system's overall performance are strongly and positively correlated.

Intelligent software systems are gaining popularity within the computer science community. These systems unobtrusively integrate computer and networking technologies to surround their human users [7], [8]. The goal of this technology is to provide users with meaningful information and take actions to improve their environment. As the number of users grows, the resources enabling intelligence may quickly become saturated, leading the system to become unstable and giving users the impression that the assessments are of low quality.

Intelligent systems are currently a novel approach lacking a comprehensive, quantifiable understanding of the technology. This study offers suggestions for assessment criteria for such systems and provides an overview of potential challenges in software measurement of intelligent systems. We need measures to assess the current state of science and technology and to weigh the advantages and disadvantages of different systems. Without metrics, we cannot recognize success and penalize failure. Metrics are numerical measurements used to assess specific quality attributes of a utility or software [9]. Moreover, without quantifiable measures, science and engineering become impossible. Using metrics for software developed using object-oriented and procedural approaches is challenging, but it becomes even more complex and sophisticated when applied to software agents exhibiting anthropomorphic qualities. This is because the domain of intelligent systems spans multiple academic fields, including control theory, neural networks, artificial intelligence, and cognitive sciences [10].

Developments in computer technology over the past several decades have enabled the storage of vast amounts of data electronically. The demand for computer software to incorporate more intelligent capabilities is driven primarily by two compelling arguments. The first justification concerns the existing bottleneck in data processing. Emphasis on storage efficiency rather than processing significance was based on historical manual information handling practices and the implicit acknowledgment that computer operators of computer-based data storage devices must analyze data into knowledge and information [11]. Data file and database management approaches often focused on the processes of storing, retrieving, and manipulating data outputs, rather than the context in which the collected data would be used for planning, monitoring, evaluation, and decision-making. The second justification has a slightly different nature, relating to the complexity of communication systems and networking computers, and how organizations increasingly rely on the dependability of such IT environments as a critical component of their efficiency, effectiveness, and survival.

Data-centered applications that only incorporate their data environment represent the initial stage in the evolution of intelligent systems [12]. Subsequent stages involve ontology-based applications with computational intelligence capabilities. It is argued that under a broader concept of intelligence, a distinction between component capabilities and human intelligence can be established, allowing the former to be integrated into software. The primary driving force behind the pursuit of intelligent software has been the growing recognition of the significant role played by information and data, rather than the application's functionality and logic.

To evaluate the relative capacities of software systems along their evolutionary trajectory over the past 50 years, Rahman et al. [13] employ an assessment framework with six categories, each featuring suitable attributes or capabilities that serve as a set of evaluation criteria. To assess the degree of intelligence to which a software program or system can execute intelligent activities, each domain's individual characteristics or capacities are ordered in ascending order of complexity.

This paper presents a systematic review of the current state of the taxonomy of quality assessment for AI-based software. The main goal of this work is to study the field's landscape and consolidate knowledge to help researchers identify further areas of research. Our research is aimed to investigate the following aspects of quality assessment of AI-based software:

1) existing approaches;
2) measurable attributes, statistical and machine learning models used to estimate quality;
3) effectiveness of the found attributes and models.

The primary contributions of this work include providing insight into the defined research area through a comprehensive analysis of existing literature and producing a systematic summary.

The paper is structured as follows: Section II describes the protocol used in this research; Section III illustrates the findings of our work; Section IV offers an interpretation of our findings in relation to the research questions characterizing this systematic literature review and provides their critical interpretation; Section V assesses the constraints and numerous shortcomings that may impact the research, while Section VI addresses the questions related to the quality of this work. Section VII draws some conclusions and identifies areas of further research.

II. PROTOCOL DEVELOPMENT

To present a comprehensive picture of quality for AI-based software, we conducted a systematic literature analysis following the approach by Kitchenham [14]. Systematic literature studies in software engineering can include reviews and syntheses of prior work, enabling researchers to gain an understanding of the state of a particular research subject. The findings of this SLR research should provide a more comprehensive picture of the gaps and support for software quality in the context of AI-based software systems. The following sections conduct key processes such as literature selection, string searches, and data extraction techniques in accordance with the guidelines of Peterson [15].

A. RESEARCH QUESTIONS

The initial step in conducting a literature review is to identify a set of individual research questions. These questions guide the development of this work and inform the readers about the main focus of the study. To formulate the most appropriate research questions, we referred to the "Goal Question Metric" model developed by Caldiera et al. [16]. With this model, it is necessary to predetermine the analysis's objectives, target objects and issues, as well as the analytical vantage points. According to the model, we specified the purpose, target objectives, issues, and viewpoints of the analysis as follows:

- Purpose - Systematic literature review.
- Objective - Peer-reviewed research papers in computer science and software engineering.
- Issue - Taxonomies for evaluating the quality of intelligent software systems.
- Viewpoint - Software engineers and industry practitioners.

This SLR aims to answer the general question, "How is quality defined or studied for AI-based software?" Using the formulated GQM model above, this review focused on AI-based software attributes and attempted to find answers to the following research questions:

RQ1: What are the existing approaches for assessing the quality of software systems for artificial intelligence?

RQ2: Which measurable attributes of software systems for artificial intelligence, and which statistical or machine learning models are commonly used for estimating the quality of such systems?

RQ3: How effective are such attributes and models?

The motivation for RQ1 was to understand what existing research has produced in terms of general approaches to evaluate the quality of AI-based software projects originating from open-source software project sources.

The motivation for RQ2 was to identify which particular subsets of measurable software attributes are used to define the ongoing status of intelligent systems and which methods are commonly used for the quality assessment of such systems.

The motivation for RQ3 was to rank such methods, approaches, and attributes in terms of effectiveness and reliability.

B. LITERATURE SEARCH

This section outlines the search process, including the use of predefined keywords and queries to generate literature search results. Appropriate keywords are formulated in Section II-B1. The formulated keywords are then used to search for publications in databases mentioned in Section II-B2. Subsequently, in Section II-B3, the keywords are combined using Boolean operators to formulate search queries in a way that increases the precision of generating more potentially relevant papers from the selected resources. The initial results are then passed through the inclusion and exclusion criteria introduced in Section II-C to remove all irrelevant publications. After this step, the reading log is finalized in Section III-A1 and evaluated for their qualities using appropriate questions presented in Section II-D.

1) SEARCH KEYWORDS

The initial step of the search process involves extracting topic-related keywords from the research questions mentioned earlier.

Table 1 show a large number of results found by using the extracted keywords. Thus, in Section II-B3, these keywords are used to generate appropriate search queries to increase the precision of the search process.

2) SEARCH RESOURCES

The next step involves selecting some of the most popular literature databases in the field. The selected databases are:

- ACM Digital Library
- Google Scholar
- Scopus
- ScienceDirect
- IEEExplore
- The Lens

These databases are chosen based on their size and popularity in the Computer Science and Software Engineering fields.

3) SEARCH QUERIES

The predefined keywords are then arranged with boolean operators to form potentially effective search queries that help extract relevant works from the resources. The following search query has been selected for conducting the main literature search:

```
("AI software" OR
"artificial intelligence software" OR
"machine learning software" OR
"intelligent systems software" OR
"AI-based software" OR "AI-based system")
AND ("quality assessment" OR
"quality assurance")
```

The setting for using the query focused on the All-Text format, which allows fetching more papers to be examined. The reason for selecting this particular setting was to collect as many related papers as possible and then manually go

TABLE 1. Preliminary results on search databases.

Preliminary queries	ACMDL	GS	Scopus	ScienceDirect	IEEExplore	The Lens
"software projects"	6,506	174,000	58,453	8,951	2,593	22,654
"artificial intelligence"	122,592	3,060,000	3,013,647	171,515	328,456	837,779
"machine learning"	1102,842	3,570,000	1,386,737	196,982	127,184	556,779
"AI-based software"	37	1,650	160	156	8	0
"intelligent systems"	10,162	1,450,000	736,648	38,795	57,223	131
"quality assessment"	3,621	2,560,000	319,195	87,769	10,365	115,287
"quality assurance"	7,219	2,780,000	352,343	143,977	5,826	196,518
"software metrics"	2,044	54,500	18,420	2,181	6,026	5,311
"measurable properties"	186	21,300	670	3,397	35	2,114
"quality metrics"	3,422	130,000	36,919	17,419	2,516	20,518
"AI software quality"	2	23	3	2	3	2

through each one of them to gain more knowledge in this relatively new field of research.

The results of the preliminary search are presented in Table 1. The formulated main query is used to conduct the main search on the databases specified in II-B2 with the ‘‘All Text’’ search method if available. The results are presented in section III-A1 and in Table 4.

C. INCLUSION AND EXCLUSION CRITERIA

To ensure the identification of the most relevant and high-quality literature for this systematic literature review, a comprehensive set of inclusion and exclusion criteria were established. These criteria were designed to filter out irrelevant or low-quality papers, and focus only on the publications that could provide valuable insights and contribute to a better understanding of quality evaluation in AI-driven software projects.

Inclusion criteria that were used are:

- IC1 The paper is written in English, ensuring accessibility to a wide range of readers and researchers.
- IC2 The paper is peer-reviewed and published by a reputable publisher (indexed by databases such as CMDL, Google Scholar, Scopus, ScienceDirect, IEEE-Xplore, The lens), ensuring a certain level of quality and academic rigor.
- IC3 The paper provides a comprehensive taxonomy, framework, or discussion of the most effective attributes for quality evaluation of software projects in the context of artificial intelligence.
- IC4 The paper presents quantitative measurements or empirical evidence supporting the validity and effectiveness of the proposed quality attributes or metrics in evaluating intelligent software properties.

Exclusion criteria that were used are:

- EC1 The paper includes a duplicate work by the same author(s) or is a reiteration of previously published research without significant additional contributions.
- EC2 The paper is an abstract, poster, summary, keynote, opinion piece, editorial, short paper, book chapter, or any other format that does not provide substantial information or insights related to the topic.
- EC3 The paper does not present any type of experimentation, comparison, empirical analysis, or results that could support the claims made by the authors.
- EC4 The paper is not directly related to software engineering or quality assessment in the context of artificial intelligence, machine learning, or intelligent systems.
- EC5 The paper fails to provide clear descriptions, justifications, or explanations for the selected metrics, making it difficult to assess the validity and usefulness of the proposed quality attributes.

By applying these inclusion and exclusion criteria to the initial search results, the final reading log can be populated with high-quality, relevant publications that will contribute significantly to the systematic literature review and help develop a comprehensive understanding of quality assessment in AI-driven software projects.

D. QUALITY ASSESSMENT

The quality of the publications included in this systematic literature review plays a vital role in ensuring the reliability and validity of the findings derived from them. To objectively assess the quality of the selected publications, a set of predefined questions was formulated, and a scoring system was established. Each question was assigned a score of 1, 0.5, or 0 based on whether the answer was ‘yes,’ ’partial,’ or ‘no,’ respectively. The partial score was introduced to accommodate potential human error and reduce subjectivity in the assessment process. The questions used for quality assessment are presented in Table 2, and the corresponding weights and descriptions for each question are provided in Table 3. The final quality scores for each paper can be found in Appendix. After scoring the publications, the classification of the papers based on their quality is presented in Section III-B1.

By implementing a structured and objective quality assessment process, this systematic literature review aims

TABLE 2. Questions - quality assessment.

QA1	Were the objectives / goals and the research questions clearly specified?
QA2	Was the research process transparent and reproducible?
QA3	Were the results evaluated critically and comprehensively?
QA4	Were the outcomes of the research clearly presented and discussed?

to provide a comprehensive and reliable synthesis of the most relevant and high-quality publications in the field. This approach ensures that the findings and conclusions drawn from the reviewed literature are robust and trustworthy, thereby contributing to the overall validity of the research.

III. RESULTS

In this section, we present the results derived from an analysis of all papers included in the reading log. After a meticulous selection process, a total of 38 papers were chosen for further examination (refer to Appendix). To minimize potential human error, the inclusion and exclusion processes were rigorously assessed.

Given the results of the selection process, it is evident that the current field is still in its nascent stages, with a limited number of relevant publications available in academia. In the subsequent subsections, we provide a statistical overview of the obtained results.

A. PRELIMINARY CLUSTERING AND ANALYSIS

In this systematic literature review (SLR), we focused on examining the most reputable and reliable databases available to researchers, as previously mentioned. It is important to note that we allocated publications to specific databases based on the chronological order of the searches conducted, even though some papers might be present across multiple databases.

1) SEARCH RESULTS BY SOURCES

In this subsection, we detail the search process employed. We utilized the search queries formulated in section II-B3 to retrieve results from the databases specified in section II-B2, and compiled them into the search results shown. Subsequently, duplicates from different sources were removed. The results were then subjected to the inclusion and exclusion criteria, as outlined in section II-C, to be considered for the final reading log. The primary search results from each source are presented in Table 4. After finalizing the reading log, we conducted a quality assessment in section II-D, and the results are displayed in Appendix.

The distribution of the publications included in our final reading log by the search sources is presented in Table 5 and Figure 2. As one can notice over 60% of the publications in our reading log were indexed in two databases, ACMDL and Science Direct.

TABLE 3. Weights - quality assessment.

Question	Answer	Score	Description
QA1	Yes	1	if the objectives and research questions were explicitly stated
	Partial	0.5	if the goals of the paper and its research questions were sufficiently clear but could be improved
	No	0	if no objectives were stated, if the research questions were hard to determine, or if they didn't relate to the research being carried out
QA2	Yes	1	if the paper specified the the methodology as well as the technologies used and the data gathered, or if all the necessary steps and sources needed to reproduce the research were transparently available to the reader
	Partial	0.5	if minor details were lacking (for example, a dataset is not readily available)
	No	0	if it was impossible to restore the sequence of actions, or if other critical details (such as an algorithm or technologies used) were missing
QA3	Yes	1	if the authors of the paper provided a critical, balanced, and fair analysis of their results
	Partial	0.5	if the results were only partly (sufficiently) scrutinized and a comprehensive critical analysis was missing
	No	0	if the authors did not evaluate their results
QA4	Yes	1	if the results provided evidence for the conclusion, or if the conclusion was logical and sound
	Partial	0.5	if the results could only partially justify the conclusion
	No	0	if the conclusion was overstated or if it couldn't be justified by the results presented in the paper

TABLE 4. Papers selection. The table shows the procedure through which potentially relevant papers were screened out through the adoption of IC and EC criteria. The number of papers included in the final reading log is shown in the column "Selected papers."

Source	Initial selection	Potentially relevant	Removed papers									Selected papers
			IC1	IC2	IC3	IC4	EC1	EC2	EC3	EC4	EC5	
ACMDL	107	80	1	0	3	0	0	5	0	18	47	6
Google Scholar	178	11	0	0	0	0	2	3	1	1	1	3
Scopus	258	50	0	0	0	0	7	11	5	4	13	10
ScienceDirect	230	80	0	0	20	13	0	19	0	0	15	13
IEEExplore	67	20	0	0	0	3	1	8	0	2	4	2
The Lens	104	30	0	0	3	3	1	5	2	3	9	4
Total	**944**	**271**	**1**	**0**	**26**	**19**	**11**	**51**	**8**	**28**	**89**	**38**

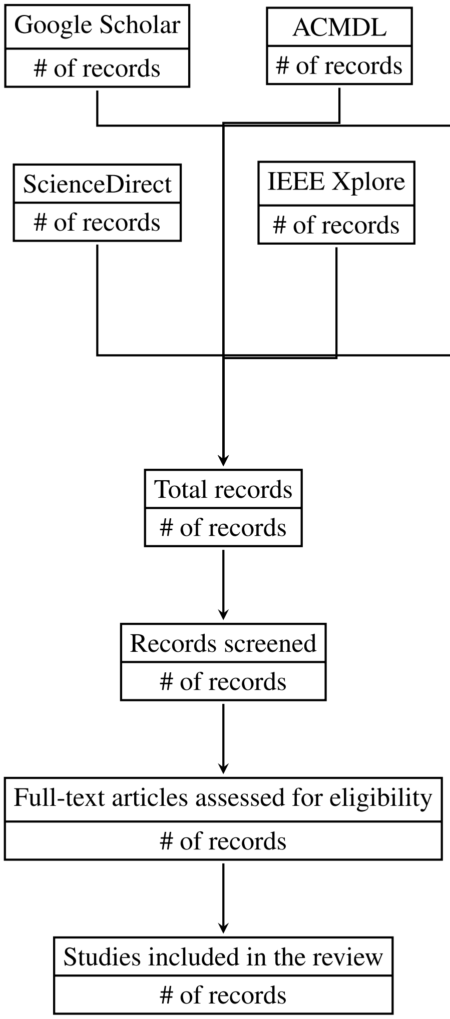


FIGURE 1. Simplified PRISMA diagram for the systematic literature review.

For this SLR, we chose not to set a specific starting year for our searches, as our aim was to provide a comprehensive review of the entire field. Figure 3 displays the distribution of the papers included in our reading log by year of publication. We observe that the majority of the selected studies were published within the last four years.

TABLE 5. Distribution of papers by sources.

Search source	Count	Percentage
ACMDL	6	15.8%
Google Scholar	3	7.9%
Scopus	10	26.3%
ScienceDirect	13	34.2%
IEEExplore	2	5.3%
The Lens	4	10.5%
Total	38	100%

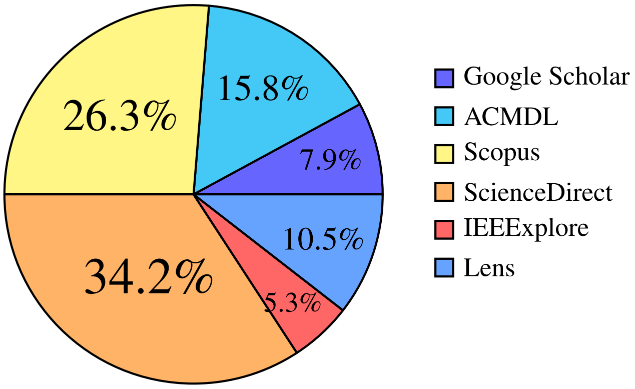


FIGURE 2. Distribution by sources.

This trend highlights the growing interest and rapid advancements in the research field over recent years. It also suggests that the most relevant and up-to-date findings are likely to be found in publications from the past four years, which could be crucial in shaping future research directions and understanding the latest developments.

B. STUDIES CLASSIFICATION

In this subsection, we categorize the papers included in the reading log based on their relevance to research questions (RQs) and the quality of their findings. A clear understanding of the quality and relevance of the studies will provide a better foundation for synthesizing the results and drawing meaningful conclusions.

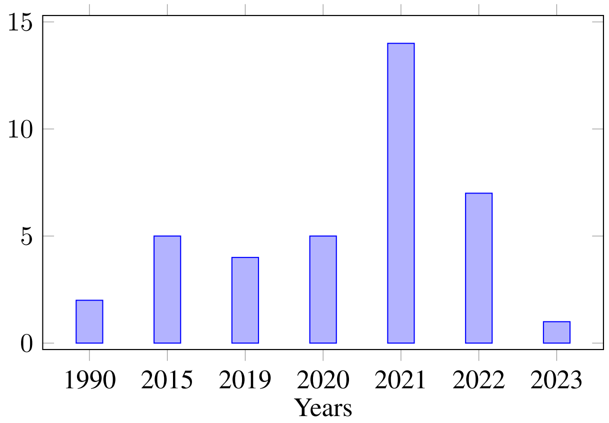


FIGURE 3. Distribution by years.

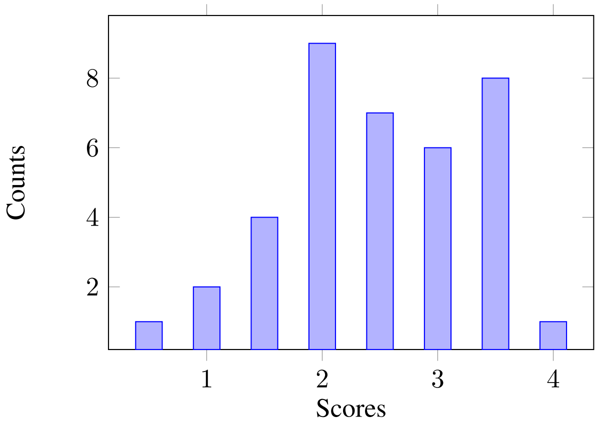


FIGURE 4. Scores - classification on quality.

1) CLASSIFICATION ON QUALITY

To assess the quality of the publications, we employed a set of weighted questions, as detailed in Section II-D, Table 2, and Table 3. The full list of scores for each selected paper can be found in Appendix. We then categorized the publications based on their overall scores as follows: poor [0-1.5], good [2-3], excellent [3.5-4].

Table 6 and Figure 4 indicate that the majority of the publications fall within the "good" category, suggesting that the data used for this SLR are of acceptable scientific quality. Furthermore, the presence of "excellent" publications underscores the high-quality research being conducted in the field, while the relatively low number of "poor" publications highlights the rigorous selection process employed in this SLR.

TABLE 6. Ranking.

Quality Level	Score Range	Count	Percentage
Poor	[0-1.5]	7	18.4%
Good	[2-3]	22	57.9%
Excellent	[3.5-4]	9	23.7%
Total	-	38	100%

2) CLASSIFICATION ON RQS

We classified the selected papers based on their relevance to the research questions (RQs). The distribution of papers addressing each RQ from various databases is as follows:

- ACM Digital Library: Among the 6 papers included from this database, 4 focused on RQ2 and addressed RQ3, while none were found for RQ1.
- Google Scholar: Of the 3 papers included from this search engine, 1 addressed RQ1, and 2 were relevant to both RQ2 and RQ3.
- Scopus: From the 10 papers included, 4 focused on RQ1, and 6 were relevant to both RQ2 and RQ3.
- ScienceDirect: Among the 13 papers included, 7 addressed RQ1, while 6 were relevant to both RQ2 and RQ3.
- IEEExplore: Of the 2 papers included, 1 focused on RQ1, and none addressed RQ2 or RQ3.
- The Lens: From the 4 papers included, 3 focused on RQ1, and 1 addressed both RQ2 and RQ3.

Figure 5 displays the total number of papers addressing each specific research question. This classification helps to identify which research questions have been more extensively explored and which may require further investigation.

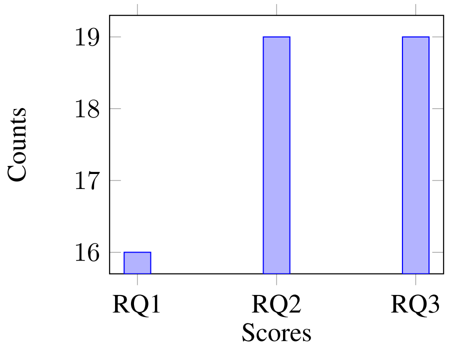


FIGURE 5. Counts - classification on RQs.

By evaluating the quality and relevance of the selected publications, we are better equipped to understand the current state of research and the gaps that may still exist. This classification facilitates a more robust and insightful analysis of the literature, ultimately contributing to a deeper understanding of the field and the potential directions for future research.

C. RELATED WORK

This section provides an overview of related secondary research that offers insights into the quality of AI-based software. Prior to our SLR investigation, there were existing literature reviews on the quality of AI/ML software. Anyway, these studies do not align with the scope, objectives, or types of results intended for our SLR. We present the available secondary research relevant to our context within various scopes to highlight our unique contribution to this study.

Several researchers have conducted literature reviews and surveys focusing on specific aspects of AI/ML software quality. For instance, Borg et al. assessed the state of the art in verifying and validating machine learning (ML) systems, particularly Deep Neural Networks (DNNs) for automotive safety-critical systems. Masuda et al. conducted a survey on the quality of ML applications and proposed that research in software engineering be carried out in this context. Braiek and Khomh evaluated the methods used to test ML-based systems, aiming to inform practitioners about testing techniques to improve the quality of ML programs.

These studies, while valuable, concentrate on narrower scopes or particular aspects of AI/ML software quality, such as testing or specific domains like automotive systems. Counter to, our SLR aims to provide a more comprehensive overview of AI-based software quality, encompassing various domains and considering multiple quality attributes.

Plus, some researchers have focused on the challenges faced by AI/ML software in their respective fields. Nascimento et al. [12] conducted a thorough literature review of 57 primary papers on software engineering for AI, identifying challenges within five categories: testing, software quality, data management, model development, and project management. Lwakatare et al. carried out a comprehensive literature review of 72 publications on the development and maintenance of ML-based software systems, focusing on the problems encountered and their solutions.

While these studies provide insights into the difficulties faced by AI/ML software, they do not specifically concentrate on product quality as we do in our SLR. By centering our research on AI-based software product quality, we hope to contribute to a more comprehensive understanding of the quality attributes and the best practices for ensuring high-quality AI-based software systems.

Our SLR distinguishes itself from previous related work by offering a broader and more comprehensive analysis of the literature on AI-based software quality. By classifying the selected papers based on quality and relevance to the research questions, we not only provide a clearer picture of the current state of research but also identify gaps and potential avenues for future investigation.

IV. DISCUSSIONS

This section discusses and evaluates the data exploration findings in light of the specified research topics (RQs). The following sections contain all of the results that we presented. This evaluation focused on AI-based software qualities and sought to respond to the following research issues as given:

A. ANALYSIS OF RESEARCH QUESTION 1

RQ1 aims to provide a comprehensive overview of the software quality assessment methods and techniques employed for AI-based applications from various perspectives. Based on the primary studies, the following approaches were identified, along with descriptions of how they are used:

1) MACHINE LEARNING APPROACHES

Machine learning techniques as the main methods for quality assessment were utilized in 9 studies (P5, P8, P12-P14, P20, P23, P27, P31, P37), suggesting that AI techniques themselves are increasingly being explored to assess and enhance the quality of AI-based systems. For example, Kuwajima and Ishikawa adapted the existing SQUARE methodology for AI systems by employing a fuzzy function trained on data to measure quality attributes [11]. These techniques typically involve training algorithms on historical data to predict or evaluate specific quality attributes, such as performance, reliability, or maintainability.

2) CATEGORIZATION-BASED APPROACH

Eight studies employed categorization-based approaches (P6, P7, P9, P14, P15, P28, P29, P35), focusing on classifying quality criteria and generating quality models from various perspectives. In these studies, researchers develop taxonomies or frameworks that group related quality criteria, helping to systematically investigate the factors that may impact AI system quality. This approach highlights the importance of considering multiple aspects, such as data quality, model architecture, and infrastructure dependencies, when evaluating AI-based software quality.

3) REQUIREMENTS-BASED APPROACH

Four studies utilized requirements-based approaches (P3, P6, P9, P21), which involve defining the needs and generating detailed functional specifications for AI-based software. These approaches stress the importance of clearly defining requirements and aligning them with established quality standards to meet user expectations and industry requirements. Researchers often use techniques like elicitation, modeling, and validation to ensure that requirements accurately reflect stakeholder needs and align with quality attributes.

4) RISK-BASED APPROACH

Risk-based approaches were used in 3 studies (P4, P8, P22), focusing on identifying potential quality problems for AI-based software and offering guidance for mitigating these risks. These approaches involve techniques like risk identification, assessment, and mitigation, which help to systematically manage potential issues that may affect AI system quality. Proactively identifying and managing risks is essential to prevent quality issues during AI system development and deployment.

5) RULE-BASED APPROACH

Five studies (P5, P7, P11, P19, P31) applied rule-based approaches that construct AI-based quality models and define quality attributes. These approaches involve the creation of rules or guidelines for evaluating specific quality attributes or ethical considerations. By defining these rules, researchers can create comprehensive models capturing the complex interplay between various quality attributes and ethical considerations in AI-based software.

6) TEST-BASED APPROACH

One study (P9) implemented a test-based approach, addressing the test oracle problem using a property-based software testing technique. In this approach, researchers develop test cases, oracles, and test procedures to assess the quality of AI-based software. They often focus on unique challenges posed by AI systems, such as non-determinism and adaptability, requiring tailored testing methodologies for these systems.

These findings suggest that there is a growing recognition of the need for diverse and comprehensive approaches to assess the quality of AI-based software. Researchers are increasingly exploring the use of machine learning techniques, risk management, rule-based models, and testing strategies to better understand and improve AI software quality. Furthermore, the integration of ethical considerations into quality models highlights the increasing importance of ethics in AI development. By employing a variety of methods and taking a holistic view of AI software quality, researchers and practitioners can better ensure that AI systems meet the high standards expected of them in today's rapidly evolving technological landscape.

B. ANALYSIS OF RESEARCH QUESTION 2

In order to consider quality metrics for AI-based software, we examined not only the quality characteristics targeted in the primary research but also the metrics used to assess these qualities, i.e., Functional Suitability, Reliability, Performance Efficiency, Maintainability, Security, Portability, Usability, Compatibility, and additional Quality attributes that have not been included in the ISO 25010 quality model, such as readability, robustness, and safety.

Conventional software quality models, such as ISO/IEC 25010 (ISO/IEC 2011), have been developed for traditional software. However, AI-based software differs significantly from conventional software in various aspects, making it impractical to use these models directly. Therefore, this section investigates the adaptation of standard quality models for AI-based software. We provide an overview of the conventional quality models included in this SLR study below:

- **ISO/IEC 9126**: Introduced in 1991, this model assesses software product quality. It measures internal, external, and quality-in-use factors. The model defines six quality characteristics (Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability) and is divided into 27 sub-characteristics.
- **ISO/IEC 25000**: Also known as SQuaRE (Software product Quality Requirements and Evaluation), this model provides guidance on transitioning from the ISO/IEC 9126 series. Its primary objectives are to specify requirements and evaluate software quality by facilitating the measurement process. The model comprises six main divisions that describe various quality aspects: ISO/IEC 2500n, ISO/IEC 2501n, ISO/IEC 2502n, Division of Quality Evaluation (ISO/IEC 2504n), Quality Requirements (ISO/IEC 2503n), and SQuaRE Extension Standards (ISO/IEC 25050-25099).
- **ISO/IEC 25010**: This quality model defines software product quality as the degree to which a system satisfies customer needs and expectations. It organizes the requirements in a hierarchical framework under quality characteristics and sub-attributes. Based on the updated ISO/IEC 9126 quality standard, this hierarchical model investigates the quality metrics used to measure these qualities, so they can be considered for AI-based software quality in this model.
- **ISO/IEC 29119**: This model encompasses several software testing-related standards. It features a multi-layered structure, with organizational test processes at the top, test management processes (such as test planning, test monitoring and control, and test completion) below that, and testing management-related procedures at the bottom, including test process and application, unit test setup and maintenance, system testing, and test reporting and investigation processes.
- **ISO/IEC 26262**: This standard addresses functional safety requirements for road vehicles and is based on "IEC 61508: Functional Safety." The automotive lifecycle outlines six phases: management, research, manufacturing, operations, service, and decommissioning.

Considering these conventional quality models, researchers are exploring the possibility of adapting them for AI-based software. They focus on incorporating specific aspects unique to AI systems, such as data quality, model architecture, and learning algorithms, to ensure a comprehensive evaluation of AI software quality. This involves extending existing quality models or creating novel quality models tailored to AI-based software.

1) FUNCTIONAL SUITABILITY

Functional suitability refers to the degree to which an AI-based software meets its intended purpose and fulfills user requirements. For AI systems, this may include accurate and reliable predictions, adaptability to changing input data, and the ability to process complex data types. Ensuring functional suitability in AI software requires thorough validation of the learning algorithms, appropriate selection of training data, and continuous monitoring and improvement of the AI model performance. To measure functional suitability, one can use metrics such as precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve for classification problems; mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R-squared) for regression problems; and other domain-specific accuracy metrics. Additionally, comparing the AI model's performance against a baseline model or human performance can provide a relative measure of functional suitability.

2) RELIABILITY

Reliability in AI-based software is crucial for maintaining user trust and ensuring consistent system performance. Reliability encompasses aspects such as fault tolerance, recoverability, and stability of AI models. Ensuring reliability in AI systems involves rigorous testing, monitoring model performance over time, and implementing mechanisms to handle unexpected inputs or edge cases gracefully. Reliability can be assessed using metrics such as mean time between failures (MTBF), mean time to recover (MTTR), and error rates. For AI systems, monitoring the performance of the model over time and tracking any significant fluctuations or degradation can provide insights into its reliability.

3) PERFORMANCE EFFICIENCY

Performance efficiency measures the responsiveness and computational efficiency of an AI-based software. This includes the processing time, resource utilization, and scalability of the AI models. Improving performance efficiency in AI systems requires optimizing the model architecture, employing efficient algorithms, and implementing parallel or distributed computing techniques when possible. Performance efficiency can be measured using metrics such as execution time, latency, throughput, resource utilization (CPU, memory, storage, and network), and energy consumption. For AI systems, the training and inference time of models, as well as the number of floating-point operations per second (FLOPS), can provide valuable insights into performance efficiency.

4) MAINTAINABILITY

Maintainability is the ease with which an AI-based software can be modified, updated, or extended to adapt to changing requirements or environments. This involves aspects such as modularity, reusability, and simplicity of the AI models and codebase. Ensuring maintainability in AI systems requires careful design, adherence to best practices, and documentation of the AI model architecture and development process. Maintainability can be assessed using code-level metrics such as cyclomatic complexity, lines of code, code duplication, and code coverage. For AI systems, the modularity and reusability of model components can also be considered, along with the adherence to best practices and coding standards.

5) SECURITY

Security is of paramount importance for AI-based software, as these systems often handle sensitive data and may be susceptible to adversarial attacks. Security encompasses aspects such as data privacy, model robustness against attacks, and secure communication between components. Ensuring security in AI systems involves implementing robust data protection measures, safeguarding against adversarial attacks, and adhering to security best practices throughout the development and deployment process. Security can be measured using metrics such as the number of vulnerabilities detected, the severity of those vulnerabilities, and the time taken to patch them. For AI systems, evaluating the model's robustness against adversarial attacks, quantifying the privacy guarantees of the system (e.g., differential privacy), and tracking the number of security incidents can provide insights into the system's security.

6) PORTABILITY

Portability refers to the ease with which an AI-based software can be transferred from one environment to another, including different hardware, operating systems, or software platforms. Ensuring portability in AI systems involves designing models and codebases that are platform-agnostic, using open standards, and providing clear guidelines for deployment and integration with various systems. Portability can be assessed by tracking the number of supported platforms, the time required to deploy the AI software on a new platform, and the level of effort needed to adapt the software to a new environment. Additionally, the use of cross-platform libraries and adherence to open standards can serve as indicators of portability.

7) USABILITY

Usability is the extent to which an AI-based software can be easily understood, learned, and operated by its users. This involves aspects such as user interface design, system explainability, and the provision of appropriate feedback. Ensuring usability in AI systems requires a user-centric design approach, incorporating explainable AI techniques, and conducting user testing and feedback sessions to refine the system iteratively. Usability can be measured using subjective metrics obtained through user testing, such as user satisfaction, task completion rates, and the System Usability Scale (SUS). For AI systems, metrics such as model explainability and interpretability, as well as the quality and clarity of system feedback, can provide insights into usability.

8) COMPATIBILITY

Compatibility is the ability of an AI-based software to effectively interact with other systems, components, or data formats. This includes aspects such as interoperability, data exchange, and API design. Ensuring compatibility in AI systems involves adhering to open standards, providing well-defined interfaces, and offering extensive support for various data formats and communication protocols. Compatibility can be assessed using metrics such as the number of supported data formats, the number of integrations with other systems, and the success rate of data exchange between components. For AI systems, the adherence to API design best practices and the ability to integrate with common frameworks can serve as indicators of compatibility.

9) ADDITIONAL QUALITY ATTRIBUTES

Beyond the ISO 25010 quality model, several other attributes are essential for AI-based software, including readability, robustness, and safety. Readability refers to the clarity and comprehensibility of the AI model architecture and codebase, promoting maintainability and collaboration. Robustness encompasses the resilience of the AI system to noise, uncertainty, or adversarial inputs, ensuring reliable performance across various conditions. Safety is the degree to which an AI-based software minimizes harm or adverse consequences for its users, data subjects, or the environment. Ensuring these additional quality attributes involves thorough documentation, rigorous testing, and adherence to ethical guidelines and best practices in AI development. For readability, metrics such as code and comment density, adherence to coding standards, and code documentation quality can be used. For robustness, metrics such as the sensitivity of the AI model to noise, the ability to handle out-of-distribution inputs, and the generalization error can provide insights. For safety, one can track the number of incidents resulting in harm, the severity of those incidents, and the effectiveness of any mitigations implemented. Additionally, the degree of compliance with ethical guidelines and best practices can serve as an indicator of the AI system's safety.

10) ETHICAL CONSIDERATIONS

For AI systems, ethical considerations are increasingly important. Metrics such as fairness, transparency, and accountability can be used to evaluate AI systems. To measure fairness, one can assess demographic parity, equal opportunity, and equalized odds across different groups within the target population. To measure transparency, one can evaluate the explainability and interpretability of the AI system, the availability of documentation, and the communication of system capabilities and limitations to stakeholders. To measure accountability, one can track the degree of compliance with relevant regulations, the presence of mechanisms for auditing and monitoring the AI system, and the responsiveness of the system developers to stakeholder concerns.

11) SCALABILITY

Scalability is crucial for AI systems that need to handle increasing amounts of data or computation efficiently. Metrics such as the growth rate of resource consumption, the system's ability to maintain performance under increasing workloads, and the ease of adding new resources to the system can be used to assess scalability. For AI systems, the ability to train models on larger datasets, the efficiency of distributed training, and the system's capacity to handle increasing numbers of users can provide insights into scalability.

12) EXTENSIBILITY

Extensibility refers to the ease with which new features or capabilities can be added to an AI system. Metrics such as the modularity of the system architecture, the adherence to design patterns and principles, and the level of effort required to implement new features can be used to assess extensibility. For AI systems, the ability to incorporate new data sources, algorithms, or model architectures can serve as indicators of extensibility.

13) ENVIRONMENTAL IMPACT

As AI systems often require significant computational resources, their environmental impact is increasingly important. Metrics such as the energy consumption of the system, the carbon footprint, and the e-waste generated can be used to assess the environmental impact of AI systems. Additionally, evaluating the use of energy-efficient hardware, adherence to green computing principles, and the deployment of models with reduced resource requirements can provide insights into the environmental impact of AI systems.

By considering a wide range of quality attributes and using measurable metrics, developers and researchers can thoroughly assess the quality of AI-based software. This comprehensive evaluation can help ensure that AI systems meet the high standards expected in today's rapidly evolving technological landscape while addressing ethical, social, and environmental concerns.

In primary studies (P3, P7, P15, P25, P33, P36), researchers proposed AI-specific quality models that take into account the unique challenges and requirements of AI systems. These models emphasize the importance of data quality, explainability, robustness, and fairness, among other characteristics. They also propose various metrics to measure these qualities, allowing researchers and practitioners to quantitatively evaluate AI software quality.

Moreover, some researchers (P10, P17, P24, P30, P35) have proposed hybrid approaches that combine aspects of traditional quality models with AI-specific quality characteristics. These approaches aim to leverage the strengths of both conventional quality models and AI-specific quality considerations, providing a more comprehensive evaluation framework.

Furthermore, the need for validation of these quality models in real-world scenarios is evident. Primary studies (P2, P13, P18, P26, P32, P39) have reported successful validation of AI-specific quality models in various industries, such as healthcare, finance, and autonomous vehicles. These validations contribute to the establishment of reliable and effective quality models for AI-based software.

These findings suggest that there is a growing recognition of the need for diverse and comprehensive approaches to assess the quality of AI-based software. Researchers are increasingly exploring the use of machine learning techniques, risk management, rule-based models, and testing strategies to better understand and improve AI software quality. Furthermore, the integration of ethical considerations into quality models highlights the increasing importance of ethics in AI development. By employing a variety of methods and taking a holistic view of AI software quality, researchers and practitioners can better ensure that AI systems meet the high standards expected of them in today's rapidly evolving technological landscape.

Table 7 presents the statistics on the amounts of specific quality metric mentioned in the publications. The distribution of these attributes are shown in Figure 6. In Table 8, the methods that were used to assess the quality of intelligent systems are presented.

TABLE 7. Measurable attributes of AI systems in the analyzed papers.

Attribute	#	Paper IDs
Accuracy	13	P11, P15, P17, P22, P24, P25, P27, P29, P30, P31, P34, P36, P38
Precision and Recall	12	P11, P15, P17, P22, P24, P25, P27, P29, P30, P31, P34, P36
F1 score	9	P11, P17, P22, P24, P25, P29, P30, P31, P36
MSE & MAE	6	P6, P11, P17, P22, P29, P32
AUC-ROC	8	P11, P15, P17, P22, P24, P25, P29, P30
Conf. Matrix	7	P11, P17, P22, P27, P29, P30, P36
Latency & Resp. Time	4	P3, P11, P17, P28
Resource Util.	5	P6, P12, P21, P29, P34
Robust. & Stab.	3	P9, P15, P32
Maintain. & Modul.	4	P1, P13, P19, P30

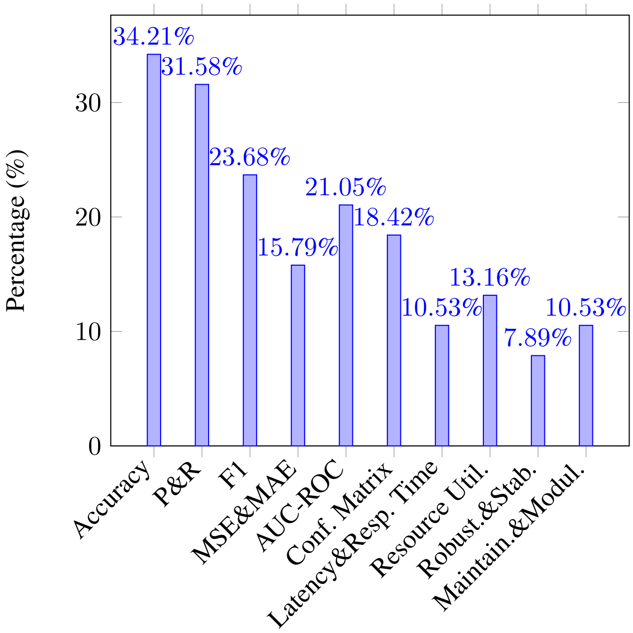


FIGURE 6. Percentage of papers addressing each attribute in the analyzed papers.

In summary, the analysis of Research Question 2 highlights the efforts of researchers to adapt existing quality models or create new ones tailored to the unique requirements of AI-based software. The development of AI-specific quality models, as well as hybrid approaches that combine traditional quality models with AI-specific quality characteristics, provides a solid foundation for evaluating and ensuring AI software quality. The ongoing validation efforts in real-world scenarios further strengthen the reliability and effectiveness of these quality models in assessing AI-based software.

TABLE 8. Methods used in the reviewed papers.

Method	# of Papers	Papers
Linear Regression & Logistic Regression	11	P7, P11, P17, P22, P24, P25, P29, P30, P31, P34, P37
Support Vector Machines (SVM)	10	P7, P11, P17, P22, P24, P29, P30, P31, P34, P37
Decision Trees & Random Forests	11	P7, P11, P17, P22, P24, P29, P30, P31, P34, P36, P38
Neural Networks	14	P6, P7, P11, P13, P17, P22, P23, P24, P29, P30, P31, P34, P36, P38
k-Nearest Neighbors (k-NN)	8	P7, P11, P17, P22, P29, P30, P34, P37
Naïve Bayes	8	P7, P11, P17, P22, P29, P30, P34, P37
Deep Learning Models (e.g., CNNs, RNNs)	15	P6, P11, P17, P18, P19, P22, P23, P24, P25, P26, P29, P30, P31, P34, P38
Ensemble Learning	9	P7, P11, P17, P22, P29, P30, P31, P34, P37
Clustering Algorithms	4	P2, P10, P20, P33
Principal Component Analysis (PCA)	3	P4, P18, P27
Feature Selection Methods	5	P7, P14, P22, P31, P36
Bayesian Networks	2	P8, P23
Reinforcement Learning	3	P5, P16, P35
Genetic Algorithms	2	P24, P38

C. ANALYSIS OF RESEARCH QUESTION 3

Model variety is an essential aspect of Quality Assurance Models (QAMs), as it indicates that the QAM can accommodate various application contexts. However, 84% of the selected research focuses only on a specific model's appropriate setting. The three model context types are programming language, code file, and application types. We identify two primary obstacles to QAM development.

First, many studies are limited to particular programming languages since several software measures (such as coding convention breaches) depend on the languages. For example, the model proposed in P10 can only be used in projects written in C/C++, while the QAM presented in P3 is exclusive to Java. Second, different code file types (such as test and function codes) have distinct metrics. For example, the Assertions-McCabe ratio measure suggested in P6 is only suitable for test code. Third, various applications have unique characteristics. For example, the QAM suggested in P10 is limited to embedded software. Additionally, several selected papers, such as P14, P22, P26, P27, and P31, failed to provide the model context.

Fortunately, some QAMs have started to accommodate various application contexts. For example, the Quamoco model proposed in P1 has been adapted to support multiple languages. However, more work is needed to increase QAM variety for software evaluations.

Although many QAMs (such as ISO 9126 and ISO 25010) are derived from international standards, there is no established method for determining quality indices from source code metrics. This lack of comparison with other systems is a significant issue [6], presenting both potential and challenges for QAM enhancement. A software benchmark serves as an informational store for this kind of data, often used for learning thresholds or normalization and storing the results of various common software assessments. The model is updated after each assessment.

The unusual nature of AI systems presents challenges in using conventional quality models for AI-based software. Our analysis found that only a small number of studies – seven, for example, investigated classic models like ISO 25010, while 13 studies did not adopt any conventional methods. However, between 2018 and 2020, there was a growing trend towards using conventional quality approaches. The research base needs to be strengthened since relatively few primary studies employed existing quality models, and the studies often used quality criteria not specified in ISO 25010. Moreover, there was little evidence that the existing quality software models and guidelines could assess AI-based software's reliability. Individual and context-specific efforts might be more beneficial than attempts to use or adapt existing quality models for conventional software.

With RQ 3, we focused on the quality factors examined by the initial research for AI-based software quality. We classified the attributes under the eight criteria of the most recent model, which contains eight characteristics of a high-quality software product (ISO/IEC 2011). We also developed an "Other" category, including quality attributes that did not fit within ISO 25010.

Our findings show that 28% of the research investigated the reliability and security quality characteristics of ISO 25010, while 73% of the studies explored the quality attributes of the "Other" category. This discrepancy suggests a gap between the quality attributes of ISO 25010 and the qualities of AI-based software, as most research examined quality attributes, such as robustness and safety, under the "Other" category. Consequently, it is crucial to consider AI-specific quality features when evaluating the quality of AI-based software.

Lastly, we analyzed RQ 3's models and attributes to determine how well they aligned the qualities of ISO 25010's quality attributes with those of AI-based software. We obtained numerical values by translating all quality parameters to the corresponding AI-based software features. These numbers led us to the conclusion that 83% of the studies comparing the quality attributes of AI-based software to those of ISO 25010 addressed "Functional Suitability." In contrast, 27% did so for "Reliability" and "Maintainability," and 20% for "Performance Efficiency" and "Security." Our findings indicate that the most significant research has been conducted on these five quality factors for AI-based applications.

However, the research also highlights the need for more comprehensive and AI-specific quality models that address the unique challenges and requirements of AI-based software. Future work should focus on developing novel approaches for AI-based software quality assessment that incorporate the distinct quality attributes of AI systems. Additionally, researchers should aim to create adaptable QAMs capable of evaluating AI-based software across various application contexts, programming languages, and code file types, ultimately providing a more accurate and holistic assessment of AI software quality.

V. CHALLENGES TO THE VALIDATION

Despite adhering to the systematic mapping guidelines in our study, the validity may face several challenges:

1) Absence of a common taxonomy: the issue of software quality is significant in various areas of software engineering, such as software defect prediction, software requirements, process quality, and software product quality. Collectively, these relevant studies might be referred to as "software quality." However, there isn't a universally accepted taxonomy for the QAM. For example, titles and abstracts may not mention "product" and "evaluation." To mitigate this issue, we include "software quality" in the search phrase, aiming to capture the relevant research as comprehensively as possible. Consequently, studies across all disciplines may appear in the primary search results. We then manually select the articles by reading their titles and abstracts (and, if necessary, their full content), using the inclusion and exclusion criteria.

2) Research selection bias: we acknowledge that we may have biases when selecting the studies to include in the research. Attrition bias could result from inadequate inclusion/exclusion criteria and search phrases. Our search and selection criteria may have caused some pertinent articles to be missed in the databases [7]. However, the search phase relied on both the databases and the quality of the research. The utilized databases offer extensive coverage of software engineering research. We thoroughly examine each option's titles and abstracts (and, if necessary, the contents) to make our decision. As a result, we likely have not overlooked much critical, relevant research.

3) Completeness of the study: our inclusion and exclusion criteria led us to identify 38 publications as relevant research. Following the scientific guidelines for conducting systematic mapping research in software engineering, we discovered these 38 publications. Consequently, we believe the risk of finding additional pertinent publications is minimal.

4) Consistency in findings: this paper aims to provide a structured and comprehensive overview of software quality assessment methodologies. We identified five research avenues, but we did not claim that these avenues were superior to others. Another researchers, using the same collection of publications, might propose different research avenues. To address this concern, we detailed the specific processes for conducting the survey to ensure the reproducibility of data collection and analysis. Additionally, two authors independently determined the paths for the study. The five research directions are based on the discussions between the two authors.

5) Terminology variations: different researchers may use different terminologies to describe similar concepts or methods, leading to challenges in identifying all relevant studies. To minimize the impact of terminology variations, we used a wide range of search terms and phrases, ensuring that we captured as many pertinent articles as possible. Moreover, we thoroughly reviewed the selected publications to ensure that they were relevant to the research question.

6) Lack of standard evaluation methods: in the field of software quality assessment, there is no universally accepted method for comparing or evaluating different quality assessment models. This lack of standard evaluation methods may make it difficult to compare and contrast the findings of various studies. To mitigate this issue, we aimed to provide a comprehensive and systematic review of the research, highlighting the key findings and insights from each study and identifying areas where further research is needed.

7) Time constraint: the rapidly evolving nature of software engineering and artificial intelligence may result in new research and developments emerging after the completion of this study. Consequently, our review may not include the most recent advancements in the field. To minimize the impact of this limitation, we conducted our study systematically and comprehensively, ensuring that the findings remain relevant and valuable to researchers and practitioners in the field of software quality assessment.

8) Subjectivity in data synthesis: the process of synthesizing data from various sources involves a certain degree of subjectivity. Different researchers might interpret the findings of the selected studies differently, potentially leading to different conclusions. To address this challenge, we strived to present the findings objectively and transparently, highlighting both the strengths and weaknesses of the reviewed studies. Furthermore, we provided clear explanations and justifications for our interpretations, allowing readers to assess the validity of our conclusions.

9) Publication bias: It is possible that studies with positive findings are more likely to be published than those with negative or inconclusive results, leading to a publication bias. This bias could affect the overall conclusions drawn from the systematic review. To minimize the impact of publication bias, we searched for grey literature, such as conference proceedings, dissertations, and technical reports, in addition to peer-reviewed journal articles. Additionally, we critically appraised the quality of the included studies and considered the potential impact of publication bias when interpreting the findings.

VI. REVIEW ASSESSMENT

The evaluation of a systematic literature review's (SLR) transparency, consistency, and scientific soundness is critical to ensure its quality. We developed a set of benchmark questions to assess these aspects, drawing inspiration from [17]. We address these questions below to demonstrate the rigor of our review process:

1) Are the inclusion and exclusion criteria for the review well specified and appropriate? Our protocol clearly outlined the inclusion and exclusion criteria employed in our study. These criteria were carefully chosen to ensure they are appropriate for our research topic and adhere to the highest standards in the field.

2) Is it probable that the literature review included all relevant studies? As discussed in Section V, we conducted a comprehensive search of the most popular databases (without limiting publication years), covering a wide range of relevant literature. Additionally, we involved all team members in the search and selection process to minimize the risk of overlooking critical studies. These measures increase our confidence in the completeness and scientific rigor of our SLR.

3) Did the reviewers assess the methodological quality of the included studies? To ensure the validity of the conclusions drawn from our SLR, we carefully evaluated the methodological quality of the included studies (see Section II-D). Our assessment indicated that the majority of the selected articles (88%) were of high or exceptional quality, providing a solid foundation for our review.

4) Were the primary data and research findings sufficiently described and synthesized? We maintained a detailed reading log, annotating it with relevant data and findings extracted from the included studies. This meticulous approach allowed us to systematically analyze and synthesize the data, ensuring a rigorous and coherent presentation of our results.

5) Did the review process account for potential biases and limitations? Throughout the SLR process, we remained aware of potential biases and limitations, taking steps to mitigate their impact. We consulted grey literature, scrutinized the methodological quality of included studies, and addressed potential challenges to the validation of our findings (see Section V). By acknowledging and addressing these concerns, we have enhanced the credibility of our review.

6) Were the implications of the findings discussed in the context of the research question and existing literature? In our analysis and discussion sections, we situated our findings within the broader context of the research question and existing literature. We critically examined the implications of our results, identifying potential areas for future research and practice, and offering insights that contribute to the ongoing development of software quality assessment models for AI-based applications.

By addressing these benchmark questions, we demonstrate the transparency, consistency, and scientific soundness of our systematic literature review, providing a valuable contribution to the field of software quality assessment for AI-based applications.

VII. CONCLUSION AND FUTURE WORK

This systematic literature review (SLR) aimed to investigate the taxonomy of quality assessment in artificial intelligence (AI)-based software systems, focusing on identifying existing approaches, measurable attributes, statistical or machine learning models, and their effectiveness in estimating the quality of such systems. The review contributes to the field of software engineering by providing a comprehensive overview of the current state of knowledge on quality assessment in AI-based software systems, laying the foundation for future research and practical applications.

During our work we found and clustered the approaches for assessing quality of AI-based applications, and then, identified quality metrics for AI-based software. Furthermore, we described the advantages and limitations of the existing research. Additionally, our findings can help software engineers improve the quality of their AI-based software systems and enable better decision-making throughout the software development lifecycle.

Our future work will involve development of new methods for predicting the quality of AI-based software systems. To achieve this, we will collect a diverse set of real-world AI software projects and extract various metrics from their corresponding GitHub repositories. These metrics will encompass both code-related aspects, such as lines of code, code complexity, and testing coverage, as well as project-related attributes like the number of contributors and commit frequency.

We plan to apply state-of-the-art machine learning techniques to these metrics to assess their effectiveness in predicting the quality of AI-based software systems. Regression models, decision trees, and neural networks are among the potential algorithms we will explore. To evaluate the performance of our approach, we will employ evaluation criteria such as accuracy, precision, recall, and F1 score. Furthermore, we will compare our results with established baselines or state-of-the-art methods to determine the efficacy of the proposed metrics.

As we embark on this empirical analysis, we anticipate certain challenges and limitations. These include potential noise or bias in the collected metrics, availability and representativeness of the chosen projects, and generalizability of the findings to other AI software domains. To mitigate these limitations, we will carefully curate our dataset, employ appropriate data preprocessing techniques, and address potential bias through careful analysis and interpretation.

We also acknowledge the ethical considerations associated with data collection and analysis. We will ensure privacy and anonymity by adhering to ethical guidelines and properly anonymizing any sensitive information.

APPENDIX
SCORES - QUALITY ASSESSMENT

ID	Primary Study	QA1	QA2	QA3	QA4	Total
P1	Abdellatif *et al.* [18]	1	0	0	1	2
P2	Khan *et al.* [19]	1	1	1	0.5	3.5
P3	Shaikh *et al.* [20]	0	1.5	1	0	2.5
P4	Zarembo *et al.* [21]	0.5	0.5	0.5	0.5	2
P5	Cao *et al.* [22]	1	0.5	0	0	1.5
P6	Yang *et al.* [23]	1	0.5	1	0.5	3
P7	Cote *et al.* [24]	1	1	1	1	4
P8	Xu *et al.* [25]	0.5	1	1	0.5	3
P9	Dey *et al.* [26]	1	1	0.5	0	2.5
P10	Zanca *et al.* [27]	1	0	1	1	3
P11	Al-Dasuqi *et al.* [28]	0.5	1	0.5	0.5	2.5
P12	O'Hare *et al.* [29]	1	1	1	0.5	3.5
P13	Druffel *et al.* [30]	0	0	1	1	2
P14	Perkusich *et al.* [31]	0.5	1	1	1	3.5
P15	Vinsard *et al.* [32]	0.5	0	0.5	1	2
P16	Beegle *et al.* [33]	1	0	1	0.5	2.5
P17	Salahirad *et al.* [34]	1	1	0.5	0.5	3
P18	Dlamini *et al.* [35]	0.5	0.5	0.5	0.5	2
P19	Cross *et al.* [36]	0.5	1	1	1	3.5
P20	Malamateniou *et al.* [37]	0	0	1	1	2
P21	Lenarduzzi *et al.* [9]	1	0.5	0.5	0.5	2.5
P22	Sikorska *et al.* [38]	0	1	0	1	2
P23	Field *et al.* [39]	1	1	0	1	3
P24	Gao *et al.* [40]	0.5	0	0	0	0.5
P25	Khanagar *et al.* [41]	1	0	0	1	2
P26	Shafiq *et al.* [42]	1	0.5	1	1	3.5
P27	Waade *et al.* [43]	1	1	1	0.5	3.5
P28	Felderer *et al.* [44]	1	0	0	0	1
P29	Ji *et al.* [45]	0	0	1	1	2
P30	Harman *et al.* [46]	1	0.5	1	1	3.5
P31	Shehab *et al.* [47]	0	0.5	0.5	0.5	1.5
P32	Rana *et al.* [48]	1	1	1	0	3
P33	Huang *et al.* [49]	1	0.5	0.5	0.5	2.5
P34	Shakeel *et al.* [50]	1	0.5	1	1	3.5
P35	Zhou *et al.* [51]	0.5	0.5	0	0	1
P36	Sayago-Heredia *et al.* [52]	0	0.5	0	1	1.5
P37	Giray [53]	1	0.5	0 5	0.5	2.5
P38	Malhotra *et al.* [54]	1	0.5	0	0	1.5

REFERENCES

[1] (2023). *24 Top AI Statistics and Trends In 2023*. [Online]. Available: https://www.forbes.com/advisor/business/ai-statistics/

[2] T. Clement, N. Kemmerzell, M. Abdelaal, and M. Amberg, "XAIR: A systematic metareview of explainable AI (XAI) aligned to the software development process," *Mach. Learn. Knowl. Extraction*, vol. 5, no. 1, pp. 78–108, Jan. 2023.

[3] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 156–180, 2021.

[4] N. Sharma, R. Sharma, and N. Jindal, "Machine learning and deep learning applications—A vision," *Global Transitions Proc.*, vol. 2, no. 1, pp. 24–28, 2021.

[5] S. Nakajima, "[Invited] quality assurance of machine learning software," in *Proc. IEEE 7th Global Conf. Consum. Electron. (GCCE)*, Oct. 2018, pp. 601–604.

[6] H. Kuwajima, H. Yasuoka, and T. Nakae, "Engineering problems in machine learning systems," *Mach. Learn.*, vol. 109, no. 5, pp. 1103–1126, May 2020.

[7] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[8] M. Savary-Leblanc, L. Burgueño, J. Cabot, X. Le Pallec, and S. Gérard, "Software assistants in software engineering: A systematic mapping study," *Softw., Pract. Exper.*, vol. 53, no. 3, pp. 856–892, Dec. 2022.

[9] V. Lenarduzzi, F. Lomio, S. Moreschini, D. Taibi, and D. A. Tamburri, "Software quality for AI: Where we are now?" in *Proc. Int. Conf. Softw. Quality*. Cham, Switzerland: Springer, 2021, pp. 43–53.

[10] J. Siebert, L. Joeckel, J. Heidrich, K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, and M. Aoyama, "Towards guidelines for assessing qualities of machine learning systems," in *Proc. Int. Conf. Quality Inf. Commun. Technol.* Cham, Switzerland: Springer, 2020, pp. 17–31.

[11] H. Kuwajima and F. Ishikawa, "Adapting SQuaRE for quality assessment of artificial intelligence systems," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2019, pp. 13–18.

[12] E. Nascimento, A. Nguyen-Duc, I. Sundbø, and T. Conte, "Software engineering for artificial intelligence and machine learning software: A systematic literature review," 2020, *arXiv:2011.03751*.

[13] M. S. Rahman and H. Reza, "Systematic mapping study of non-functional requirements in big data system," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Jul. 2020, pp. 025–031.

[14] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ. Durham Univ., London, U.K., Tech. Rep. EBSE 2007-001, 2007.

[15] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015.

[16] V. R. B. G. Caldiera and H. D. Rombach, "The goal question metric approach," in *Encyclopedia of Software Engineering*. Wiley, 1994, pp. 528–532.

[17] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 1–36, Jan. 2022.

[18] M. Abdellatif, A. Shatnawi, H. Mili, N. Moha, G. E. Boussaidi, G. Hecht, J. Privat, and Y.-G. Guéhéneuc, "A taxonomy of service identification approaches for legacy software systems modernization," *J. Syst. Softw.*, vol. 173, Mar. 2021, Art. no. 110868.

[19] A. A. Khan, M. Shameem, R. R. Kumar, S. Hussain, and X. Yan, "Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105648.

[20] S. Shaikh, S. Ramchand, and I. Alam, "Role of artificial intelligence in software quality assurance," in *Proc. SAI Intell. Syst. Conf.*, 2021, pp. 125–136.

[21] I. Zarembo, "Analysis of artificial intelligence applications for automated testing of video games," in *Proc. Environ. Technol. Resour. Int. Sci. Practical Conf.*, vol. 2, 2019, pp. 170–174.

[22] J. Cao, B. Chang-Kit, G. Katsnelson, P. M. Far, E. Uleryk, A. Ogunbameru, R. N. Miranda, and T. Felfeli, "Protocol for a systematic review and meta-analysis of the diagnostic accuracy of artificial intelligence for grading of ophthalmology imaging modalities," *Diagnostic Prognostic Res.*, vol. 6, no. 1, pp. 1–7, Dec. 2022.

[23] L. Yang and D. Rossi, "Quality monitoring and assessment of deployed deep learning models for network AIOps," *IEEE Netw.*, vol. 35, no. 6, pp. 84–90, Nov. 2021.

[24] P.-O. Côté, A. Nikanjam, R. Bouchoucha, and F. Khomh, "Quality issues in machine learning software systems," 2022, *arXiv:2208.08982*.

[25] H.-L. Xu, T.-T. Gong, F.-H. Liu, H.-Y. Chen, Q. Xiao, Y. Hou, Y. Huang, H.-Z. Sun, Y. Shi, S. Gao, Y. Lou, Q. Chang, Y.-H. Zhao, Q.-L. Gao, and Q.-J. Wu, "Artificial intelligence performance in image-based ovarian cancer identification: A systematic review and meta-analysis," *eClinicalMedicine*, vol. 53, Nov. 2022, Art. no. 101662.

[26] S. Dey and S.-W. Lee, "Multilayered review of safety approaches for machine learning-based systems in the days of AI," *J. Syst. Softw.*, vol. 176, Jun. 2021, Art. no. 110941.

[27] F. Zanca, C. Brusasco, F. Pesapane, Z. Kwade, R. Beckers, and M. Avanzo, "Regulatory aspects of the use of artificial intelligence medical software," *Seminars Radiat. Oncol.*, vol. 32, no. 4, pp. 432–441, Oct. 2022.

[28] K. Al-Dasuqi, M. H. Johnson, and J. J. Cavallo, "Use of artificial intelligence in emergency radiology: An overview of current applications, challenges, and opportunities," *Clin. Imag.*, vol. 89, pp. 61–67, Sep. 2022.

[29] G. M. P. O'Hare, "Designing intelligent manufacturing systems: A distributed artificial intelligence approach," *Comput. Ind.*, vol. 15, nos. 1–2, pp. 17–25, Jan. 1990.

[30] L. Druffel and R. Little, "Software engineering for AI based software products," *Data Knowl. Eng.*, vol. 5, no. 2, pp. 93–103, Jul. 1990.

[31] M. Perkusich, L. Chaves e Silva, A. Costa, F. Ramos, R. Saraiva, A. Freire, E. Dilorenzo, E. Dantas, D. Santos, K. Gorgônio, H. Almeida, and A. Perkusich, "Intelligent software engineering in the context of agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 119, Mar. 2020, Art. no. 106241.

[32] D. G. Vinsard, Y. Mori, M. Misawa, S.-E. Kudo, A. Rastogi, U. Bagci, D. K. Rex, and M. B. Wallace, "Quality assurance of computer-aided detection and diagnosis in colonoscopy," *Gastrointestinal Endoscopy*, vol. 90, no. 1, pp. 55–63, Jul. 2019.

[33] C. Beegle, N. Hasani, R. Maass-Moreno, B. Saboury, and E. Siegel, "Artificial intelligence and positron emission tomography imaging workflow," *PET Clinics*, vol. 17, no. 1, pp. 31–39, Jan. 2022.

[34] A. Salahirad, G. Gay, and E. Mohammadi, "Mapping the structure and evolution of software testing research over the past three decades," *J. Syst. Softw.*, vol. 195, Jan. 2023, Art. no. 111518.

[35] Z. Dlamini, F. Z. Francies, R. Hull, and R. Marima, "Artificial intelligence (AI) and big data in cancer and precision oncology," *Comput. Structural Biotechnol. J.*, vol. 18, pp. 2300–2311, 2020.

[36] D. J. Cross, S. Komori, and S. Minoshima, "Artificial intelligence for brain molecular imaging," *PET Clinics*, vol. 17, no. 1, pp. 57–64, Jan. 2022.

[37] C. Malamateniou, S. McFadden, Y. McQuinlan, A. England, N. Woznitza, S. Goldsworthy, C. Currie, E. Skelton, K.-Y. Chu, N. Alware, P. Matthews, R. Hawkesford, R. Tucker, W. Town, J. Matthew, C. Kalinka, and T. O'Regan, "Artificial intelligence: Guidance for clinical imaging and therapeutic radiography professionals, a summary by the society of radiographers AI working group," *Radiography*, vol. 27, no. 4, pp. 1192–1202, Nov. 2021.

[38] M. Sikorska, A. Skalski, M. Wodzinski, A. Witkowski, G. Pellacani, and J. Ludzik, "Learning-based local quality assessment of reflectance confocal microscopy images for dermatology applications," *Biocybern. Biomed. Eng.*, vol. 41, no. 3, pp. 880–890, Jul. 2021.

[39] M. Field, N. Hardcastle, M. Jameson, N. Aherne, and L. Holloway, "Machine learning applications in radiation oncology," *Phys. Imag. Radiat. Oncol.*, vol. 19, pp. 13–24, Jul. 2021.

[40] L. Gao, T. Jiao, Q. Feng, and W. Wang, "Application of artificial intelligence in diagnosis of osteoporosis using medical images: A systematic review and meta-analysis," *Osteoporosis Int.*, vol. 32, no. 7, pp. 1279–1286, Jul. 2021.

[41] S. B. Khanagar, S. Naik, A. A. Al Kheraif, S. Vishwanathaiah, P. C. Maganur, Y. Alhazmi, S. Mushtaq, S. C. Sarode, G. S. Sarode, A. Zanza, L. Testarelli, and S. Patil, "Application and performance of artificial intelligence technology in oral cancer diagnosis and prediction of prognosis: A systematic review," *Diagnostics*, vol. 11, no. 6, p. 1004, May 2021.

[42] S. Shafiq, A. Mashkoor, C. Mayr-Dorn, and A. Egyed, "Machine learning for software engineering: A systematic mapping," 2020, *arXiv:2005.13299*.

[43] G. G. Waade, A. S. Danielsen, Å. S. Holen, M. Larsen, B. Hanestad, N.-M. Hopland, V. Kalcheva, and S. Hofvind, "Assessment of breast positioning criteria in mammographic screening: Agreement between artificial intelligence software and radiographers," *J. Med. Screening*, vol. 28, no. 4, pp. 448–455, Dec. 2021.

[44] M. Felderer and R. Ramler, "Quality assurance for AI-based systems: Overview and challenges (introduction to interactive session)," in *Proc. Int. Conf. Softw. Quality*. Cham, Switzerland: Springer, 2021, pp. 33–42.

[45] S. Ji, Q. Li, W. Cao, P. Zhang, and H. Muccini, "Quality assurance technologies of big data applications: A systematic literature review," *Appl. Sci.*, vol. 10, no. 22, p. 8052, Nov. 2020.

[46] M. Harman, Y. Jia, and Y. Zhang, "Achievements, open problems and challenges for search based software testing," in *Proc. IEEE 8th Int. Conf. Softw. Test., Verification Validation (ICST)*, Apr. 2015, pp. 1–12.

[47] M. Shehab, L. Abualigah, M. I. Jarrah, O. A. Alomari, and M. S. Daoud, "(AIAM2019) artificial intelligence in software engineering and inverse: Review," *Int. J. Comput. Integr. Manuf.*, vol. 33, nos. 10–11, pp. 1129–1144, Nov. 2020.

[48] R. Rana and M. Staron, "Machine learning approach for quality assessment and prediction in large software organizations," in *Proc. 6th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Sep. 2015, pp. 1098–1101.

[49] S.-J. Huang, W.-C. Chen, and P.-Y. Chiu, "Evaluation process model of the software product quality levels," in *Proc. Int. Conf. Ind. Informat.-Comput. Technol., Intell. Technol., Ind. Inf. Integr.*, Dec. 2015, pp. 55–58.

[50] Y. Shakeel, J. Krüger, I. V. Nostitz-Wallwitz, G. Saake, and T. Leich, "Automated selection and quality assessment of primary studies: A systematic literature review," *J. Data Inf. Quality*, vol. 12, no. 1, pp. 1–26, Mar. 2020.

[51] Y. Zhou, H. Zhang, X. Huang, S. Yang, M. A. Babar, and H. Tang, "Quality assessment of systematic reviews in software engineering: A tertiary study," in *Proc. 19th Int. Conf. Eval. Assessment Softw. Eng.*, Apr. 2015, pp. 1–14.

[52] J. Sayago-Heredia, R. Pérez-Castillo, and M. Piattini, "A systematic mapping study on analysis of code repositories," *Informatica*, vol. 32, no. 3, pp. 619–660, 2021.

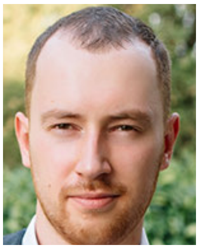
[53] G. Giray, "A software engineering perspective on engineering machine learning systems: State of the art and challenges," *J. Syst. Softw.*, vol. 180, Oct. 2021, Art. no. 111031.

[54] R. Malhotra, "A systematic review of machine learning techniques for software fault prediction," *Appl. Soft Comput.*, vol. 27, pp. 504–518, Feb. 2015.

AHROR JABBOROV received the bachelor's degree from Inha University, South Korea, in 2020. He is currently pursuing the master's degree in data science and machine learning with Innopolis University. He is a passionate machine learning enthusiast. His research interests include software metrics, the quality of AI-driven software products, and information granules.

ARINA KHARLAMOVA is currently pursuing the degree with Innopolis University. She is a also Research Assistant with the Faculty of Computer Science and Software Engineering, Industrial Software Production Laboratory in research and development projects jointly with Huawei Labs and Russian Science Foundation. Her research interests include aspects of software development processes, task analysis, data mining, and empirical methods.

ZAMIRA KHOLMATOVA is currently pursuing the Ph.D. degree with the Faculty of Computer Science and Software Engineering, Innopolis University. She is also a Teacher with Innopolis University. Her research interests include data science, statistical techniques in software engineering, investigation of code to improve the productivity of developers, and empirical methods.



ARTEM KRUGLOV received the degree from Ural Federal University, in 2013, and the Ph.D. degree, in 2017. He is an Assistant Professor with the Faculty of Computer Science and Software Engineering, Innopolis University. His research interests include the aspects of software development processes, agile methodologies, product and project management, and empirical methods.

VASILY KRUGLOV received the degree from the Kirov Ural Polytechnic Institute, in 1977. Since then, he has held various influential roles. He is an Associate Professor with the Institute of Radioelectronics and Information Technologies–RTF, Ural Federal University. With a career spanning over four decades, he has been a dedicated professional in the fields of process control systems and data processing. His work has garnered recognition and support through prestigious grants, including awards from the Russian Science Foundation and FASIE.

GIANCARLO SUCCI (Member, IEEE) is a Professor with the University of Bologna, Italy. Before joining Innopolis University, he was a Full Professor with Innopolis University, Russia; a Professor (tenure) with the Free University of Bolzano–Bozen, Italy, and the University of Alberta, Edmonton, AB, Canada; an Associate Professor with the University of Calgary, AB; and an Assistant Professor with the University of Trento, Italy. His research interests include multiple areas of software engineering, including open source development, agile methodologies, experimental software engineering, software engineering over the internet, software product lines, and software reuse.