Towards Requirements Specification for Machine Learning-Based Software Systems

Bruno Garcia de Oliveira Breda[1], **Pedro Henrique Dias Valle**[1], **Damian Andrew Tamburri**[2], **Elisa Yumi Nakagawa**[1]

[1]University of São Paulo, São Carlos, Brazil

[2]Università del Sannio, Italy, and JADS, The Netherlands

{brunobbreda3,pedrohenriquevalle}@usp.br, d.a.tamburri@tue.nl, elisa@icmc.usp.br

Abstract. *Machine Learning-based software systems (ML-based systems) have been increasingly developed for various domains, such as medicine, smart cities, and the automotive sector. These systems have posed significant challenges for Requirements Engineering (RE), including for requirements specification, while ensuring critical qualities such as reliability. The main problem addressed in this work is the difficulty practitioners have faced in specifying the requirements for these systems. Additionally, effective approaches (e.g., methods, techniques, processes, or guidelines) for this still lack consensus. The main objective of this work is to present and discuss current trends in approaches to requirements specification for ML-based systems. We observe this research field is very new, with weak collaboration with industrial environments. Novel approaches or adaptations of existing ones were proposed, but several possibilities for future research still exist; hence, we also discuss some of the main ones.*

1. Introduction

Machine Learning-based software systems (or simply ML-based systems) have been increasingly adopted in several application domains, including healthcare, transportation, and Industry 4.0 [Barrera et al. 2024]. In turn, these systems have brought significant challenges for the Software Engineering (SE) activities. In particular, managing requirements throughout their development and execution is challenging, as important quality attributes, such as reliability and performance, must be ensured. In this context, traditional Requirements Engineering (RE) practices often reach their limits when applied to ML-based systems [Villamizar et al. 2021]. A major challenge lies in the volatile nature of requirements, which are heavily dependent on the input and training data rather than deterministic functional logic. This dependency makes it difficult to specify expected behaviors accurately. Hence, requirements engineering becomes a critical challenge, mainly due to the difficulty of defining precise boundaries and handling non-deterministic outputs, which often lead to inconsistencies between stakeholder expectations and the ML model's actual outputs [Nascimento et al. 2019].

In this sense, the main problem addressed in this work is the difficulty developers have had in specifying properly the requirements of ML-based systems, given the evolving nature of these systems. The scientific literature also lacks a comprehensive overview of effective approaches (e.g., methods, techniques, processes, or guidelines) for that. Without such a knowledge base, practitioners may be left to the uncertainty of which approaches are more suitable for ensuring aspects such as quality and completeness in the specification of the requirements for these systems. This uncertainty and lack of consensus underscore the need for an investigation to map current trends, identify how researchers and practitioners are addressing these challenges, and outline future research perspectives for this area.

The main objective of this paper is to provide an overview of the current state of the art of approaches and supporting tools for requirements specification for ML-based systems, serving as a roadmap for both researchers and practitioners. By synthesizing existing approaches and identifying gaps (such as the lack of collaboration with industrial environments and agile integration), this work could guide the selection of appropriate approaches and justify the development of new ones. Furthermore, this study intends to serve as a foundation for future research by highlighting emerging trends and research opportunities.

This paper is organized as follows: Section 2 presents the research method; Section 3 presents the results; Section 4 discusses the main findings, future research perspectives, and threats to validity; and Section 5 concludes this work.

2. Research method

The research method followed a 3-phase process for systematic mapping studies (SMS) [Kitchenham et al. 2015], which includes planning, conduct, and results synthesis. The main goal of this SMS is to present the state-of-the-art requirements specification approaches (e.g., methods, techniques, processes, or guidelines) for ML-based systems (including deep learning (DL)) and the supporting tools.

2.1. Planning Phase

We defined the research protocol for this study in the planning phase and specified two research questions (RQs), the search string and search strategy, selection criteria, data extraction, and synthesis methods. The RQs were:

- **RQ1:** Which are the approaches (methods, techniques, processes, or guidelines) used for the specification of requirements of ML-based systems?
- **RQ2:** Are there any tools that could provide support for the specification of requirements of ML-based systems?

The search string defined is composed of the main keywords addressed in the RQs, namely *machine learning*, *deep learning*, *requirements engineering*, and *specification*. After calibration and validation, the following generic search string was defined, resulting from a combination of keywords and their synonyms: *((“deep learning” OR “machine learning”) AND (“requirements engineering”) AND (“modelling” OR “modeling” OR “representation” OR “specification” OR “documentation”)).* We also selected the databases most commonly used for SMS in the software engineering area [Kitchenham et al. 2015, Dyba et al. 2005], namely Scopus[1], ACM[2], IEEE Xplorer[3],

[1]https://www.scopus.com
[2]https://dl.acm.org
[3]http://ieeexplore.ieee.org

Science Direct[4] and Wiley[5]. The search string was adapted to comply with each database's search syntax, and searches were based on the titles, keywords, and abstracts of studies.

We applied the following inclusion criteria (IC) to the studies retrieved from the databases to select the primary studies for our SMS: **IC1:** *The study addresses the requirements specification for ML-based systems*. We also defined five exclusion criteria (EC): **EC1:** *The study does not address the requirements specification for machine learning-based systems*; **EC2:** *The full text of the study is not available*; **EC3:** *The study is not a primary study*; **EC4:** *The study is written in a language other than English*; and **EC5:** *The study is a previous version of an already included study.*

We also defined a data extraction form[6] to obtain detailed information from each study and answer our RQs. We collected the study title, authors, publication year, vehicles (e.g., conference, journal, or book where they were published), involvement with academia, industry, or research centers, application domain, level of maturity of the primary studies, requirements specification approaches used, and the tools used to support the requirements specification process.

2.2. Conduct Phase

We conducted the search across databases using the search string in December 2025, and 519 results were returned, as shown in Figure 1. After removing the duplicated studies, 439 studies remained. Each study was examined regarding its title, abstract, keywords, introduction, and conclusion. Applying the IC and EC, we obtained results in 48 studies. In this step, only studies outside the scope of this SMS were excluded. After reading the full text of all 48 studies and reapplying the IC and EC, 25 were excluded, and 23 (listed in Table 1) were considered relevant to answer our RQs. Following this, we extracted data using a previously validated data extraction form, ensuring that no fields were missing. Moreover, an expert analyzed and verified the correctness of the extracted data.

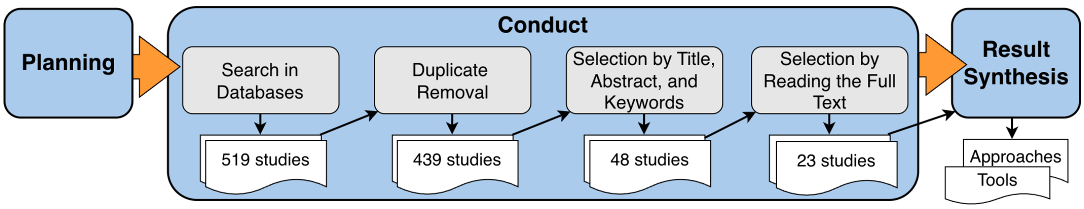

Figure 1. Research Method

Table 1. Primary studies selected in this SMS

ID	Title	Ref.
S1	Non-functional requirements for machine learning: Challenges and new directions	[Horkoff 2019]
S2	Requirements engineering for machine learning: Perspectives from data scientists	[Vogelsang and Borg 2019]

Continued on next page

[4]http://www.sciencedirect.com
[5]https://onlinelibrary.wiley.com
[6]Available at: https://docs.google.com/spreadsheets/d/1P4G0Q42zsKUzd0P5YeaArB2Fq9ei-3w32op-GautY8c/edit?usp=sharing

Table 1 – *Continued from previous page*

Id	Reference	
S3	Toward requirements specification for machine-learned components	[Rahimi et al. 2019]
S4	Towards Requirements Specification for Machine-learned Perception Based on Human Performance	[Hu et al. 2020]
S5	Requirements Engineering for Actors-with-Learning: Encompassing the Two Kinds of Modeling for Full Cognitive Cycle RE	[Yu 2021]
S6	Modeling machine learning requirements from three perspectives: a case report from the healthcare domain	[Nalchigar et al. 2021]
S7	Towards Artefact-based Requirements Engineering for Data-Centric Systems	[Chuprina et al. 2021]
S8	An MDE method for improving deep learning dataset requirements engineering using alloy and UML	[Ries et al. 2021]
S9	Use of a i*extension for Machine Learning: a real case study	[Barrera et al. 2022]
S10	A Data Modeling Method for Machine Learning Systems	[Shao and Wang 2022]
S11	Toward a Goal-Oriented Argumentation Approach for Fair ML Measures Using i*	[Sothilingam and Yu 2023]
S12	Requirements modeling: A use case approach to machine learning	[Uysal 2023]
S13	A Multi-layered Collaborative Framework for Evidence-driven Data Requirements Engineering for Machine Learning-based Safety-critical Systems	[Dey and Lee 2023]
S14	An extension of iStar for Machine Learning requirements by following the PRISE methodology	[Barrera et al. 2024]
S15	Agile-based Requirements Engineering for Machine Learning: A Case Study on Personalized Nutrition	[Cunha et al. 2024]
S16	Documentation of Non-Functional Requirements for Systems with Machine Learning Components	[Bajraktari et al. 2024]
S17	Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach	[Villamizar et al. 2024]
S18	An empirical investigation of challenges of specifying training data and runtime monitors for critical software with machine learning and their relation to architectural decisions	[Heyn et al. 2024]
S19	RM4ML: requirements model for machine learning-enabled software systems	[Yang et al. 2025]
S20	Requirements Representations in Machine Learning-Based Automotive Perception Systems Development for Multi-party Collaboration	[Saeeda et al. 2025]
S21	Towards a Framework for Operationalizing the Specification of Trustworthy AI Requirements	[Villamizar et al. 2025]
S22	Causal models for specifying requirements in industrial ML-based software: A case study	[Heyn et al. 2026]
S23	From Machine Learning Documentation to Requirements: Bridging Processes with Requirements Languages	[Peng et al. 2026]

3. Results

This section presents the main results of our SMS. First, we present an overview of the selected studies, then the main approaches for specifying requirements of ML-based systems, and finally, the tools used for requirements specification in these systems.

3.1. Overview of Studies

As illustrated in Figure 2, the first study was published in 2019, indicating that this is a new research topic. Specifically, 15 studies were published in conference proceedings, seven were published in journals, and one was included as a book chapter. In other words, two-thirds of the studies were published in conference proceedings, suggesting that this research theme is yet to be more consolidated.

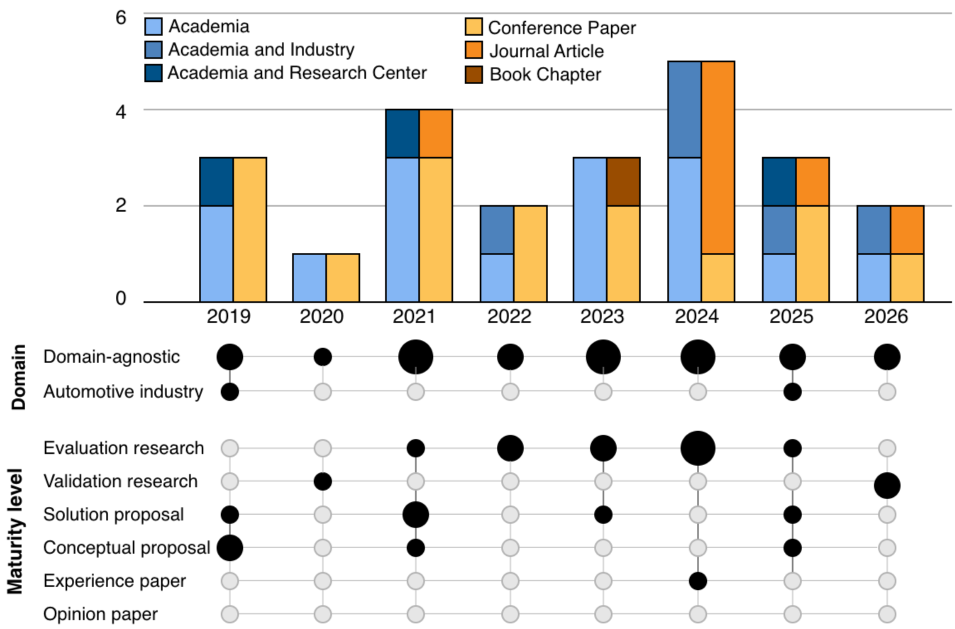

Figure 2. Overview of primary studies distributed over the years.

The studies were also classified considering the affiliation of the authors. To this end, we checked whether the authors worked in academia, industry, or research centers. As shown in Figure 2, 15 studies were conducted in an academic context; five involved a collaboration between academia and industry, and three resulted from cooperation between academia and research centers. Figure 2 also highlights that the collaboration between industry and academia has been more active in the last years, which could indicate a trend for the coming years. Regarding the application domains or contexts addressed by the studies, we identified that the automotive was the only application domain mentioned, with two studies. The other 21 selected studies had no specific application domain; thus, they were categorized as domain-agnostic.

The studies varied in characteristics and research approaches, affecting their maturity and indicating the suitability of the proposed approaches for future work. Our evaluation of the maturity level of studies consisted of analyzing the research type and evaluation method used in each study. Research types can be [Wieringa et al. 2006]: (i) *Evaluation research*, investigating a problem in practice and providing an implemented solution (e.g., case studies or field studies); (ii) *Validation research*, reporting a proposal not yet implemented in practice (e.g., simulations, mathematical analyses, prototyping, experiments); (iii) *Solution proposal*, is referred to a novel solution or considerable extension of an existing solution, although not completely validated (e.g., application example or proof-of-concepts); (iv) *Conceptual proposal*, presenting a new view to solving a specific problem, including a conceptual framework or taxonomy; (v) *Experience paper*, reporting personal experiences and lessons learned from a real-world project; and (vi) *Opinion paper*, reporting the personal opinion of the authors regarding a specific technique or tool and its development, without validation. Although solution proposals (Conceptual proposal, Experience paper, and Opinion paper studies) are classified as less mature, they offer important contributions to introducing new ideas and results that can serve as a foundation for future work. And those classified as Evaluation research and Validation research are considered more mature, offering empirical validation, systematic and rigorous approaches to provide evidence of results and ensuring their validity.

Figure 2 shows that the level of maturity of the studies. Studies that did not propose any approach were also classified regarding their level of maturity: two studies of 2019 (S1 and S2) were classified as a conceptual proposal, one of 2024 (S18) was classified as an experience paper, and one of 2025 (S20) was also classified as a conceptual proposal. Moreover, it can be observed that the level of maturity of the studies has increased over the years, with two validation research (S22 and S23) and a higher concentration of evaluation research studies proposing or extending approaches in the last five years (S9, S10, S12, S13, S14, S15, S16, and S19). However, it can also be seen that less mature work (such as experience papers, conceptual proposals, and solution proposal studies) is being done during the same period (S11, S18, and S21). Additionally, it is noticed that less mature work was conducted in the first three years, with a higher concentration of conceptual proposal and solution proposal studies (S3, S5, S7, and S8), and two studies classified as the more mature type, one validation research study (S4) and one evaluation research study (S6). Therefore, this analysis indicates that the maturity of studies has somewhat increased, yet the research topic remains in constant exploration, suggesting it is still not fully mature.

3.2. Approaches for the Specification of Requirements of ML-based systems

Among the 23 studies, 19 proposed (or extended) an approach to support the requirements specification for ML-based software systems, contributing to answer RQ1. Most of the studies have brought extensions to the traditionally used RE approaches (11 studies), as shown in the upper part of Figure 3, and seven studies that proposed new approaches addressing the requirements specification for ML-based software systems, shown at the bottom part of Figure 3. This figure also shows the number of approaches over the years, which peaked in 2024, with five proposals or extensions.

ML-based software systems differ from traditional software systems since their behavior is directly dependent on the data used to train the models; this data-centric nature leads to a non-deterministic behavior that may change and evolve when the model is trained or retrained based on the input data received. This characteristic brings difficulties when applying traditional approaches to the requirements specification for these systems, as reported by the studies that proposed extensions to traditional requirements specification approaches. Another reported challenge is that these systems are rarely defined systematically following a SE methodology (S8), which leads to a high dependence on the knowledge and expertise of data scientists and analysts to correctly translate requirements into concepts, features, and ML metrics (S14). Besides that, since no approaches are widely established, traditional ones have been used, although they are not capable of adequately addressing the specification of these systems because they usually focus on different constructs, such as classes and layers instead of focusing on the data processing and the development and deployment of models (S9, S12).

Figure 3 shows that goal-oriented requirements engineering (GORE) combined

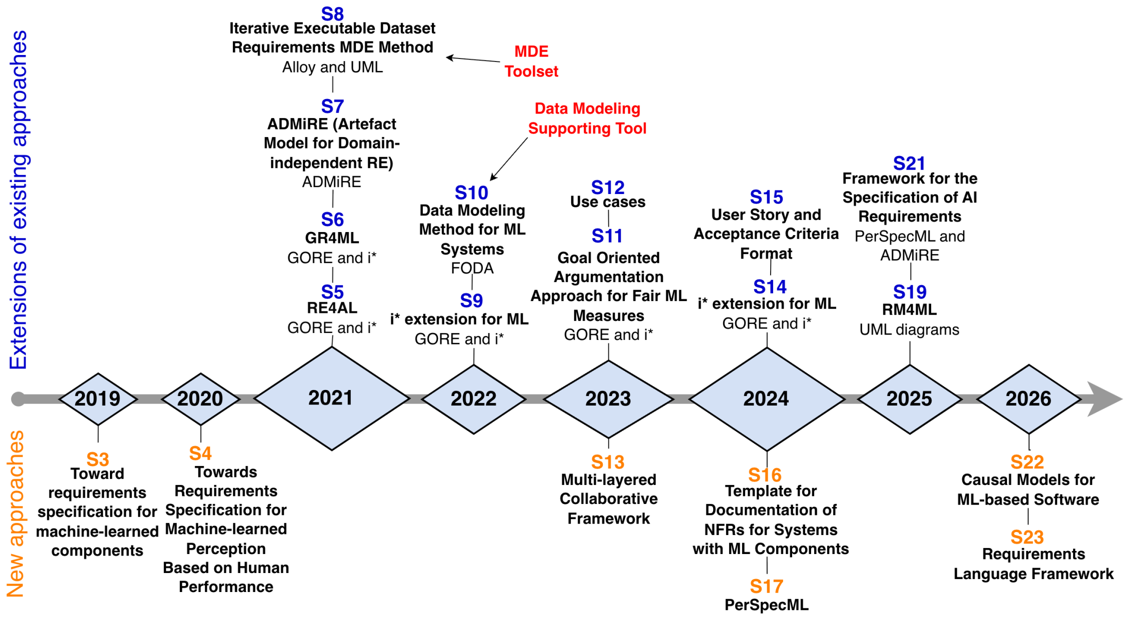

Figure 3. Approaches for the specification of requirements of ML-based Systems.

with the i* framework was the most extended approach (by S5, S6, S9, S11, and S14). For instance, S6 proposed GR4ML (Goal Requirements for ML) framework, consisting of three complementary modeling views (business, analytics design, and data preparation) that mediate the viewpoints of the stakeholders involved in developing an ML-based software system, aiming to reduce the conceptual distance among them. Modeling the solution in terms of goals and i* elements, this study provides a set of design catalogs and patterns and a user story template to facilitate the elicitation of elements in the business view. Other studies also proposed extensions: combining i* and GORE elements to filter non-valid configurations with metrics related to the goals (S9); assessing and measuring the impact of individual and group fairness of ML-based systems improving the support for decision-making processes (S11); and mitigating the gap between domain experts and ML developers with a comprehensible requirements language to specify the goals the project must meet while connecting it to ML concepts (S14). Additionally, S5 proposed an agent-oriented requirements modeling approach to integrate goal-oriented models with ML models and align the two sides of a cognitive cycle, perception-to-knowledge (ML insights) and knowledge-to-action (decision-making), by extending the GORE and i* approach to incorporate the learning capabilities of those data-driven systems and treating actors as cognitive agents, to blend human and machine intelligence.

The approaches based on the Unified Modeling Language (UML) were the second most extended (S8, S12, and S19). Two of the studies (S12 and S19) presented UML use case extensions. S12 proposed an approach that consists of four steps for the RE of ML-driven software systems (MLD-SS) with the use of use case modeling and use case textual descriptions in the requirements specification step to provide a stakeholder-centered modeling approach, allowing not only better communication, but also support for testing, verification, and validation processes. S19 followed the same idea but proposed extensions to three UML diagrams (class, state, and sequence diagrams). The class diagram was extended to describe the environment by combining the data quality requirements description with the environment classes. Furthermore, in addition to the extension of the traditional UML diagrams, the study also proposes a new diagram, named the machine learning diagram, which consists of two parts: the first is composed of the ML model and its evaluating indicators, and the second is related to the data requirements, including inherent data required by the model, and domain-related data. S8 also extended the class diagram to specify the dataset requirements of DL-based systems; the main goal of this approach is to execute the dataset requirements (through the Alloy Analizer tool) to improve their specification, allowing for the previous detection of inconsistencies in dataset requirements before the DL model training phase, as reported by the study.

Extensions of other approaches were also found. S7, S10, and S15 agreed that conventional methods are not capable of adequately capturing the specific requirements associated with data characteristics, which could compromise the quality of ML models, as their behavior is driven by the data. Addressing these challenges, S7 proposes an artifact-based approach to the RE, extending the ADMiRE model (which is composed of three layers, namely Context, Requirements, and System Layer) with a new layer focused on data (the Data Centric Layer) to capture the requirements at a lower abstraction level. S10 introduced a two-layered modeling approach: the learning context layer, which describes the elements of the ML system and its environment, and the properties-based specification layer, which defines detailed requirements, such as data quality and structure, incorporating the Feature-Oriented Domain Analysis (FODA) approach to represent mandatory, optional, and alternative characteristics of the data. S15 focused on agile methodologies, proposing an extension of the user story and acceptance criteria format approaches. The focus of the extension is to provide adequate, well-known approaches to agile development and ML-based software development, with a user-centric perspective, incremental evolution of requirements, measurable and testable acceptance criteria, and improved traceability and management of the requirements. Additionally, it was reported that the acceptance criteria must be continuously validated, as changes to data-related aspects could alter the system's behavior. S7, S10, and S15 reported that their proposed approaches can improve the traceability of requirements and their integration into the development process.

The bottom part of the same figure shows the studies that proposed new approaches to the requirements specification for ML/DL-based software systems. A reported issue (by S3 and S4) is that traditional approaches do not properly specify the behavior of ML components, which could lead to undesired outputs because their behavior is determined by the dataset used. S4 also reports that specifying hard-to-specify concepts (e.g., a pedestrian in a pedestrian detection system) makes the process even more challenging, since it could pose safety risks when such concepts are misspecified. S3 also agrees that although the main focus is on specifying functional and non-functional requirements for machine-learned components (MLC), more attention must be paid to specifying specific data requirements.

To this end, S3 proposed an approach to identify and visualize domain-specific concepts through the various stages of development and deployment of an MLC for Advanced Driving Assistance Systems (ADAS) in the automotive industry domain, focusing on extracting universally accepted benchmarks for these hard-to-specify concepts to find and fill the gaps in the dataset associated with and on the ML model constructed. S4 reports that one of the most-studied properties that guarantees safety in MLC-based systems is robustness to small changes; however, the concept of ”small changes” is not well-defined. Therefore, the study proposed an approach to ensure the robustness of ML-based systems, reporting that robustness requirements are crucial to guarantee that ML decisions can be trusted in safety-critical contexts. The approach focuses on reporting changes and transformations in the data, using a set of value parameters to measure this variation and determine the limitations the MLC must be robust to. S13 proposes a multi-layered framework for data RE that relates diverse types of stakeholders across its layers: the Problem Layer, Data Layer, and Evidence Layer. Each of its layers is different in terms of engineering activities, artifacts, and stakeholders involved. The requirements specification occurs in the Data Layer, where a structured process is followed to ensure its traceability and verifiability.

In addition, agreeing with the previous works, S16 not only reports that requirements engineering approaches for ML-based systems are still not completely comprehended but also brings attention to the non-functional requirements (NFR) in the context that systems with MLCs have their own NFRs, which are used to define their relevant details, like quality aspects of training datasets, retrainability of the ML model, and other specific aspects of the ML training pipeline. To fill this gap, the study proposes a template for the requirements specification of NFRs for ML-based systems to ensure consistent documentation of these systems’ NFRs.

Other challenges to the engineering of ML-based systems consist of how to adequately align the (sometimes unrealistic) expectations of various stakeholders (i.e., customers and managers) about the capabilities of ML and how to connect business values to data science and engineering activities, which are composed of multi-disciplinary teams, as outlined by S17. To tackle these challenges, the study proposed a perspective-based approach (PerSpecML) for the requirements specification for ML-based systems by analyzing 60 concerns related to common practices found in ML projects grouped into five perspectives: system objectives, user experience, infrastructure, model, and data, to provide a structured way to analyze and address different aspects of the systems.

Additionally, S22 proposed a novel approach to requirements specification for industrial ML-based systems by utilizing causal models, specifically Directed Acyclic Graphs (DAGs) and Structural Causal Models (SCMs). This approach addresses the inherent ambiguity of natural language and the difficulty of linking high-level system behavior to the specific data requirements needed for training and validation. By formalizing domain knowledge through causal structures, the approach allows for the identification of relevant features, ensuring that the data used for ML models aligns with the intended system requirements.

Aiming to bridge the gap between ML documentation and RE through the use of standardized templates and specialized requirements languages, S23 proposed an approach to transform informal data science artifacts, such as model cards and data sheets, into formal specifications understandable to software engineers. By providing a structured framework for documenting ML-specific properties, including data provenance, model limitations, and intended use, the approach aligns stakeholders’ expectations and improves the traceability of non-deterministic behaviors throughout the system lifecycle.

Following a different perspective from those studies that extended conventional approaches to provide means for the requirements specification for ML-based software systems. Focusing on the operationalization of trustworthy AI-enabled systems, S21 proposed a framework as an extension of the existing approaches ADMiRE and PerSpecML (presented by S7 and S17, respectively). This integrated approach leverages AMDIRE's structured, artifact-centric methodology and extends it with the multi-perspective guidance provided by PerSpecML to address the data-driven nature of ML-enabled systems. The framework explicitly focuses on mapping high-level ethical principles, such as transparency and robustness, into concrete technical specifications through structured templates and perspective-based analysis. By centering the specification process on the creation of specific artifacts, the approach ensures consistency and traceability among stakeholders throughout the development life cycle, bridging the gap between abstract trustworthiness goals and technical implementation.

Additionally, four studies (S1, S2, S18, and S20) adopted different approaches to this research field, making significant contributions by investigating its practical applications and challenges. S1 and S2 emphasize the need for a paradigm shift from logic-based, deterministic rules to data-driven, probabilistic measures, highlighting that both functional and non-functional requirements must be redefined to accommodate the data-driven nature of such systems, and outlining challenges and new research directions. This empirical focus was central to S18 and S20, which conducted practitioner interviews to identify real-world challenges and the specific approaches currently used in industrial settings. While S18 investigated safety-critical applications to link training data requirements with operational goals, S20 explored the automotive domain to identify how natural language, ODDs, and statistical KPIs serve as standards in multi-party collaboration.

3.3. Tools for the Specification of Requirements of ML-based systems

From the 19 studies that proposed (or extended) an approach to the specification of requirements of ML-based systems, only two studies (S8 and S10)also provided a tool (or toolset) to ease the requirements specification process. These studies are shown in the upper part of Figure 3 (with the name of the proposed tool (or toolset) in red), which indicates that they proposed an extension to an existing approach for the requirements specification for conventional software systems.

S8 provided a toolset of four tools to support the proposed approach for specifying and validating the structural requirements of datasets for DL-based systems. The first tool is a graphical editor for modeling the DRCM, developed using the Sirius framework (Viyovic et al., 2014), which allows the creation of visual models in UML-compatible notation. With this tool, the modeled DRCM can automatically generate the FDRCM from its contents using the XTend language. The second tool is a textual editor for the FDRCM, using the Alloy Analyzer to complete the specification of the FDRCM generated by the previous tool, in Alloy. The third tool is responsible for executing the model in Alloy using Kodkod, a SAT-based constraint solver, which allows execution of models and provides instances that satisfy the specifications within the context of an execution command. The fourth tool is an ad hoc visualizer of the data specifications generated by the Alloy formal engine that interprets the results of the specification instances generated by Kodkod and creates possible visual representations of the data based on the domain.

S10 proposed a supporting tool for specifying data requirements in ML-based systems, built on the two-layer requirements modeling method previously described. The tool consists of two main modules: the knowledge base and the interactive interface. The knowledge base is organized in two layers, storing structured information in JSON format derived from the proposed approach, consisting of a metadata model (lower layer) to describe the learning context—including both inherent elements and elements specific to the ML system—and a set of property-based specifications (upper layer), which represent the expected data requirements. This structure uses the FODA technique to organize elements into feature models with mandatory, optional, or alternative relationships within a domain. The interface guides the user through selecting characteristics and defining data properties, using a hierarchical visualization of the selected elements and automatically generating the requirements specification. The interface includes three modules: feature selection, hierarchical output of selections, and output of the property-based specifications. The interface's guidance logic is based on a depth-first search of the feature tree, with recommendations adapting to the user's choices.

4. Discussion

This section presents the main findings, future work perspectives, and the main threats to the validity of this work.

4.1. Main findings

The main findings from the insights gathered through the conduct of this SMS are:

- **Research topic under development:** The distribution of studies along the years (presented previously in Figure 2) reveals that the first study was published only a few years ago (in 2019). Considering the analysis of studies, it was noticed a lack of collaboration between the industry and academic environments, with only five studies presenting such collaboration, although ML-based systems are being constantly developed in industrial settings. This limited industrial engagement could be hindering the field's maturity, as the lack of real-world validation often keeps proposals at the conceptual or theoretical stage. The analysis on the maturity of studies also suggests that this research topic is still under development, with more mature studies being published only in recent years. At the same time, the constant production of less mature studies during the same period indicates that a consensus on better approaches for the requirements specification for ML-based systems is yet to be achieved.
- **Novel approaches proposed:** Most of the studies retrieved from the scientific literature focused on extending or proposing approaches to bridge the gap between the requirements specification for traditional and ML-based software systems and their unique characteristics. Approaches usually addressed to conventional software systems, such as GORE and UML-based requirements specification, have been extended to incorporate the data-centric nature of ML, since they were not initially designed to handle such non-deterministic behaviors. However, many studies still noted that such approaches were not capable of properly addressing the data-driven nature of ML, leading to the proposal of novel means to address the requirements specification for such systems. The motivation for proposing these novel approaches was the insufficiency of traditional, logic-based specifications to accurately capture

probabilistic behaviors and data-related constraints. To this end, it was reported that such novel means aim to solve this problem by providing a more rigorous and structured representation of ML components and serving as a shared language, facilitating communication between diverse stakeholders typically involved in such projects.

- **Supporting tools:** From the 19 studies that proposed or extended an approach, only two (S8 and S10) provided a specific tool or toolset to facilitate the specification process. In short, S8 introduced a toolset comprising a graphical editor, a textual editor, and a visualizer to support the modeling and validation of dataset requirements, while S10 proposed a supporting tool with a knowledge base and an interactive interface to guide users in selecting data characteristics. For the remaining studies, the lack of tools suggests that specialized software is not strictly necessary for adopting the proposed techniques. Many of these approaches rely on existing techniques, such as UML diagrams, user stories, and natural language templates, which can be managed using existing modeling and documentation infrastructure. However, the lack of integrated tools specifically designed to address ML-related complexities remains a potential research opportunity, especially in large-scale industrial environments.
- **Agile development-based approaches:** The absence of requirements specification approaches specifically tailored for integration with agile software development methodologies was also noticed. Despite the widespread adoption of agile practices in industrial settings, only study (S15) directly proposed an extension focused on agile formats, specifically on adapting user stories and acceptance criteria. This lack of specialized approaches is concerning, given that agile environments prioritize incremental evolution and frequent updates, which could bring value for the management of the evolving, data-centric nature of ML models. There are, therefore, several opportunities for future research to develop and evaluate specification methods integrated into agile workflows of ML-based systems projects.

4.2. Future perspectives

Through this SMS, we identified different opportunities for future research that could advance the state of the art in the field. Some of the main ones are:

- **Collaboration with industry**: Since there is a significant gap between academic proposals and real-world industrial applications. Robust partnerships to validate the approaches in complex, real-world settings would benefit the field's maturation, enabling the transition from conceptual frameworks to standardized, rigorous practices integrated into the software development life cycle; and
- **Proposal of agile development-based approaches**: Since only one study proposed a specifically agile-based approach, this research gap leads to an opportunity for further exploration. Proposing additional approaches that integrate seamlessly with agile practices would benefit the field's evolution and possibly facilitate their practical application in industrial development of ML-based software systems.

4.3. Threats to validity

The main threats to the validity of this work and the countermeasures are:

- **Missing studies.** To mitigate the risk of missing important studies during the study's search, the search string was iteratively refined through pilot searches, including synonyms and alternative terms. Furthermore, five major digital libraries widely used in SE (Scopus, IEEE Xplore, ACM DL, ScienceDirect, and Wiley) were scrutinized to ensure a broader coverage of the scientific literature. Regarding the selection phase, there is a risk that relevant studies were excluded due to insufficient details in the titles or abstracts. To address this, an SMS protocol with well-defined IC and EC was followed, and all studies with unclear information were re-evaluated through partial or full-text reading. Additionally, a double-validation process was applied, with an expert also validating the selection of these studies.
- **Validity of the data.** There is a risk of incorrect or incomplete data extraction due to inconsistencies or lack of clarity in how the information was presented by the primary studies. To mitigate this risk, a standardized data extraction form was developed to obtain detailed information to answer the research questions. This form was previously validated by an expert in the field to ensure that no fields were missing during the planning phase of this SMS. During the conduct phase, the extracted data were continuously validated to ensure their correctness and completeness and to mitigate the risk of biased data interpretation.
- **Validity of the research.** To ensure the replicability of our study, we defined the research protocol followed based on the well-established three-phase process for SMS [Kitchenham et al. 2015]. This structured process was essential to mitigate the threats inherent to the qualitative and analytical nature of this study.

5. Conclusions

ML-based software systems are increasingly being developed across several domains to deliver value and innovation. The probabilistic and data-centric nature of such systems is posing significant challenges for requirements specification, where consensus on appropriate approaches has yet to be reached. This study presented the state of the art on how requirements specification is addressed in ML-based systems. Based on 23 studies identified in our work, we found that this research field is very new, with the first study published in 2019, and that collaboration with the industry remains limited. We also observed that, while extensions of traditional approaches exist, new ones seem to be indeed necessary. We also identified a gap in integrating such approaches into agile methodologies. Hence, the future perspectives for the field are promising with several open research opportunities that should be surely investigated to ensure the quality of those increasingly large and critical software systems that rely on ML models.

Acknowledgment

This study was funded by FAPESP (2023/00488-5, 2024/00292-6, 2024/20302-6) and CNPq (313245/2021-5).

References

Bajraktari, E., Krause, T., and Kücherer, C. (2024). Documentation of non-functional requirements for systems with machine learning components. volume 3672.

Barrera, J. M., Reina-Reina, A., García-Ponsoda, S., and Trujillo, J. (2022). Use of a i*extension for machine learning: a real case study. volume 3231, page 14 – 20.

Barrera, J. M., Reina-Reina, A., Lavalle, A., Maté, A., and Trujillo, J. (2024). An extension of istar for machine learning requirements by following the prise methodology. *Computer Standards and Interfaces*, 88.

Chuprina, T., Mendez, D., and Wnuk, K. (2021). Towards artefact-based requirements engineering for data-centric systems. volume 2857.

Cunha, C., Oliveira, R., and Duarte, R. (2024). Agile-based requirements engineering for machine learning: A case study on personalized nutrition. *International Journal of Intelligent Systems and Applications in Engineering*, 12(2):319 – 328.

Dey, S. and Lee, S.-W. (2023). A multi-layered collaborative framework for evidence-driven data requirements engineering for machine learning-based safety-critical systems. page 1404 – 1413.

Dyba, T., Kitchenham, B. A., and Jorgensen, M. (2005). Evidence-based software engineering for practitioners. *IEEE Software*, 22(1):58–65.

Heyn, H.-M., Knauss, E., Malleswaran, I., and Dinakaran, S. (2024). An empirical investigation of challenges of specifying training data and runtime monitors for critical software with machine learning and their relation to architectural decisions. *Requirements Engineering*, 29(1):97 – 117.

Heyn, H.-M., Mao, Y., Weiß, R., and Knauss, E. (2026). Causal models for specifying requirements in industrial ml-based software: A case study. *Journal of Systems and Software*, 232.

Horkoff, J. (2019). Non-functional requirements for machine learning: Challenges and new directions. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 386–391.

Hu, B. C., Salay, R., Czarnecki, K., Rahimi, M., Selim, G., and Chechik, M. (2020). Towards requirements specification for machine-learned perception based on human performance. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 48–51.

Kitchenham, B., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*, volume 4. CRC Press.

Nalchigar, S., Yu, E., and Keshavjee, K. (2021). Modeling machine learning requirements from three perspectives: a case report from the healthcare domain. *Requirements Engineering*, 26:237–254.

Nascimento, E. d. S., Ahmed, I., Oliveira, E., Palheta, M. P., Steinmacher, I., and Conte, T. (2019). Understanding development process of machine learning systems: Challenges and solutions. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–6.

Peng, Y., Heyn, H.-M., and Horkoff, J. (2026). From machine learning documentation to requirements: Bridging processes with requirements languages. *Lecture Notes in Computer Science*, 16361:119–136.

Rahimi, M., Guo, J. L., Kokaly, S., and Chechik, M. (2019). Toward requirements specification for machine-learned components. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 241–244.

Ries, B., Guelfi, N., and Jahić, B. (2021). An mde method for improving deep learning dataset requirements engineering using alloy and uml. page 41 – 52.

Saeeda, H., Rohacova, Z., Jakobsson, O., Heyn, H.-M., Knauss, E., Knauss, A., and Horkoff, J. (2025). Requirements representations in machine learning-based automotive perception systems development for multi-party collaboration. *Lecture Notes in Computer Science*, 15588:197–213.

Shao, W. and Wang, X. (2022). A data modeling method for machine learning systems. page 1 – 5.

Sothilingam, R. and Yu, E. (2023). Toward a goal-oriented argumentation approach for fair ml measures using i*. volume 3533, page 4 – 9.

Uysal, M. P. (2023). *Requirements modeling: A use case approach to machine learning*.

Villamizar, H., Escovedo, T., and Kalinowski, M. (2021). Requirements engineering for machine learning: A systematic mapping study. In *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 29–36. IEEE.

Villamizar, H., Kalinowski, M., Lopes, H., and Mendez, D. (2024). Identifying concerns when specifying machine learning-enabled systems: A perspective-based approach. *Journal of Systems and Software*, 213.

Villamizar, H., Mendez, D., and Kalinowski, M. (2025). Towards a framework for operationalizing the specification of trustworthy ai requirements. In *IEEE 33rd International Requirements Engineering Conference Workshops (REW)*, page 533–537.

Vogelsang, A. and Borg, M. (2019). Requirements engineering for machine learning: Perspectives from data scientists. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 245–251.

Wieringa, R., Maiden, N., Mead, N., and Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11:102–107.

Yang, Y., Zeng, B., and Gao, J. (2025). Rm4ml: requirements model for machine learning-enabled software systems. *Requirements Engineering*, 30(1):1 – 33.

Yu, E. (2021). Requirements engineering for actors-with-learning: Encompassing the two kinds of modeling for full cognitive cycle re. volume 2857.