Requirements Engineering for Artificial Intelligence Systems: A Systematic Mapping Study

Khlood Ahmad[a], Mohamed Abdelrazek[a], Chetan Arora[b], Muneera Bano[c] and John Grundy[b]

[a]*Deakin University, Geelong, VIC, Australia*
[c]*CSIRO's Data61, Clayton, VIC, Australia*
[b]*Monash University, Clayton, VIC, Australia*

ARTICLE INFO

Keywords:
requirements engineering
software engineering
artificial intelligence
machine learning
systematic mapping study

ABSTRACT

[Context] In traditional software systems, Requirements Engineering (RE) activities are well-established and researched. However, building Artificial Intelligence (AI) based software with limited or no insight into the system's inner workings poses significant new challenges to RE. Existing literature has focused on using AI to manage RE activities, with limited research on RE for AI (RE4AI). [Objective] This paper investigates current approaches for specifying requirements for AI systems, identifies available frameworks, methodologies, tools, and techniques used to model requirements, and finds existing challenges and limitations. [Method] We performed a systematic mapping study to find papers on current RE4AI approaches. We identified 43 primary studies and analysed the existing methodologies, models, tools, and techniques used to specify and model requirements in real-world scenarios. [Results] We found several challenges and limitations of existing RE4AI practices. The findings highlighted that current RE applications were not adequately adaptable for building AI systems and emphasised the need to provide new techniques and tools to support RE4AI. [Conclusion] Our results showed that most of the empirical studies on RE4AI focused on autonomous, self-driving vehicles and managing data requirements, and areas such as ethics, trust, and explainability need further research.

1. Introduction

The increase in the volume and velocity of the data and the need for automation have made Artificial Intelligence (AI) a practical solution to many of the challenges we face today. This technology shift has allowed AI to become a favored software alternative to many organizations [1]. Software systems with an AI component are currently under demand and are being investigated in multiple domains ranging from automotive and self-driving cars [2], healthcare [3], science [4], virtual assistants chatbots [5, 6], predicting privacy decisions and providing privacy recommendations for IoT services [7]. The process of building AI-based software differs from traditional software development approaches, making it more challenging to apply existing techniques and tools [8, 9]. Traditional software engineering involves collecting requirements, analysis, and detailed designs to implement an executable program which mainly includes writing code [10]. On the other hand, engineering software with an AI component involves additional configuration aspects, such as data collection, selecting an appropriate algorithm (e.g., a machine learning or natural language processing algorithm), and training the model based on the desired input/output with relatively less emphasis on the source code writing [8, 11, 12, 13].

The configuration process in AI-based software, i.e., selecting and validating data and other activities, need to be specified in the requirements phase. However, RE for AI systems (RE4AI) is not as established as the traditional RE approaches for non-AI software, and AI systems usually lack precise requirements and proper RE techniques [14, 15]. The differences between the RE practices for traditional software (mostly deterministic) and AI-based software (mostly uncertainty-prone and black-box) create a need to adapt existing tools and RE methods [16].

AI has several definitions and encompasses several areas. John McCarthy, a pioneer of AI, defined AI in 1955 as "the science and engineering of making intelligent machines" [17]. The overall definition of AI is broad, and thus a number of sub-disciplines, such as Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), Expert Systems, and Robotics are considered under this broad umbrella term [17]. In context of this paper, due

✉ ahmadkhl@deakin.edu.au (K. Ahmad); mohamed.abdelrazek@deakin.edu.au (M. Abdelrazek); chetan.arora@monash.edu (C. Arora); muneera.bano@csiro.au (M. Bano); john.grundy@monash.edu (J. Grundy)
ORCID(s):

to the lack of substantive research on RE for any individual sub-discipline, we do not focus on specific sub-disciplines and consider the AI systems as a whole.

In this article, we conducted a systematic mapping study to identify existing empirical evaluations, emerging theories, and occurring limitations and challenges in RE4AI. We observed that most of the results focused on using AI to manage RE [18, 19, 20, 21], with little research supporting RE4AI. We also found most empirical studies focused on autonomous systems, with fewer empirical evaluations on ethics, explainability, and trust, and an apparent lack of research on managing data requirements and modeling requirements for AI systems.

We followed Kitchenham et al.'s [22] guidelines on performing systematic reviews and Petersen et al.'s [23] guidelines for conducting systematic mapping studies in software engineering to answer the following research questions (RQs):

RQ1. Which requirements engineering frameworks, notations, modeling languages, and tools have been proposed to build AI systems?
RQ2. Which evaluation methods are used to assess empirical studies in RE4AI?
RQ3. Which target application domains and AI areas of focus are considered in the existing approaches?
RQ4. What are the limitations and challenges of existing requirements engineering techniques when applied to AI systems?

We published preliminary results of our mapping study at the 29th IEEE Requirements Engineering (RE) conference [24]. The initial results were based on 27 studies published from 2010 to mid-2020. This article extends the results with an additional search between mid-2020 and mid-2021 to include 16 new research studies. One research question was added (RQ2), and R1, RQ3, and RQ4 were extended to include newer findings. RQ1 extends to include exiting frameworks and tools used to manage RE4AI, and the methods used to model requirements noticeably changed with the added studies. In RQ3, we included which evaluation methods were more dominant in specific application domains. And finally, RQ4 results remained relatively similar, with additional papers to support existing issues and challenges and one new issue emerging. Also, this article provides a more comprehensive and detailed search method and lists two new research recommendations.

The main research contributions of this mapping study include:

- We identify a list of 43 primary studies that focus on RE for AI systems. Out of these 43 studies, 30 conduct some form of empirical investigation of a relevant problem or validate a solution. The remaining 13 studies are 'non-empirical' papers that propose an idea or a model and do not conduct empirical investigation of any sort.
- We find that UML is the most popular modeling notation used, followed by domain specific modeling languages.
- We found (only) two tools used in RE4AI research, including the jUCMNav [25] and the Sirius framework [26].
- We find that the most popular application domains of RE4AI are autonomous driving and robotics. We further found that there are no or little empirical studies on ethics and trust aspects of RE4AI.
- We identify that the most discussed challenges in RE4AI research are related to issues with data requirements and calculating trade-offs when building AI software.

The rest of the paper is structured as follows: Section 2 provides a brief background on RE4AI. Section 3 presents details of our structured literature search and selection process on RE for AI systems. Section 4 reports results from the selected primary studies. Section 5 addresses threats to validity. Section 6 discusses key results and summarizes emerging theories, and Section 7 concludes.

2. Background and Related Work

2.1. Requirements Engineering

RE is arguably the most crucial phase in the software engineering lifecycle and plays a significant role in every stage of software development [27]. In RE, understanding stakeholders' requests are important, and requirements act as the communication channel between the system developers and the stakeholders [28]. RE acts as that channel to gather and document stakeholders' needs [29]. Therefore, it is vital to establish requirements early on when building software systems to ensure all stakeholders' needs and specifications are captured and documented correctly.

Koelsch defined requirements as “a need, desire, or want to be satisfied by a product or service" [30]. If this need or desire is not fulfilled, the product is not usable. An established process for RE is obtained to achieve this need and undergoes the following phases: elicitation, analysis, specification and documentation, validation, and management [31]. This process makes sure that requirements are extracted, documented, managed correctly, and comply with users’ needs. However, the software process differs when AI components are involved. Specifications do not always drive such systems, as parts of the system can be driven from data [32]. Therefore, RE techniques would need to be adjusted to the changes introduced by this new paradigm of AI systems.

In RE, requirements are classified to be either functional or non-functional. Functional requirements represent the system’s features and business rules to what the system should include. Whereas non-functional requirements include systems qualities and constraints [30]. Non-functional requirements (NFR) are more difficult to present [33]. However, several modeling languages and tools are available that can help display properties of non-functional requirements.

Modeling languages display WHY behaviors and functionalities are selected and WHAT capabilities are needed to support these choices. Modeling languages focus on the high-level abstraction aspect of the required system rather than the details of operations, which is helpful during the early stages of building software systems [34, 35]. There are different RE modeling languages available such as Goal-Oriented Requirements Engineering (GORE) and User Requirements Notation (URN)[34]. Goal modeling languages can help present requirements that showcase the stakeholders objectives of what is needed from the proposed system [36].

2.2. Requirements Engineering and AI-based Systems

When engineering software systems with an AI component, new processes are appearing, such as managing data, training the models and the design process [37, 38]. In SE, the ML code is relatively small compared to the actual process when building systems with ML components. Most of the work focuses on managing data, feature extraction, analyzing, configuring, etc. [8]. At Microsoft, the SE process is employed over a nine-step model. It includes: gathering requirements, collecting, cleaning, and labeling data, feature engineering, training and evaluating the model, and finally deploying and monitoring the model over time. Requirements are decided based on how feasible it is to implement features and find appropriate models for given problems [11].

Because of the difference in the development process, new challenges are appearing when managing requirements for AI software, some of these challenges include defining, eliciting, and specifying requirements. Kuwajima et al. [37] explains that the lack of requirements specifications in current Machine Learning (ML) systems significantly impacts the ML model’s quality. And that most ML models lack requirements specifications. One of the reasons for the difficulties in writing requirements specifications for AI software is the inconsistencies in inputs and outputs patterns. Several tools are used for traditional SE practices to manage code and other issues. However, because of the vast difference between AI software and traditional SE processes, it is hard to use these tools in managing such issues [8].

From the literature, it is evident that the existing tools for traditional SE practices (more specifically RE) cannot be appropriately utilized when building AI systems [8]. Also, including AI components in building software systems has impacted RE, and new requirements have appeared in the process, such as data, ethics, explainability, and trust. Some existing requirements have changed. For example, Non-functional requirements (NFR) for ML systems have changed to include transparency, trust, privacy, safety, reliability, and security [39]. From a RE perspective, we are facing a new set of challenges and questions when building AI systems [40]. Specifying and defining requirements has introduced new challenges in RE practices [41, 37]. This motivated us to analyze the existing body of work and collate the state of the art of empirical literature on RE4AI.

2.3. Related Work

RE4AI has gained traction from the research community in the past few years. Previous studies (on RE4AI or Software Engineering for AI (SE4AI) with RE recommendations) focused on showing the difference between traditional software and AI software [42], identifying issues and challenges in RE4AI, and reporting on research directions for future development [43, 44, 45, 46, 47]. Other studies investigated the use of non-functional requirements and issues related to non-functional requirements [39, 48], ethical requirements [49, 50, 51], explainability [52], and transparency [53]. A number of authors have investigated what methods and tools could be used in RE4AI. In [14], the study suggests a list of techniques that can be used for each RE phase when working with expert systems, and [54] lists a number of tools that can identify and mitigate any issues related to AI fairness and bias during RE.

We have further identified two relevant mapping studies. The first study identified 348 existing SE4AI studies and provided some insights on RE-related issues [55]. This study only presents a general overview and does not provide details on the current methods, frameworks, notations, or tools used to manage RE4AI. The second mapping study, conducted by Villamizar et al. [56], identified 35 RE for ML-related studies and provided an overview of some of the existing challenges, popular RE phases, and existing evaluations. The findings showed that 40% of the selected papers did not have any form of empirical investigation, similar to our results (discussed later). Our study complements the second mapping study, and sheds light on additional aspects of RE4AI (in addition to the second study), such as identifying existing methodologies, tools, and modeling notations used.

3. Systematic Mapping Study

This section shows our search strategy to extract relevant studies based on Kitchenham et al.'s [22] guidelines to conduct an SLR and Petersen et al.'s [23] guidelines for conducting systematic mapping studies in software engineering. The mapping study followed three stages to include planning, conducting, and reporting the review. The initial planning phase involved writing a protocol and identifying a set of research questions. The protocol[1] included a plan for a search strategy. To exhaust our exploration of any existing empirical evidence, we identified relevant keywords and search strings. The second stage involved identifying and analyzing existing primary studies to answer our research questions. The final step involved evaluating, reviewing and reporting the final document. We performed the search over two periods. The first period was from 2010 and mid-2020, and the second was from mid-2020 to mid-2021. We then combined the papers resulting from both searches as displayed in Figure. 1.

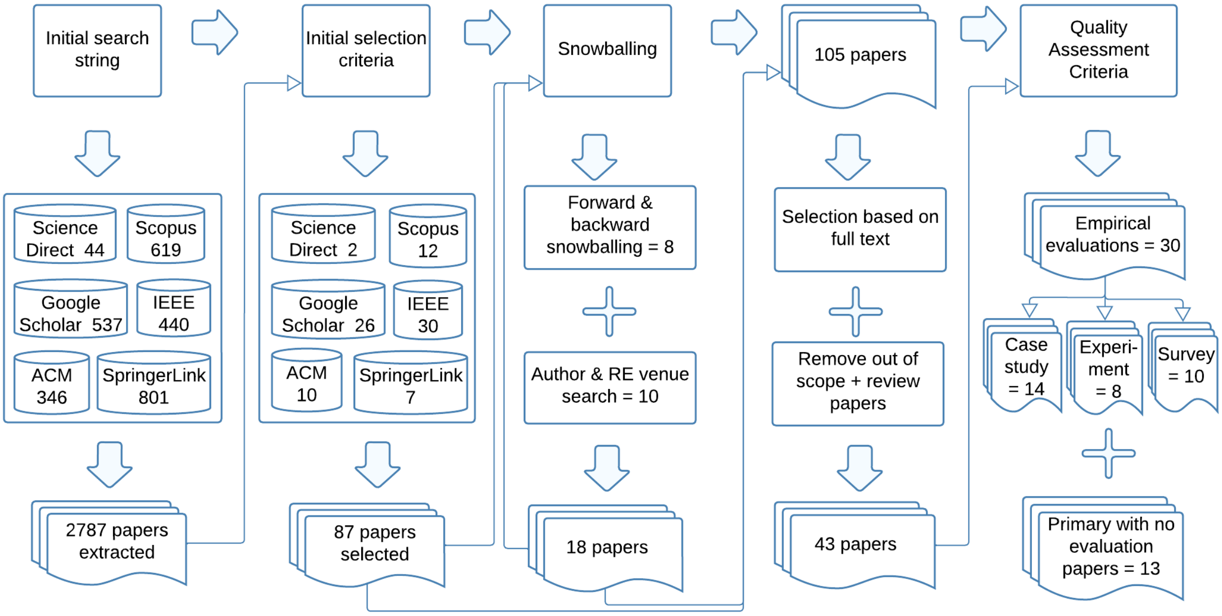

Figure 1: Paper extraction process

3.1. Search Strategy

Our research questions guided the identification of the main search terms to include "Requirements Engineering" and "Artificial Intelligence". We extracted synonyms and alternative words from each search term and customized a search string for each database. For both main search terms, we derived several alternative words as shown in Table 1. For "requirements engineering", the search terms were derived from existing research on RE [57, 58] and meetings conducted between the first and fourth authors to finalize the list of keywords used. Keywords selected for "Artificial Intelligence" were based on a combination of existing literature [59, 60] and meetings conducted between the first and second authors to decide on which keywords to include.

[1] Protocol Link

Table 1
List of Search Terms

Main Terms	Alternative terms
requirements engineering	requirements process, requirements elicitation, requirements gathering, requirements identification, requirements analysis, requirements validation, requirements verification, requirements specification, requirements development, requirements documentation, requirements management, requirements testing, requirements driven, functional requirements
artificial intelligence	machine learning, speech recognition, deep learning, natural language processing, computer vision, machine intelligence, chatbot, chat-bot, expert systems, self driving, autonomous, recommendation system, robot, AI, ML

We used boolean operators to link the key terms and their synonyms when connecting the search string. As a result, we formed the following search string:

> ON ABSTRACT ((“requirements engineering” OR “requirements process” OR “requirements elicitation” OR “requirements gathering” OR “requirements identification” OR “requirements analysis” OR “requirements validation” OR “requirements verification” OR “requirements specification” OR “requirements development” OR “requirements documentation” OR “requirements management” OR “requirements testing” OR “functional requirements” OR “requirements driven”) AND (“artificial intelligence” OR “machine learning” OR “expert systems” OR “deep learning” OR “computer vision” OR “natural language processing” OR “speech recognition” OR “machine intelligence” OR AI OR ML OR chatbot OR “expert systems” OR “self driving” OR autonomous OR “recommendation system” OR robot)).

We searched the following online databases: IEEE Explore, ACM Digital Library, Google Scholar, Science Direct, SpringerLink, and Scopus. For most databases, the search string was conducted based on abstracts. However, some databases allowed us to include titles and keywords, and others, such as Google Scholar, limited search results on abstract to the last year only. For Google Scholar, the initial search returned 17000 results. To narrow them down, we first performed a search based on abstracts, and to cover as many results as possible, we then completed another search on titles only. When entering the search string, we found that most online databases had constraints and limits. Most searches limited either the number of words used, the number of boolean connectors or the length of the string itself. Therefore, the search string was modified to accommodate each database's search criteria while keeping the logical order consistent. Initial tests on the search strings were carried out to check the resulting papers, and the results were verified by the second and fourth authors. When testing the search string we found that some of the papers that we had extracted prior to the search did not appear in our search. Thus, we kept modifying the search string and keywords until all known papers were found. For example, we could only find the paper [61] after including the keywords “requirements driven” to the search string.

3.2. Selection Criteria and Snowballing

The initial search resulted in 2,787 papers. To filter the results further, we performed an initial selection criteria to extract relevant studies. We assessed each article based on its title and abstract and performed an inclusion and exclusion assessment as shown in Table 2. A folder was created for each database, and papers that passed the inclusion criteria were given an identifier number and placed in the corresponding folder. A total of 26 papers were found in more than one database so we removed duplicates from the final list. For example, [39] was found in IEEE, Google Scholar, and Scopus, thus, we removed the duplicates found in Google Scholar and Scopus. For studies that reported several publications, we grouped them under the same identifier. After completing the initial selection criteria, we found a total of 87 studies.

Next, we performed backward snowballing on all the selected papers by examining the references and forward snowballing by checking the citations and authors [62]. The backward snowballing involved extracting relevant research from references. Whereas, the forward snowballing reviewed all the papers that cited the selected studies. We examined all the references and citations of the 87 articles resulting from the initial selection criteria and searched the author's Google Scholar profiles and the proceedings of RE-related publication venues. A total of 18 new papers were found

Table 2
Initial inclusion and exclusion assessment criteria

Inclusion criteria:	**Exclusion criteria:**
Papers published in English language	Review papers and secondary studies
Primary studies on requirements engineering for AI systems	Studies that used AI to manage or analyze requirements
Papers published at peer reviewed conferences and journals	Abstract-only papers – the full paper is not available or only found as an arXiv pre-print (Not published)
Full resource papers available	Exclude book chapters, magazines, general articles, project plans or theses

Table 3
Classifications used for selected primary studies

Classification	**Explanation**	**Example**
Non-empirical	Idea paper with no evaluation	Köhl et al. [63] present a conceptual analysis for explainability as a NFR and proposed to model them using "Softgoal Independency Graph SIG" with other NFR and minimize conflicts
	Propose a new model, or prototype with no evaluation	Aydemir and Dalpiaz [64] provide an "Ethics-Aware SE Method" but is not evaluated
Empirical	The evaluation component is used to validate the proposed RE4AI method, idea or tool	Nalchigar et al. provide a new "GR4ML" framework that is evaluated using a case study [65]
	The evaluation is used to propose a model or investigate an RE4AI problem,	Vogelsang and Borg [66] interviewed practitioners to investigate the challenges in RE for ML systems

during snowballing. The same backward and forward snowballing method was applied to the identified 18 papers. However, no new papers were identified in the process.

Next, the first author read the full text on all 105 papers to determine if they were relevant to our RQs. This selection process was carried out in several meetings among the first four authors to decide on which papers to include. Furthermore, papers that felt out of context were flagged by the first author and validated by the second and third authors. Out of these papers, we excluded 26 secondary papers, i.e., papers that report their findings based on existing primary studies, such as literature reviews. From the remaining studies we removed 23 of the papers as they were not relevant, did not answer our RQs, or were out of scope. We further excluded one paper, that on closer analysis focused on AI for RE, instead of RE4AI. Nine more papers were excluded because the focus of the proposed method and evaluation was on non-AI related tasks. For example, in [7], the paper is excluded as the proposed requirements are for the Internet of Things (IoT) aspect of the project and not the ML model used. Also, we excluded two arXiv papers that were not yet published. However, we did include papers that appeared as an arXiv pre-print but were peer-reviewed and accepted at a later dated conference. The remaining 44 papers were identified as primary studies. Two of these papers reported the same study under two different publications so we only included the most recent publication. Thus, a total of 43 papers were selected for quality assessment.

3.3. Quality Assessment

Of the total 43 papers, we classified them as 'empirical' or 'non-empirical' papers as shown in Table 3. We found that 13 of the primary studies did not have an evaluation component, i.e., they proposed an idea, model or a solution, but did not present any evaluation of the proposed artefact or left the evaluation for future work. We classified these papers as 'non-empirical' and used the quality assessment check in Table 5. The remaining 30 papers that had an evaluation component, e.g., conducted an experiment, survey or a case study, were classified as 'empirical', and we performed the quality assessment check based on Kitchenham's guidelines as shown in Table 4. We note that we did not discard any paper in quality assessment (due to the already limited number of studies on the topic), and the assessment helped in ranking the primary studies.

The different types of empirical investigations [67], found in 30 'empirical' studies are:

1. **Surveys**: These included collecting data from a selected population sample using methods such as questionnaires and interviews, and the main purpose of a survey was to answer questions to a given problem [68, 69]. We checked if the sample size was justified, inclusive and if any biases were evident in the selection process.

Table 4
Quality Assessment Criteria for Empirical Papers based on [22]. Each question is given a grade of either Yes, No or Partly (Y/N/P)

Study type	Question	Grade
General	Are the aims clearly stated?	Y/N
	Are the measures used clearly defined?	Y/N/P
	Is the methodology clearly described?	Y/N/P
	Data collection methods adequately described?	Y/N/P
	Does the report have any implication for practice?	Y/N/P
	Do the researchers explain the consequences?	Y/N/P
Survey	Was the sample size justified?	Y/N
	Is the sample representative of the population?	Y/N/P
	Is the survey likely to have introduced bias?	Y/N/P
	Is there a comparison or control group?	Y/N/P
	Evidence of selection biases in the group selected?	Y/N/P
Experiments	Was the sample size justified?	Y/N
	Are the research questions answered?	Y/N/P
	Were the experiments randomly allocated?	Y/N
	Choice of subjects influence the outcome?	Y/N/P
	Could lack of blinding introduce bias?	Y/N/P
Case study	Is the context of the case study defined?	Y/N
	Are the study participants adequately described?	Y/N/P
	Are sufficient raw data presented?	Y/N/P
	Are ethical issues addressed properly?	Y/N/P
	Is a clear Chain of evidence established from observations to conclusions?	Y/N/P

Table 5
Quality Assessment Criteria for Non-Empirical papers

Question	Grade
Does the study propose an idea or a model?	Yes/No
Are the aims clearly stated?	Yes/No
Are the measures used clearly defined?	Yes/No
Does the report have any implication for practice?	Yes/No

2. **Case studies**: Case studies included in-depth investigations to test a new or existing theory based on a set of research questions that aim to address the *how* and *why* a phenomenon works [70, 71]. Data collection methods in case studies included surveys and interviews with participants or using data from existing repositories [72]. We noted if the process had a clear chain of evidence and if the context was clearly defined. We also checked for any ethical issues related to participants or data selection. Are the participants adequately described, and if the data is adequately presented.
3. **Experiments**: Selected experiments involved testing a hypothesis with one or more independent variables against dependant variables. For example, testing out a newly proposed method against the traditional approach to carrying out a procedure [73]. We checked if the sample size was justified, randomly selected, without biases and if they answered the research questions.

For empirical evaluation, checklist items were graded by yes, no, or partial. Each item had a score of 1 for yes, 0 for no, and 0.5 for partial. After finalizing the quality assessment criteria, we selected any study that scored more than 4. The final score for empirical evaluations was out of 11. We labeled papers with a score between 4 and 6 as low. Medium for scores between 6.5 and 8.5. And high for scores between 9 and 11 as shown in Figure. 2. For papers that did not have an evaluation method, the quality was assessed based on the criteria shown in Table 5. We kept the papers that scored low on the quality assessment due to the limited available work. Also, we felt that each of these papers could contribute to our findings. A total of 30 empirical studies and 13 non-empirical studies were selected as shown in Table 6.

Table 6
Selected primary studies

Classification	Selected Papers
Empirical	[74, 32, 75, 76, 77, 78, 66, 79, 80, 61, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 65, 94, 95, 96, 97, 98, 99]
Non-empirical	[100, 64, 101, 63, 102, 103, 104, 105, 106, 107, 108, 109, 110]

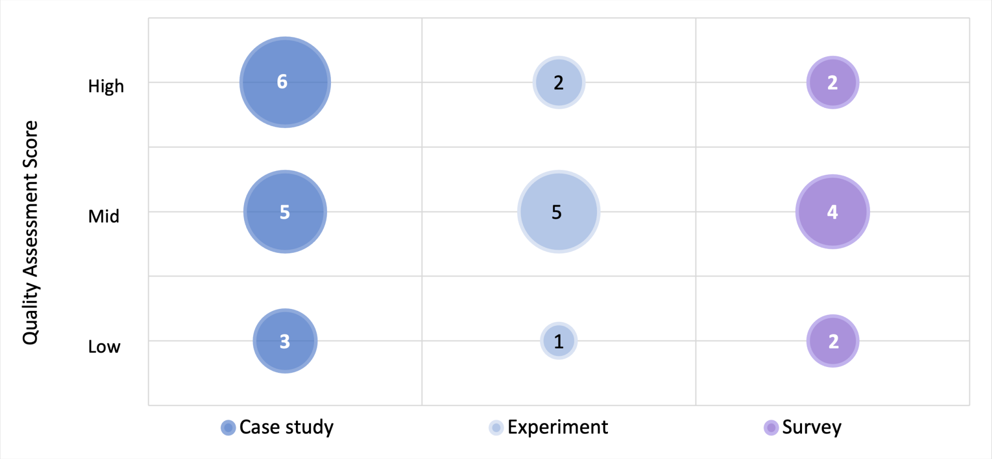

Figure 2: Distribution of the resulting empirical papers after performing the quality assessment score

3.4. Data Collection and Analysis

We had several meetings between the first, second, and fourth authors to decide on what information to extract from each of the selected studies. A shared Excel document[2] was created, and each paper was given a unique ID. We extracted publication details that included: title, authors' names, citation count, source, type (conference, workshop, or journal), and ranking. We then documented the type of the study, the proposed framework (if any), and the evaluation methodology, e.g., case study, experiment, survey, or a 'non-empirical' paper. We further extracted the data collection methods used in empirical studies and the modeling tools or requirements notations used in each paper. We further noted the application domain (e.g., medical and autonomous systems). Next, we listed the publication date for each paper to determine if there was a trend in the number of publications per year on RE4AI related research.

For data analysis, we conducted a 'theoretical' thematic analysis that was driven by the research questions (RQs) [111]. The codes were initially selected based on the RQs in a couple of meetings between the first and the second authors. Some of the codes emerged during the coding process. For example, we did not know what limitations or challenges existed in the literature, so the codes and themes were established as they appeared. Once the codes were established, we entered the selected 43 papers into Nvivo for coding. The main codes and sub-codes were created in Nvivo, and the first author read through each paper and used an open-coding procedure on each transcript to assign relevant text to its matching code, and the data was extracted with a top-down coding strategy [112]. The third author closely reviewed the coding results for a randomly selected sample of ≈25% papers. They expressed only minor disagreement on the coding results for one paper in terms of limitations and challenges. We discussed this, and however, no changes were made in the coding results after agreement. Once the coding process was completed, we combined relevant codes into themes, as shown in Figure. 3. The final themes were then presented and discussed in a number of meetings among the first, second, and third authors for review and analysis. We then used these themes to answer our RQs and present any emerging theories.

4. Results

[2] Data extraction for selected papers

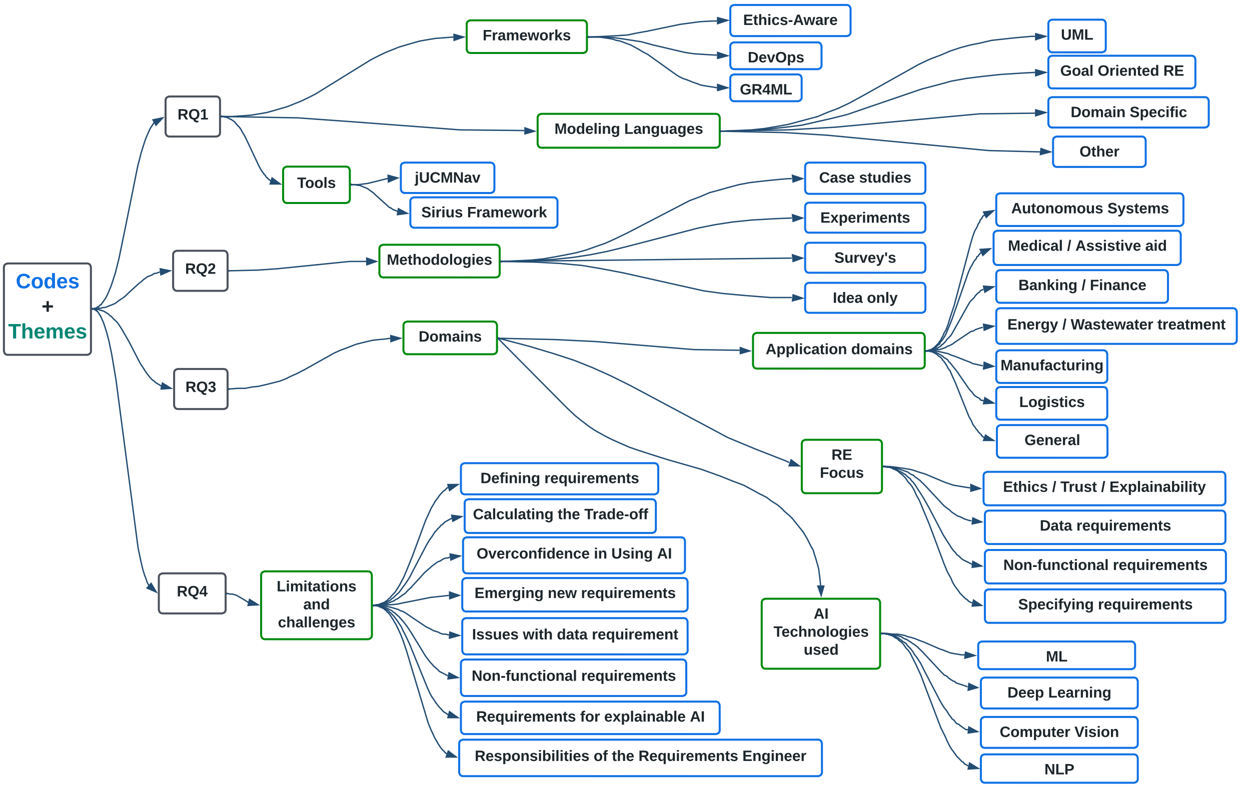

Figure 3: Codes and themes emerging during data extraction based on our research question's

During the initial search, we found that more studies focused on using AI to manage RE and less work on RE4AI. For example, from the first ten results returning from the IEEE Xplore database during the initial search, eight of these papers researched ways to manage RE using AI, and only two focused on RE4AI. Similar patterns were evident in most of the search results obtained from other databases. We also observed that the amount of research in RE4AI has increased lately, as shown in Figure. 4. We found that 90% of the primary studies were published between 2017 and 2021, with 74% of the results published in the last three years. The increase in publications indicates that more researchers are looking into addressing RE-related issues when building AI systems.

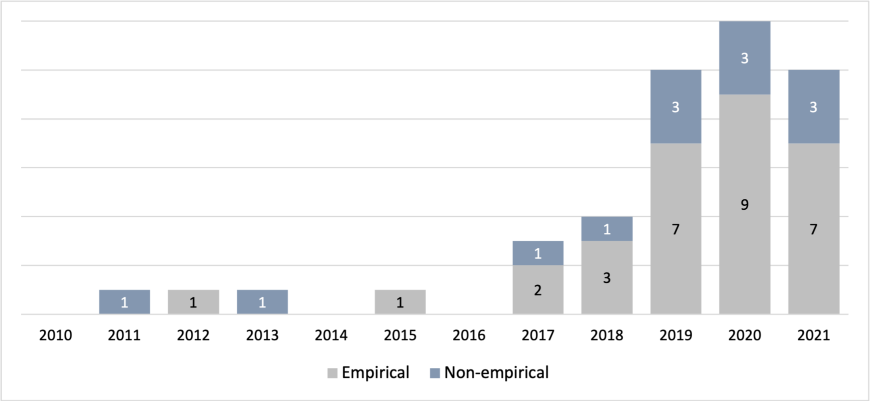

Figure 4: Increase shown in the number of publications per year

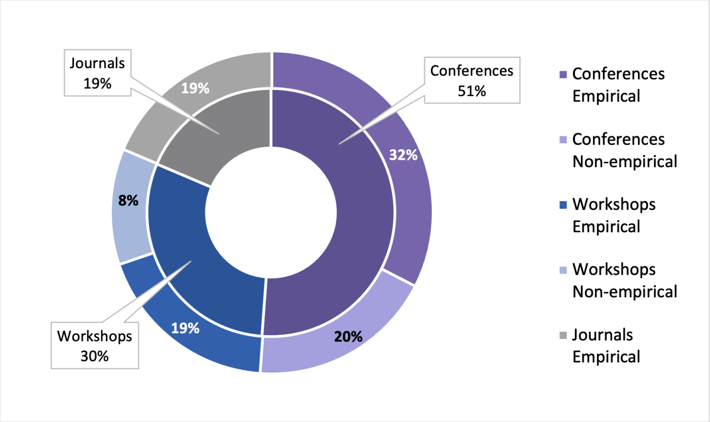

Figure 5: Distribution of selected papers at current publication venues

Figure. 5 illustrates the publishing venues for the final papers. Conferences were the most popular place for publishing, with more than half of the published selected primary studies being from conferences. Workshops were the next popular venue for publications. Journals were the least popular platform for publications on RE4AI, possibly because the work is relatively new and needs further development to improve its maturity. Although only eight of the primary papers were published in journals, they were all empirical studies. Conferences and Workshops had a higher number of non-empirical studies, and more "idea only" papers (with no empirical evaluation) were presented at such venues.

4.1. RQ1 Results

In this section, we provide results from our study for RQ1: *Which Requirements Engineering Frameworks, Notations, Modeling Languages, and Tools have been Proposed to Build AI Systems?*.

4.1.1. Requirements Engineering Frameworks

For this research question, we wanted to identify currently used or proposed frameworks to manage and combine RE methods for AI systems. We identified studies with frameworks if they provided more than one level or layer that build up in providing a holistic solution to manage requirements for AI systems. We only selected frameworks that focused entirely on RE. For example, in DoReMi [94], only one layer of the framework focuses on RE, therefore it was not included as an RE4AI framework. Three different frameworks were identified to include: "Holistic DevOps" [61] that combined multiple approaches to building AI software. "Ethics-Aware SE" [64] for analyzing ethical requirements. And finally, RE4ML [65] that aided in requirements elicitation for ML systems.

Holistic DevOps Framework: The authors in [61] first identified three different approaches to building software, and a "Holistic DevOps Framework" was created to combine all three practices. The framework sets rules as to when each approach should be used. The authors mention that some companies currently use this framework in the industry. However, it is not addressed in research. These practices included:

- *Requirements driven development* can be applied when features are clearly understood, well documented, and used as the basis for building a software system to deliver stakeholders needs. This approach is suitable for projects that do not require frequent changes to the system [61]. Tuncali et al. [79] utilized this approach to test requirements in a virtual environment for a self-driving car. They found that this method reduced the need for resources to design and test the system.

- *Data/outcome driven development* experiments with different methods and solutions to achieve the desired outcome. This approach works with projects that require frequent updates and new features. The method monitors large amounts of data to find specific patterns and is commonly used in online businesses [61]. Design decisions and systems characteristics of data-driven approaches are usually defied by the analysis of recorded data [113].

- *AI-driven development* is used when large amounts of data exist but limited resources to manage the data. Automotive companies with image recognition or user interfaces that use speech recognition tend to use this method. It is also ideal for predicting future activities from existing patterns found in data [61].

Bosch et al. [61] explained that businesses are moving towards data-driven approaches, as decisions are becoming more dependent on data to determine the system's functionalities. This change resulted in the demand to modify current RE practices to become more adaptive to data-driven approaches. Also, data-driven RE is changing the way requirements are elicited and obtained as the traditional techniques of interviews and questionnaires are changing to include information obtained from social media and online forms [42].

Ethics-Aware SE: Aydemir and Dalpiaz [64] proposed a method that would allow requirements engineers and stakeholders in analyzing ethical requirements. Ethics-aware SE consisted of five phases to include:

- Articulation: This phase involved eliciting and modeling ethical requirements from stakeholder while ensuring that both parties were inline with the ethical values for the proposed system. Ethical values could include diversity, ensuring user privacy, transparency, work ethics, etc.

- Specification: Involves matching the ethical requirements identified in the first step to the systems functional and quality requirements.

- Implementation: Applying the ethical requirement and building the software product.

- Verification: Continuously checking if the product is inline with the ethical requirements.

- Validation: Testing if final software product is aligned with the ethical requirements proposed by the stakeholders in the first phase.

Each phase involved a method to extract, manage or evaluate ethical requirements. The authors argued that ethical requirements need to adapt to the changes in today's software systems. So providing a platform that allows the stakeholder, developer, and requirements engineer to collaborate during the different RE phases could help mitigate ethical issues.

GR4ML Framework: Nalchigar et al. [65] presented a conceptual framework that offered three modeling views. The three views include:

- Business View: The first view facilitated addressing and setting requirements related to the business domain and user needs. In this layer the business related goals are elicited and modeled. The goals include 'decisions' to be made, 'questions' to help make decisions, and 'insights' to provide knowledge to the question goals.

- Analytics Design View: The second view involved the selection of technical features such as machine learning algorithms and what qualities and trade-offs should be considered when choosing an algorithm. Analytic had three main goals to include: 'prediction' goal that intends to predict a value based on existing attributes, 'description' goals for describing the selected dataset, and 'prescription' goal to find the best alternative among given options. The analytic view also included softgoals to capture qualities and an indicator to measures the performance of the softgoal.

- Data Preparation View: The third view assisted in the selection and understanding of available data sets. This view helped in preparing data to use and share among the team [114].

All three views were combined and linked together to provide a holistic view of the entire system and its connections.

4.1.2. Existing Modeling Languages, Requirements Notations found in RE4AI

In total, 21 studies used modeling languages or requirements notations to present requirements. The most popular modeling notations and languages among the studies were UML, Domain Specific Modeling (DSM), followed by GORE, as shown in Figure. 6. The study in [74] combined two modeling techniques to produce their model, while [75] created an extension to an existing one. We found that with the new search (mid-2020 to mid-2021), there was an increase in studies that used DSM. Five of the added 16 new studies used DSM to display requirements for specific

Table 7
Modelling languages and requirements notations used in selected studies

Modelling language	Study
GORE (Goal-Oriented RE)	[74, 82, 85, 86, 104]
UML / SysML / Use Cases	[74, 75, 78, 102, 106, 89, 96, 94]
Signal Temporal Logic (STL)	[79]
Traffic Sequence Charts (TSC)	[80]
Casual Diagrams	[110]
Domain Specific Models	[83, 90, 107, 87, 95, 97]

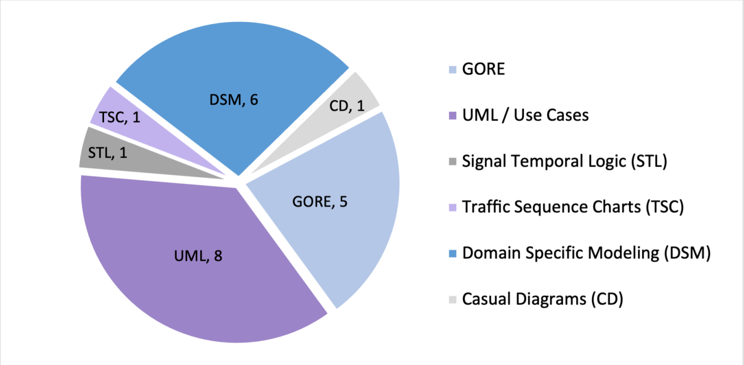

Figure 6: The distribution of modeling languages and requirements notations used in selected studies

Table 8
The different GORE methods presented in selected studies

Gore type	Description	Study
FLAGS (Fuzzy Live Adaptive Goals for Self-adaptive systems) [115]	Used to presents requirements for tasks performed by a smart surgical robot	[74]
CORE (Capability Oriented Requirements Engineering)	Uses goals to capture the systems current and desired capabilities	[82]
GRL (Goal-oriented Requirements Language)	Modeled requirements for an autonomous aircraft system to detect radiation levels in disasters	[85]
i*	Created a model for people living with dementia using i* soft goals	[86]
GORE-MLOps	Proposes a methodology to models uncertainty in requirements for AI systems	[104]

application domains. Three studies focused on using UML, and no additional studies were found to use GORE. This shift shows that more studies are growing towards using DSM to present requirements for AI-software applications.

GORE: Five studies preferred the use of goal-modeling techniques. The authors in [74] argued that using goals to present requirements for surgical tasks was more favorable as they seemed to be in line with how surgeons think or perceive tasks from a medical perspective. Dimitrakopoulos et al. [82] stated that goal-oriented methods were more suited for capturing business requirements. Neace et al. [85] chose to model with GRL because it provided better support for modeling non-functional and quality requirements. They also stated that GORE has become more popular in modeling requirements for autonomous systems. Ishikawa et al. [104] proposed to use GORE-MLOps, a methodology that adapts requirements analysis from GORE methods to ML systems. Finally, [86] favored the use of soft goals in i* models as it provided a better description of a person's objective, such as quality of life. Table 8 shows the different studies that used GORE as a modeling language.

UML: Eight papers used UML to model requirements, as shown in Table 9. Three of these studies used Use Cases [78, 102, 94]. In [75] SysML is used to model functional and non-functional requirements. SysML allowed them to graphically present requirements and their relationships, therefore providing a visual view of the behaviors

Table 9
The different UML methods presented in selected studies

UML type	Description	Study
Statechart and sequence diagrams	Uses statecharts and sequence diagrams to model the medical robots procedure, and the interaction between the system and the user	[74]
SysML	An extension of SysML is used to model functional and non-functional requirements for automotive car systems	[75]
Use Cases	First study created use cases for traffic scenarios and proposed navigation solutions for vision impaired people. Second study generated a use case for a young girl suffering from Attention Deficit/- Hyperactivity Disorder (ADHD)	[78, 94]
Actionable Use Case	Proposes an actionable use case diagram as a means of collaboration between the data scientist and the software engineer	[102]
OntoUML	Proposes an ontology using UML to model trustworthy requirement	[106]
Semi-formal UML	Proposed a model called "Dataset Requirements Concept Model" (DRCM) to display requirements needed for the structure of datasets	[89]
Activity Diagram	The study gathers requirements and models them using use cases and activity diagrams to model a configuration system for an Unmanned Aerial Vehicles	[96]

between each requirement. However, the drawback to using SysML was that it did not provide enough aid to model non-functional requirements. As a result, the study proposed and tested an extension to SysML to support non-functional requirements. Amaral et al. [106] proposed an ontology of trust to help define requirements for trustworthy AI. The study implements these trustworthiness requirements in an OntoUML model. The model also assisted in displaying any risks related issues when it came to trust. In [96] Activity Diagrams were used to model requirements for a configuration system. And finally, [89] used a semi-formal UML model to generate their dataset requirements concept model (DRCM). As reported by the authors, the rationale behind using UML was the ease of using UML for non-software engineers.

Signal Temporal Logic: Signal Temporal Logic (STL) is a specification language that enables real-time reasoning of properties by providing past, and future variables [116]. The study in [79] used STL to specify requirements for a perception system in an autonomous vehicle. It provided features such as reachability, safety, and reactive requirements to include in the specifications. They then mapped these requirements into three testing scenarios using the Sim-ATAV framework and a virtual environment to generate test cases for autonomous vehicles.

Traffic Sequence Charts: Traffic Sequence Charts (TSC) is a graphical specification language used for traffic scenarios. TSC is based on snapshots, and each snapshot represents a traffic situation. When assembled, snapshot charts consist of history, future, and consequence. Snapshots are also linked or combined with operations such as sequences and choice. The work in [80] used TSC to display requirements for an autonomous vehicle. The main objective of the study was to find any inconsistencies in TSC requirements specifications.

Causal Diagrams: Causal diagrams are mathematical graphical notations that represent statistical relationships between objects [117] and are used to identify specific variables or measures. They are often used in epidemiology to minimize biases [118, 119]. In [110] the author proposes to use casual diagrams to generate different explanation paths in order to provide a user-understandable explanation for self-explainable systems. The study uses the framework proposed in [120] “Monitor, Analyse, Build, Explain" (MAB-EX) to provide explanations for the systems crossing controller that monitors traffic intersections.

Domain Specific Modeling: Domain Specific Modeling (DSM) is defined by Fowler as a “computer programming language of limited expressiveness focused on a particular domain” [121]. Six studies used DSM to model requirements [83, 90, 107, 87, 95, 97]. The study in [83] builds a model for a Smart Process Control System based on requirements gathered from the industry and survey results. In [90] the authors used the Knowledge Management on a Systematic process for Requirements Engineering (KMoS-REload) to elicit and model external and internal knowledge using the KMoS-REload process. In [107] requirements were modeled to show the interaction between humans and a drone to detect hot spots and victims through a fire in a building. In [95], a model was built to present the framework for a prototype application that was based on requirements elicited from interviews for a mobile health application. And finally, Hall et al. [97] built a model for explainable AI that was then applied to an industrial case study.

4.1.3. *Modeling Tools*

We found the use of two different tools. The jUCMNav tool was used in two studies and is a free graphical editor that supports modeling requirements in Goal-oriented Requirement Language (GRL) and Use Case Maps (UCM) [25]. Neace et al. [85] implemented the requirements proposed in [122] using the goal-oriented requirements language GRL. These requirements were then modeled using the jUCMNav tool to present them graphically. In [104] the study used jUCMNav to present their goal model that represented uncertainty in AI systems. The second tool was created by Ries et al. [89] who developed a toolset based on the Sirius framework [26] to present their modeling language visually. The Sirius framework is an open-source graphical editor that allows for domain-specific models to be presented visually [123].

4.2. RQ2 Results

In this section, we discuss results pertaining RQ2: *Which evaluation methods used to assess empirical studies in RE4AI?* We identified three types of empirical investigations used in evaluating existing RE4AI techniques, models, and frameworks. These methods included case studies, experiments, and surveys. Figure. 7 shows that the most common type of methods used were case studies followed by experiments. We also obtained the different data collection methods used in each empirical evaluation as shown in Figure. 8. We found interviews to be the most popular form of data collection, followed by using existing or creating new databases. Other data collection forms included workshops, meetings, focus groups, gathering data from online websites and user feedback, etc.

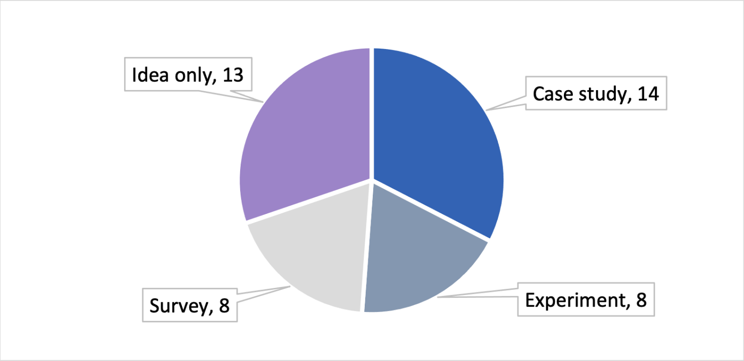

Figure 7: The distribution of the methods found in the selected primary studies

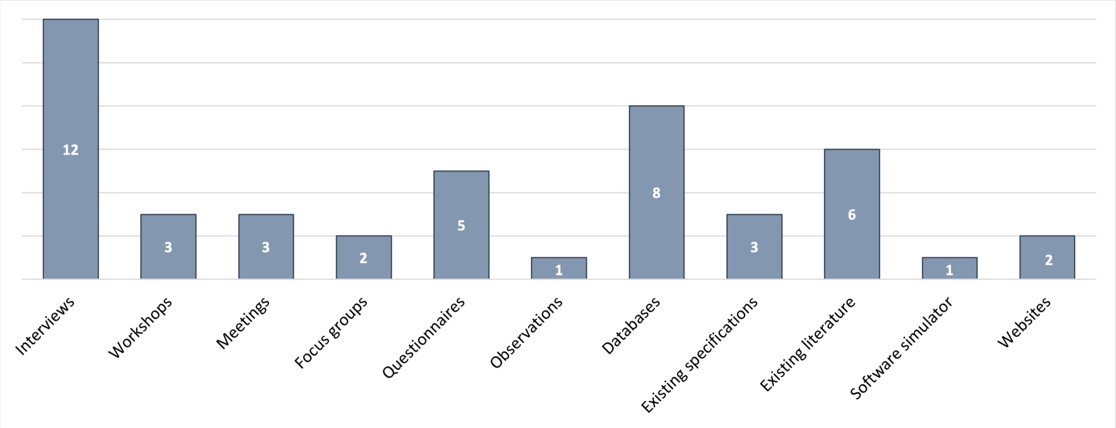

Figure 8: The different data collection methods used in selected empirical evaluations and how often they were used

Table 10
Papers with Case Study Research

Study	Case Specification	Outcomes / Aims	Data Collection
[75]	An Active Lane Change Assistant, to support drivers when changing lanes in autonomous cars	Better connectivity and requirements tracing and was easier to track and implement changes	Existing specification sheets
[76]	Feasibility of organizations in using AI solutions and for a banking systems fraud detection software	The case did not have enough training data on fraud cases to use AI as a solution	Interviews
[77]	Finding requirements for "pedestrians" for a self-driving	A model that displays all requirements needed to describe a pedestrian	Web-search engines + "Caltech" dataset
[61]	Identify different approaches when building AI software	Proposed a "Holistic DevOps" Framework	Meetings, interviews, workshops
[84]	Addresses three NFR's for a robots vision system for a RoboCup competition	Results showed better performance in the new vision system	A dataset with 5 prerecorded image streams
[85]	Implement requirements for an autonomous aircraft system to detect radiation levels in disasters	Aimed to optimize high-performance, maximize safe operations and low cost	Requirements from [124] + Published data
[87]	Evaluate the effectiveness of the quality characteristics for ML systems	Found that 22 of the issues were present in the requirements definition	Surveys, prior research, collaborations
[88]	Feeding faulty data into a DL system for a wastewater plant and monitors the behavior	Identifies characteristics needed to improve the quality of data requirements	Data collected from an overflow sewer site
[89]	Identifying images of digits between 0 to 9 using a DL model	Improved quality of datasets and identify issues with digits early	Images of five-segments "characterized digits"
[90]	Update a traffic monitoring and routes management systems for a freight company using AI	Establish functional requirements and improved teams communications	Focus groups
[93]	Eliciting Requirements for a navigation system for vision-impaired people and the prototype is tested	Collected requirements could have potential benefits to developing user-centric SAS	Literature, Interviews, questionnaires, focus groups
[65]	Evaluates the "GR4ML" framework on a start-up company in healthcare to elicit requirements	New ML requirements are established, and improved team communication	Interviews, meetings
[94]	Gathering requirements from pediatricians for a young girl suffering from ADHD	Used to evaluate the method called "DoReMi" for AI-generated explanations	Questionnaire and interviews
[95]	Prototype for a mobile application to identify if the user had any signs of depression and anxiety	Initial prototype had potential to aid with people's mental wellbeing during the Covid-19 lockdown	Interview 37 people and 20 participated in a survey

4.2.1. Case Studies

We identified 14 studies that performed a case study. Table 10 display's the list of all case studies found and the outcome of each one. The authors in [75, 85, 90] conducted case studies to evaluate a model that was created based on a set of given requirements. Nakamichi et al [87] propose a quality evaluation model and a method to measure functional correctness for ML systems, and three case studies were carried out to evaluate the proposed model. In [76] the results justified what information was needed and what was missing to implement AI as a software solution for organizations. In their case, they found that they did not have enough training data to implement AI as a solution. The authors in [61] conducted several case studies with the industry to identify three different approaches to building AI-based software systems and proposed a "Holistic DevOps Framework".

Fenn et al. [84] evaluate a new architecture design for a vision system using a case study. The new architecture focused on three non-functional requirements: portability, extensibility, and modifiability, and the authors found the system performed better with the new design. In [95] requirements are elicited for a mobile health application and the feasibility of these requirements are evaluated. Rahimi et al. [77] worked on finding specifications for machine learning components on safety-critical domains. The study proposed a solution to extract and visualize domain specifications for requirements. They focused on extracting requirements for "pedestrians" and how a self-driving car would recognize pedestrians.

Challa et al. [88] proposed characteristics needed for data requirements to provide quality data for deep learning systems. In [93] the study elicits requirements for a smart assistive system and creates a prototype to validate and

Table 11
Papers with Experiments Research

Study	Experiment	Outcomes / Aims	Data Collection
[74]	Requirements for a surgical robot to find the force required when inserting and removing the needle into cancerous cells	Data collected is to be used in a future case study to train a robot to perform a procedure on removing cancerous cells from a kidney	Interviews + manually collecting force to insert and remove a needle
[32]	Adding noise to a vision detection system (YOLO) to identify images for a self-driving car	Using YOLO is not robust against changes and could have severe consequences in safety-critical systems	Image from Berkeley DeepDrive [125] and Gaussian noise
[79]	Requirements are used to build a virtual environment to test functionalities of a self-driving car	Capturing unacceptable behaviors and identify the test cases that violate the requirements	Data gathered from 3 sensors: CCD camera, lidar, and radar
[80]	Prototype tool to maintain requirements consistency for autonomous driving and analyze a traffic situation	They could not identify all conflicts and requires further research to create refined consistency notations	Results taken from previous studies
[81]	Implementing different algorithms on a dataset to find the effects of sampling rate and house numbers	Algorithms such as classification and regression performed badly when the sampling rate was low	Data on energy consumption for 3 appliances and 58 houses
[82]	Capabilities are presented in a goal model and tested in a traffic simulation to demonstrate its efficiency	The method could be effective in the assessment of real-time traffic situations	Software simulator that generated traffic scenarios
[91]	Satisfaction of NFR - the threshold is calculated based on the trade-off on the impact of a NFR on the rest	The outcomes showed improved results when compared to other methods used to prioritize requirements	Existing requirements specifications
[96]	Requirement driven approach is used to create a configuration system for a product line	The study gathers requirements and models them using use cases and activity diagrams	Literature and Existing requirements

test the effectiveness of these requirements. Ries et al. [89] used a case study to improve data requirements for Deep Learning (DL) systems. The study identified requirements needed for the database's structure. And finally, in [94] a method called "DoReMi" is built for AI-generated explanations. A case study is conducted on a clinical trial for a young girl suffering from Attention-Deficit/- Hyperactivity Disorder (ADHD) to evaluate explanations generated by the AI system.

4.2.2. *Experiments*

Eight studies used experiments to evaluate their proposed solutions as shown in Table 11. Three studies generated experiments to evaluate a model or method presented [80, 82, 96]. Another two studies experimented on finding data requirements. In [81] the study focused on finding data requirements for monitoring energy consumption for houses in Japan, and [74] gathered data to use for training a surgical robot. In the last three experiments, [32] evaluates the robustness of requirements for a computer vision system, [79] tests requirements for a self-driving car, and [91] experiments on the threshold needed to find the trade-off on the impact of a NFR on the rest.

4.2.3. *Surveys / Interviews*

Survey methods were used as means to gather data for both case studies and experiments. As shown in Tables 10 and 11, several studies used surveys to collect data either to build an artefact or to evaluate it. However, we found seven studies that explicitly used surveys as an evaluation technique. These techniques included questionnaires, interviews, workshops, and observations. Table 12 shows the studies that only performed surveys along with the outcomes. In [78] the study uses interviews to gather requirements for an orientation and navigation system that is based on computer vision for people who are vision impaired. Lockerbie et al. [86] conducts workshops to identify soft goals for an AI application that would provide better quality of life for people with dementia. In [99] the study carried out interviews to identify requirements for a system that flags users with the potential of filing a lawsuit against a given power company. The system also provided information on these customers to the company's lawyers and approaches such customers to discuss available solutions through a chatbot function. Both studies in [66, 98] conduct interviews to shed light on the challenges RE faces when building ML systems. And in [83] the authors evaluate a model for a Smart Process Control System through several questionnaires and workshops. Finally, both [97, 92] use interviews to find requirements for explainable AI.

Table 12
Studies using interviews / surveys

Study	Survey Specification	Outcomes / Aims	Method
[78]	Extract requirements, and use cases for six traffic scenarios for a computer vision system	Proposes the use of ADAS algorithms and provide solutions to how they can detect objects	Interviews
[66]	Identify data scientists role in writing, eliciting, documenting and analyze requirements when building ML systems	The author addresses these challenges by outlining a process for RE4AI systems based on the traditional RE activities	Interviews with 4 data scientists
[83]	Identifies functional requirements and performs surveys to evaluate the implementation of these requirements	Identifies an improved set of requirements and a set of four features are specified to build a conceptual model	Questionnaires, workshops and observations
[86]	An initial model for quality of life is evaluated. Goals were merged and new ones identified	Proposal for an explainable AI app to provide better quality of life for people with dementia	Workshops with 12 experienced care workers
[92]	"Scenario-based Elicitation" method is used to identify requirements for explainable AI in fraud detection	Two scenarios of fraud detection are extracted to be used in future experimental prototypes	Interviews 3 fraud experts working at a bank
[97]	Interviews with 12 experienced aerospace engineers	Identified explainability characteristics to help map requirements to explainable techniques	Formal interviews and informal discussions
[98]	Interviews people from industry to find challenges related to NFR for ML	Identifies the important and less important NFR in ML systems	Interviews ten ML specialists
[99]	Identify requirements for building a system that identifies power users that have potential to file a lawsuit	Provide information to the company's lawyers and use a chatbot to approach these customers and discuss available solutions	Interviews, complaint websites, Power companies laws and rules

4.3. RQ3 Results

In this section, we elaborate on results for RQ3: *Which target application domains and AI areas of focus considered in the existing approaches?*. The application domain varied between the studies, as shown in Figure. 9. 15 papers listed under general did not specify a domain when applying their concepts. For example, [100, 98], focused on issues and challenges when applying non-functional requirements to AI and ML. Nakamichi et al. [87] tried to find ways to improve RE techniques for AI systems. They conducted a study to evaluate quality requirements and deliver customer's needs. Bosch et al. [61] identified the different approaches to building AI systems in general and proposed a framework to combine them. And Ishikawa et al. [104] presented a model to show uncertainty in AI systems and its impacts on current RE techniques. Our results showed that the domain with the highest interest in RE4AI were autonomous driving and robotics, and most studies in the field of autonomous driving and robotics were based on empirical investigations. The following most popular application domains in RE4AI research were medical and assistive technology, and similar to autonomous systems, most of these studies were empirical investigation.

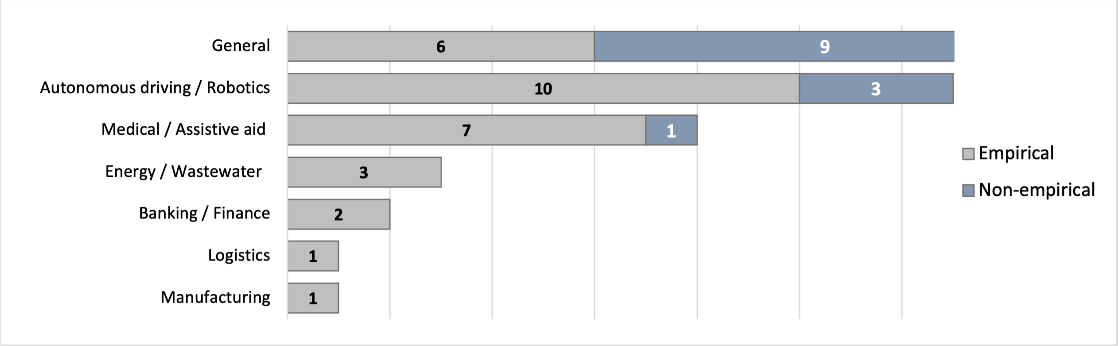

Figure 9: Number of studies found in each application domain in RE for AI systems

Figure. 10 shows the different requirements focus domains. Investigating data requirements seemed to gain the most attention. We found that all empirical studies that involved investigating data requirements were conducted during the past three years (2019, 2020, and 2021), showing that this is an emerging topic and needs further exploration. Several studies have also emphasized the importance of managing data requirements in building AI systems as “data replaces code" [66]. The next most popular research areas were explainability and Non-Functional Requirement (NFR). Some studies reported explainability as a NFR. However, in our classification, we noted studies that only focused on explainability as a separate entity. Studies on explainability were only found after the year 2019. Also, domains that did not have an empirical investigations (non-empirical) were ethics and trustworthy AI as they appeared to be theoretical papers and proposed methodologies that were not yet evaluated.

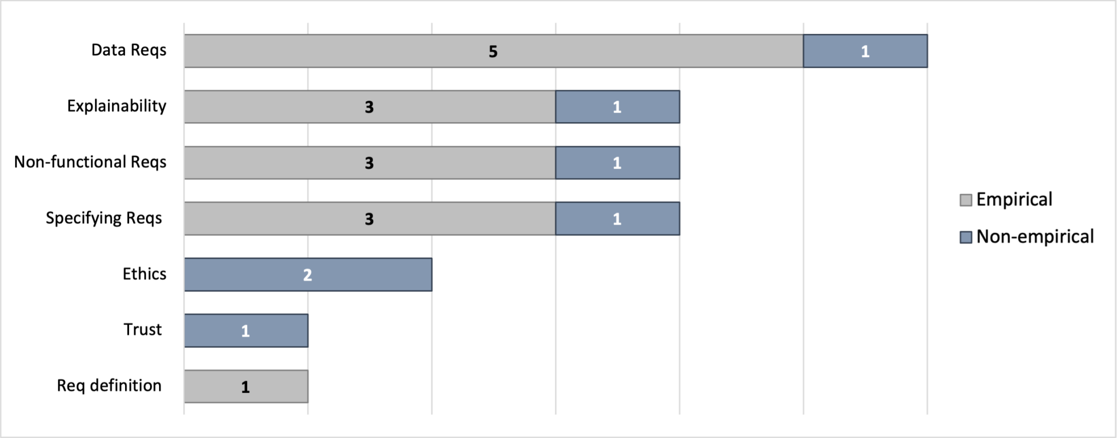

Figure 10: The different requirements topics addressed in the selected studies

When looking at the different AI technique used in RE4AI research, we found that most of the studies worked with ML and computer vision as shown in Figure. 11. Only one study proposes to use Natural Language Processing (NLP) and presents requirements for a chatbot system [109]. Moreover, we did identify a high number of studies that used NLP from our initial search results. However, these studies investigated the use of AI to manage RE.

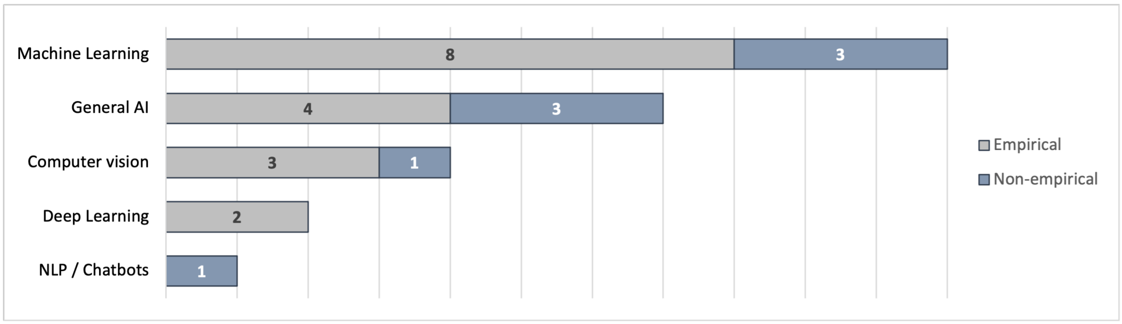

Figure 11: The AI topics used in the selected studies

In Figure. 12 we mapped all the application domains against the methodologies used to evaluate empirical work. We found that most of the non-empirical studies were not linked to a specific application domain, thus listed under the general domain. Some of the studies that focused on autonomous systems used case studies to evaluate their methodologies. However, a more significant number performed experiments. In contrast, medical and assistive technology studies preferred case studies.

4.4. RQ4 Results

This section discussed our study’s results for RQ4: *What are the limitations and challenges of existing requirements engineering techniques when applied to AI systems?*. We identified challenges and issues presented in the selected

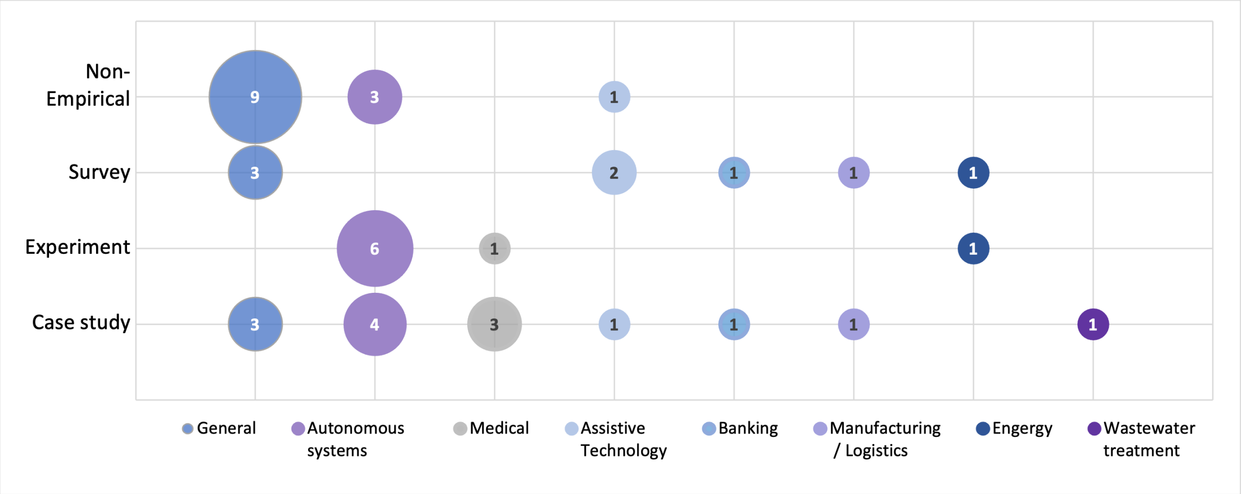

Figure 12: Evaluation methodology distribution for each application domain

primary studies that have emerged due to the shift in RE4AI. Figure. 13 displays the recurrences of each issue in the literature. Issues that appeared more often were linked to data requirements and deciding on trade-offs, followed by the emergence of new requirements. We found that this research question did not change significantly with the additional results obtained from the second search (between mid-2020 and mid-2021). The observation was that most added studies tended to focus on evaluating proposed solutions rather than presenting issues and challenges.

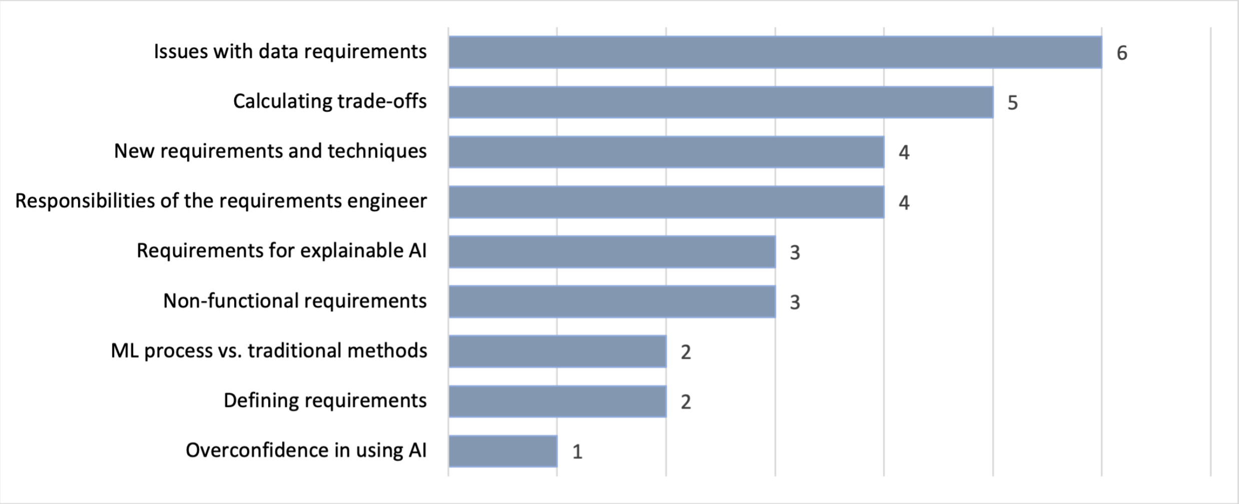

Figure 13: Number of RE4AI issues appearing in selected primary studies

4.4.1. Overconfidence in Using AI

There is sometimes a misconception that “AI will solve everything" for organizations [126]. Sandkuhl [76] explained that the general public usually overestimates the capabilities of AI solutions. During a series of workshops and meetings with several industrial partners, they found that many organizations would choose to use AI without having the experience and expertise in building systems with AI components. Sandkuhl emphasized that capturing requirements early on in the project is essential so stakeholders clearly understand the capabilities and limitations of AI. And while AI is not always a feasible solution, companies should establish the need to use AI before proceeding further in the projects.

4.4.2. Defining Requirements

Defining requirements for AI systems can be somewhat challenging. Some requirements might be vague or hard to define. For example, how do we define a “Pedestrian" to a self-driving car? Every person might have a different definition for a pedestrian. Rahimi et al. [77] focused on finding the requirements for “pedestrians" and how a self-driving vehicle would recognize pedestrians? The process involved searching for any feature that represented a pedestrian. Horkoff [100] explained that our understanding of non-functional requirements is not complete, and we need to set out standards to how we define them; for example, how do we define fairness?

4.4.3. Nature of Machine Learning Systems and the Traditional Approach of RE

Traditional non-AI systems are usually deterministic in nature and have a process for RE techniques that are well researched. However, this is not the case in AI-based software, as outcomes can be unexpected. In [87] the study emphasized the need to implement new quality techniques for ML systems. Vogelsang and Borg [66] explained that the existing methods used in RE need to change to accommodate the different activities currently used for AI systems. The author defined new types of methods for RE to be included when creating requirements for ML systems. For example, the elicitation phase should identify requirements, such as data and explainability. Hence, the need to develop new tools or re-evaluate existing ones to support RE4AI.

4.4.4. Calculating the Trade-off

The issue with deciding how to calculate trade-offs came up in 5 different studies. Horkoff [100] explained that one of the challenges with ML systems’ non-functional requirements was to calculate trade-offs when choosing an ML algorithm. For instance, do we trade privacy for transparency or fairness for accuracy? And how would we specify or express these choices? How do we decide on what requirements could be traded and at what cost? Calculating the trade-off between the impact an NFR will have on the rest of the requirements is also imperative, whether positive or negative. In [91] a satisfaction threshold is established for the trade-off of NFRs.

The authors in [94] highlighted the importance of calculating the trade-off between sensitivity and specificity in clinical studies, as clinicians always stress the importance of reducing the number of false negatives when diagnosing patients. Thus, missing a diagnosis can lead to negative consequences. In [84], the authors traded a slight cutback to efficiency for a substantial increase in modifiability to the system’s design. This cutback provided a more straightforward method to maintain data validity. Although there was a slight reduction in efficiency, it did not affect the system performance and was still within the processing power range. Therefore accuracy and reliability remained unchanged.

Shin et al. [81] worked on finding data requirements for monitoring energy consumption in houses in Japan. They focused on 2 data requirements: sampling rate and number of samples used. The study involved collecting data on energy consumption for three different appliances from 58 houses. They found that it was essential to measure trade-offs between performance/cost when it came to the data’s sampling rate. Algorithms such as classification and regression performed poorly when the sampling rate was low. So the higher the sampling rate, the better the quality of data was. The number of houses used was also important when it came to better performance, and including more samples in the dataset provided a more comprehensive range of diversity. However, increasing the samples could be costly. Therefore, to what extent can we invest in cost? There should be a limit to how far we can choose between performance vs. cost.

4.4.5. Responsibilities of the Requirements Engineer

Vogelsang and Borg [66] stated that data scientists are responsible for writing requirements in current ML systems. As AI and ML are integrating into software systems, a new role for data scientists is emerging in the process, forcing software teams to adapt to these changes. These new roles have resulted in a gap between SE practices, AI communities, and data scientists. Nalchigar et al. [65] explains that the gap between stakeholders and data scientists can result in mismatches between business needs and data needs. On the other hand, Challa et al. [88] reports that the RE community is not equipped to handle the vast amounts of data needed in building AI systems. In [102], the authors emphasize the importance of including the data scientist in the process of defining and eliciting requirements, especially requirements related to data extraction. For AI systems that are data-centered, it is crucial to have a data scientist work closely with the requirements engineers to elicit and identify relevant data for the project. Therefore, there should be some type of communication between the requirements engineer and the data scientist, especially during the early phases of RE.

4.4.6. The Emergence of new Requirements and Techniques

With the emergence of new requirements for AI systems such as data, ethics, trust and transparency, new challenges are born for RE. Bosch et al. [61] emphasized the need to adapt and complement old practices and techniques with new ones rather than replace the old practices entirely. Some studies noted that research on RE4AI is not applied to practice, and findings are not being used or addressed by other researchers. For example, Shin et al. [81] identified some data requirements that should be used in AI systems for energy consumption. However, they found that similar projects were not practicing the use of such requirements.

The same goes for ethical requirements. Aydemir and Dalpiaz [64] argued that ethics is usually overlooked, and with the change in today's software systems and the introduction of AI, ethical requirements need to adapt to these changes. The authors argue that ethics is widely discussed for AI systems but neglected during the building process. Kuwajima et al. [105] noted that most software standards such as ISO/IEC 25000 series did not apply to ML systems and has no support for ethical requirements.

4.4.7. Issues with data requirements

One of the significant issues with data requirements is the expense that comes with data-generation [81]. Then there is the availability, quality [83], training and testing of data [87]. Requirements need to make sure the quality of data is appropriate, whether the data is available, how to test it, and which data to select for training. Altarturi et al. [102] explain that RE methods focus on requirements that are user-centric and do not give enough attention to data requirements. In [89] the author argues that Deep Learning (DL) models usually rely heavily on datasets and are not addressed in RE. The study tries to identify requirements needed for the structure of the datasets and involves finding datatypes, attributes, and properties that need to be elicited during RE.

The emergence of data requirements has posed new issues for RE. Sandkuhl [76] found that data needed for an AI project was easily accessible for many companies. However, the available data lacked the structure and rules that were necessary to implement and train an AI system. The study then listed numerous AI requirements to consider. These requirements included data quality, structure, and format. For AI-based software that uses data as a primary driver to build the system, it is essential to set rules and carefully select requirements for data selection and management.

4.4.8. Non-functional Requirements

Our understanding of non-functional requirements (NFR) has changed with the addition of AI components when building software systems, and the traditional approaches for managing NFR's require new methods and solutions. These methods need to evolve and re-evaluate how NFR fit into RE4AI. For example, some NFR's, such as compatibility and modularity, are not as important in ML systems as they were in traditional software systems. In contrast, other overlooked requirements such as fairness and transparency hold more value [100, 98]. There also appears to be less research on modeling non-functional requirements, and research tends to focus mainly on functional requirements [75].

4.4.9. Requirements for explainable AI

Providing requirements for explainable AI has created new issues in RE4AI. For example, in the case of complex systems such as autonomous cars, it can get challenging to specify which parts of the system need to be explained or how to explain them [110]. The author in [97] points out that explainability requirements could collide with risk factors of an AI systems such as in safety-critical systems. Also, explainability requirements can conflict with others, such as security, cost, and precision[110]. Having an AI system that is more explainable might be more expensive to build. In such a case, when would it be worth the expense to have a more explainable system? Kohl et al. [63] proposed to use the Softgoal Independency Graph (SIG) to model explainability along with the other NFR to minimize conflicts.

5. Threats to Validity

In all phases of our SLR, we considered and attempted to mitigate potential threats to validity, common in SLRs for software engineering [127].

Internal Validity: To reduce any selection bias while conducting our search, we had an initial protocol to identify a comprehensive set of keywords that were considered relevant in RE4AI literature and tested them out first. We chose six databases for our search to broaden our results and ensure most studies were included and only selected peer-reviewed papers. Pilot tests were performed and checked by second and third authors to validate the results. Snowballing was also performed to capture papers that might have been missed in the initial search results.

From the initial selection criteria, we manually read all the titles and abstracts to filter them further. The first author did the first round of selection and then verified the results in consultation with the second, third and fourth authors to reach the consensus for the final results. Once the final list was established, we performed a more detailed scan of the entire document for the resulting papers. The second filtration process involved a more comprehensive quality assessment test to only include primary studies that passed a specific grade or were in scope. These studies were selected based on the criteria we set, and for some papers, we were not sure if they fit our criteria. In such situations, we discussed the paper's selection in several meetings among the authors to ensure the selected article's focus was on RE4AI and to reduce researchers' bias. A similar process was followed for data extraction and analysis to reach a consensus on the results that were used to answer the research questions. Another concern for internal validity is publication bias. To mitigate the publication bias concerns, we checked each primary study as part of the quality assessment criteria to see if there are any reports on any issues and what reliability measures they have performed. We did observe that out of the 30 empirical studies, 9 did not report negative results or address validity issues. However, all the selected studies passed the overall quality criteria.

Construct Validity: The major concern for construct validity is based on the appropriateness of the papers selected. We note that we selected papers that were relevant for our RQs, or if the application was entirely focused on RE4AI. We further had clearly defined inclusion and exclusion criteria, developed over extensive discussion sessions and relevant to our RQs. The time frame of our SLR was between 2010 and mid-2021, so any study outside our time frame would not have shown in our results.

Conclusion Validity: One of the major threats to conclusion validity in SLRs is the bias in data extraction. In our study, during data extraction, codes and themes were based on our RQs, so the data selected was entirely focused on answering our questions. In order to reduce conclusion validity, we used Nvivo to extract the data by using thematic analysis. This allowed us to group all results based on the pre-defined codes and themes. We also found some emerging codes in the process of data extraction, so they were added as needed. The coding process was done by the first author, so to reduce bias we conducted regular meetings between the first, second and third authors to discuss the data extraction and analysis process to agree on which data should be used and how it would be displayed.

6. Discussion and Research Roadmap

We found from our analysis of the literature that using existing RE techniques for AI-based software development could be challenging. This is primarily due to the different nature of traditional and AI-based software development processes, which has led to new gaps in RE processes for engineering AI software. In this section, we present the gaps that we identified in the selected RE4AI studies, and the recommendations based on these gaps as suggestions for practitioners and future research. Table 13 presents an overview of these recommendations and maps them to the issues presented in the literature. Below we elaborate on seven recommendations, presented in the table.

6.1. Identify the Need and Feasibility of AI

The first key issue identified in our analysis of the literature was that many organizations are adopting AI just because "everyone else was" and without understanding what the AI component could provide or how much it could solve. Therefore, before deciding to use AI as a software solution, its need should be established first. These needs include the type of AI solution, data availability and having the resources and expertise to build and manage such systems [128]. Is AI required to provide predictions, personalize, or make recommendations? Or is there a need for speech and language understanding, image recognition, or fraud detection? In [94] the author stresses that before starting an AI project, we need to establish who the users are, what tasks they will perform, and what benefits they would gain from using the system. Other aspects of the AI system should be discussed early on such as understanding what the systems capabilities and limitations are.

We propose that *practitioners* maintain a checkpoint to note all required elements needed to create an AI software system. The checkpoint should include the problem it is going to solve, why is it required, and how will it be used. And finding out if the organization has all the resources needed to build the AI product.

Table 13
Mapping of the recommendations to the issues presented in literature

Recommendation	Issue presented in the mapping study	Suggestions for further research
Identifying the need for AI	Overconfidence in Using AI	Maintain a check point to list all required elements needed to create an AI-based software system.
Specifying requirements for AI systems	Defining requirements - Non-functional requirements - Emergence of new requirements - Issues with data requirements	Construct a reference map that would capture the key components and attributes needed when specifying AI system requirements
Using existing RE tools to build AI software	Nature of Machine Learning systems vs. traditional approach of RE - The emergence of new Requirements	Create a taxonomy that gathers all the new techniques and methods for creating AI software
How do we decide on what modeling language to use?	Modelling non-functional requirements	Extend a modeling language to support RE4AI
How do we bridge the gap between requirement engineers, data scientists, and ML specialist	Existing gap between data scientists and software engineers - Requirements Engineer Responsibilities	Create a platform to share and visually present requirements
How do we address issues such as trade-offs or provide techniques to new RE methods	Calculating the trade-off	Trade-offs should be displayed along with requirements when modeling
Empirical evaluations on Ethics, Explainability and Trust	The Emergence of new Requirements and Techniques - Requirements for explainable AI	Conduct more empirical evaluations in future research
Limited studies on RE for Human-centered AI		Use industrial guidelines to create a framework to include RE for human-centered AI.

6.2. Requirements Specifications for AI Systems

In [55] the authors found that requirements can be hard to specify for AI-based software due to the issues related to measuring and defining requirements for non-deterministic systems. Also, the emergence of new requirements such as data, ethics and explainability has posed issues to requirements specifications. How do we specify ethical requirements or explain decisions made by a self-driving car? For example, a vehicle might suddenly brake in front of a bus instead of changing lanes. This decision would be because it weighs between injuring four people crossing the road versus the one person driving the car getting hurt. Would the driver make the same decision or disagree with the ethical choices the car company has made on their behalf [39]? What requirements do we need to provide in such cases?

Specifying requirements for data has also proven to be a challenge. First we need to identify the type of AI-based software solution used for the data collected. Data requirements differ depending on the AI component used. For example, in supervised machine learning, it is important to have data with learning features and proper labels, whereas an unsupervised model trains on unlabeled data. Specifying key characteristics of data should be set early on to avoid discrimination and biases. Discrimination can happen when data collection does not include minority groups. Human-labelled data can also produce biases [66] and using existing data can make it difficult to explain why a given prediction is provided [63]. In other situations, such as in safety-critical situations, data needs to be carefully collected and selected. For example, for a medical-surgical robotic application, a requirement would be to find the accurate exertion force for the needle to enter soft tissue. Data needs to be collected by carefully setting up several experiments [74]. We also found limited studies in our results that focused on identifying and specifying requirements for AI systems.

> We recommend that *researchers* should construct a reference map to document requirements for AI systems. The reference map should capture key components and attributes needed when specifying AI system requirements. We propose using research from the industry and literature to plan and list all possible requirements for AI systems that can be used as a guide to map any emerging requirements into the reference map. The map may be broken into separate sub-maps to ensure all elements are captured, or can be used for specific AI-based systems, e.g., agent-based systems or NLP solutions.

6.3. Using Existing RE Methods to build AI Software

6.3.1. Existing RE Tools used in RE4AI

Engineering AI-based software has had an impact on the way existing tools and techniques are used in RE. Some of these existing tools are not be applicable in RE4AI and the way requirements are elicited and obtained can change when dealing with AI software. In some cases, traditional approaches that require human intervention in gathering data such as interviews and questionnaires are now being replaced by new forms of data collection such as online forms, social media [42], sensors [129], and immersive techniques [130]. Data collected from such sources requires new RE techniques to elicit and manage them. There has been an increase in building tools to manage AI software such as the "AI Playbook" [131] to identify and reduce failures in AI through early AI prototyping, and HINT "Human-AI INtegration Testing" [132] that automates tests for user interactions with AI systems. Despite these advancements, these tools are still in their early stages; they need further testing and need to be validated in RE context [55].

We recommend that *researchers* should identify the available tools used in building AI software, and determine which ones could be used directly or can be customized for RE. We also need to build new tools or extend existing ones to support in eliciting, modeling, specifying, and managing requirements for building AI solutions.

6.3.2. Modeling Languages used in RE4AI

Although around 60% of the selected primary studies demonstrated the use of a modeling language to support requirements, most of the available modeling languages still lacked proper support for RE4AI. Languages such as UML and GORE were more popular, but had their limitations. GORE is difficult for non-software engineers to use, considering that team structure in building AI software involved other roles such as data scientists, project managers and ML engineers. Therefore, it becomes more challenging for non-software engineers to learn and use GORE. In contrast, UML was chosen for its ease of use. However, UML is not as flexible when modeling non-functional requirements and business rules, which are a core part of RE4AI. For example, [64] proposed developing a modeling language that would capture ethical requirements, and [100] suggested one to capture NFRs, specific to ML systems.

Future research needs to be done by *researchers* to extend or augment existing modeling languages to aid in capturing and presenting AI requirements, and cutomization is required for different types of AI systems, such as ML systems, agent-based systems and robotic systems.

6.4. The Communication Gap Between Requirements Engineers, Data Scientists, and AI Stakeholders

As new development team roles and responsibilities emerge when engineering AI software, we found that in order to build proper AI-based software these roles have to communicate information correctly among them. Currently, there is a lack of communication and integration between roles such as, data scientists, AI stakeholders (e.g., ML engineers and conversational NLP developers) and software engineers [41, 66, 65]. For instance, in [66] the authors observed that in current ML systems, data scientists were responsible for writing high-level requirements, resulting in practices that focus on data and model testing rather than understanding the business domain and stakeholders' needs. The authors emphasized that the job of the requirements engineer should be eliciting data requirements, while maintaining data provenance, avoiding biases and validating requirements as data might change over time. At the same time, requirements engineers need to work closely with data scientists as they do not have the expert knowledge to handle and maintain large amounts of data [88]. Based on our investigation, data scientists, AI stakeholders and requirements engineers need to improve their knowledge and understanding of the issues arising from blending AI into software systems, and there should be some form of continuous communication among them to set and manage requirements [11, 55].

We propose that *practitioners and researchers* should create or utilize existing platforms to share and visually present requirements. These platforms should allow all sides of the building team to collaborate, and share ideas and tools in an environment that could enable aspects of RE and AI to be linked and traced. The platform should also allow to capture and present the requirements from different stakeholders' perspective.

6.5. Addressing Issues Related to Calculating Trade-offs

Trade-offs should be calculated in order to prioritize the importance of different AI-based software requirements. Google provides a set of guidelines for creating human-centered AI and indicated the importance of weighing different trade-offs, especially in the case of predictive AI systems. For instance, an incorrect prediction in diagnosing a cancer patient would have more significant stakes than providing a movie recommendation that the user does not like. When calculating trade-offs, what other requirements can make up for the lost cause? For example, the study in [81] experimented with ML algorithms to find which one could produce better results with lower costs. They found that specific ML algorithms performed better than others. So in such cases, trade-offs could be replaced with other measures that can make up for the loss. Another study explained that some algorithms provide more reliable predictions but are not easily explained. Whereas others can better explain why a prediction is delivered, but predictions are less in confidence[133]. So how do we decide on which algorithm to choose? In what situations do we prefer to use explainable algorithms vs higher confidence.

Google PAIR pointed out the importance of calculating trade-offs in the reward function, and to evaluate and weigh the risks of choosing an appropriate reward function that would suit the user's needs. For example, in an ML model that uses classification such as a notification system in an autonomous car, a false negative would *not* notify a sleeping driver in case of an emergency (weightage to precision), which could lead to deadly consequences. While having too many notifications that are false positives (weightage to recall) can lead the driver to ignore them [126]. When building AI software, we should always calculate the trade-off between precision and recall, or any other relevant metrics for a given AI solution. When can we choose precision over recall or vice versa? For example, Dimatteo et al. [126] explained that in the notification system finding alternative human-centered ways to engage the driver, using recall might be a more feasible and safer choice. Furthermore, Berry [134] reports that the choice of the reward function should also reflect on how the human would perform the task manually and how much impact and value would the chosen reward function provide in return.

Another common trade-off is that some requirements might contradict others (either positively or negatively). For example, ethics, trust, and transparency can conflict with privacy, safety, and security [135]. How do we calculate the trade-off between transparency and privacy? One might consider the trade-off depending on the application domain. So a transparent recommendation system might not affect privacy. However, a transparent medical application might reveal some private information that might go against the rules and regulations of the organization. Trade-offs should be calculated carefully in order to prioritize the importance of requirements. When do we decide on what to choose? Which model fits best to the organization's needs? How are trade-offs calculated? These are all questions that should be addressed during the initial phases.

> We recommend that *practitioners* list all trade-offs and the corresponding choices with the rationale for the decisions made. These trade-offs should be made explicit along with the requirements. Also we recommend that *researchers* include trade-offs in modeling languages for RE4AI.

6.6. Lack of empirical evaluations on Ethics, Explainability, and Trust

In 2018, the European Commission formed a set of ethical guidelines for trustworthy AI [136]. The guidelines emphasized that to create trustworthy AI, the outcome should be lawful, ethical, and robust. Lawful means that developers should comply with all legal regulations. For example, it should abide by the European General Data Protection Regulation (GDPR) rules and regulations. Ethical, concluded that the system should have respect for humans, prevent harm, be fair and explicable. Finally, robust meant that delivered systems should be safe, secure, and reliable. With the introduction of these guidelines, more research is growing towards creating trustworthy and responsible AI software. However, certain concepts, such as ethics are not clearly defined and are thus difficult to apply [135]. There is a scope for more and in-depth investigation in this area.

Explainable AI systems can help build up users' trust and strengthen their ability to form a more accurate mental model of the product [137]. However, they are still not applied to AI development as they should. Amershi et al [138]. identified eighteen guidelines for human-centered AI interaction by examining research from over 20 years of human-centered interaction with AI systems. The study involved over 150 AI design recommendations collected from research and industrial sources. The study demonstrated that most violations made in line with the guidelines in current AI systems were linked to explainability.

People usually expect intelligent systems to provide an explanation similar to how a person behaves and respond [137]. Miller argues that most developers provide explanations based on their own intuition or understanding of what a good explanation would be, rather than explaining from a social science perspective [139]. Explainable AI could benefit from implementing models that have been used to generate explanations in social and behavioural science research. However, there is not much research that uses any of these models [140]. Existing models such as Prospector, “an interactive visual analytic system" provides explanations to black-boxed machine learning predictions that are hard to explain [133]. Guidotti et al. also provides a survey listing the different methods used to explain black-box models [141]. AI developers should benefit from such models and incorporate them into RE practices.

Explaining programs that provide unpredictable outputs can be a challenging task. However, when given correctly, explanations to predictions can improve people’s choices in decision making when it comes to using and adopting AI software [142, 143]. Our mapping study found limited empirical work on explainability and none for ethics and trust.

> We find the need for future *researchers* to conduct more empirical research on RE for explainable, trustworthy and ethical AI software.

6.7. Lack of studies on RE for Human-centered AI

Many new software systems are moving towards using various AI-based software components [11]. However, it is still far more common to focus on the technical side than on diverse human-centered aspects, such as different user age, gender, ethnicity, and emotions, which are often ignored in the process [144, 145, 146]. Shneiderman [147] explains that the focus on machine autonomy in AI software systems can lead to hidden biases. For example, an ML algorithm used in healthcare favoured white people over people of color after the developers failed to include race as a variable when training the algorithm [148]; or NLP chatbots are known to learn social biases because of the underlying generic text they are built on [149]. Human-centered aspects in RE (in general) have gained traction in the past years, such as emotions [150], gender [151, 152], age [153], and mental and physical challenges [154]. However, we found limited studies that focused on RE for human-centered AI and most RE4AI studies lacked human-centered aspects when specifying requirements.

We think that identifying the user needs is just the first step in building more human-centered AI-based software systems. We need to identify the human-centered needs for other AI-related tasks such as collecting data, choosing the model, explaining outputs, and providing feedback. For example, how do we make sure that the data collected is inclusive, responsible, and non-biased? We found two studies that focused on human-centered aspects in RE4AI [101, 94]. The first study [101] involved analyzing requirements for a social robot and emphasizing on emotion when designing the system. The robot’s purpose was to interact with elderly people via speech recognition in an attempt to slow down the chance of developing dementia. The second study [94] presents an RE approach for incorporating context-specific human-centered explanations for AI-generated feedback provided to the end users. Despite some preliminary work on the topic, we found limited work overall and in particular empirical evaluations on the topic of human-centered AI.

Organisations such as Microsoft have proposed toolkits to promote ethical and responsible AI software development, e.g., “The HAX Toolkit Project” [155]. Although, the HAX toolkit addresses human-centered aspects related to mitigating failures in human-system-interaction, it is limited to the design phase. We further believe that there is a need to create an RE approach that includes all human-centered aspects when building AI-based software.

> We recommend that *researchers* should take into consideration the industrial guidelines for building human-centered AI (in addition to existing human-centered related studies), such as Google Pair [128], Apple’s guidelines on human-centered AI [156] and Microsoft’s guidelines for human-centered AI interaction [138, 155].

7. Conclusion

AI-based techniques have recently become much more embedded into many software systems and are increasingly used by companies to improve performance and reduce costs. However, using existing RE techniques for current AI systems are challenging due to the different nature of the development process between traditional software engineering methods and AI-based systems. Current AI systems show a lack of integration with existing RE tools and methodologies, with limited research on the topic. In this paper, we have presented a mapping study that has identified 43 studies on RE4AI. We have analyzed different frameworks, methodologies, tools, modeling techniques, and requirements notations currently proposed for RE4AI. Our results show that most studies favored UML and GORE to model requirements. Some favored GORE as it had better support for NFR's and business rules, whereas others favored UML as non-software engineers found them easier to use. We also noticed an increased interest over the past year in using DSM to present requirements. We found that research presented for the autonomous industry is more established. Whereas work on ethics is more theoretical and has just recently gained attention in the research industry. Our findings identified many issues and challenges in current RE for AI techniques. For example, defining requirements, explaining predictions, addressing ethical issues, and issues with data requirements. Another major issue presented in the literature was the lack of integration between software engineers and data scientists. We concluded by providing a list of research recommendations for future work. With the lack of current practices available, there is a need to introduce and research new methodologies alongside integrating existing RE techniques. The next step should involve documenting any requirements for AI systems, identifying modeling languages, and creating a platform for requirements engineers and data scientists to collaborate and share their ideas. In the future, we want to evaluate how these techniques can be adopted between different AI areas and investigate specialisations (customisations or new ideas) that need to be developed to meet the needs in specific AI fields.

Acknowledgements

Grundy is supported by ARC Laureate Fellowship FL190100035.

References

[1] L. E. Holmquist, Intelligence on tap: artificial intelligence as a new design material, interactions 24 (4) (2017) 28–33.

[2] J. Schroeder, D. Holzner, C. Berger, C.-J. Hoel, L. Laine, A. Magnusson, Design and evaluation of a customizable multi-domain reference architecture on top of product lines of self-driving heavy vehicles-an industrial case study, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Vol. 2, IEEE, 2015, pp. 189–198.

[3] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, Artificial intelligence in healthcare: past, present and future, Stroke and vascular neurology 2 (4) (2017) 230–243.

[4] D. Li, Y. Du, Artificial intelligence with uncertainty, CRC press, 2017.

[5] M. Mekni, Z. Baani, D. Sulieman, A smart virtual assistant for students, in: Proceedings of the 3rd International Conference on Applications of Intelligent Systems, 2020, pp. 1–6.

[6] D. Theosaksomo, D. H. Widyantoro, Conversational recommender system chatbot based on functional requirement, in: 2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA), IEEE, 2019, pp. 154–159.

[7] H. Lee, A. Kobsa, Confident privacy decision-making in IoT environments, ACM Transactions on Computer-Human Interaction (TOCHI) 27 (1) (2019) 1–39.

[8] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, in: Advances in neural information processing systems, 2015, pp. 2503–2511.

[9] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, H. H. Olsson, Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions, Information and software technology 127 (2020) 106368.

[10] H. Van Vliet, H. Van Vliet, J. Van Vliet, Software engineering: principles and practice, Vol. 13, Citeseer, 2008.

[11] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, T. Zimmermann, Software engineering for machine learning: a case study, in: Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, IEEE Press, 2019, pp. 291–300.

[12] M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[13] A. Arpteg, B. Brinne, L. Crnkovic-Friis, J. Bosch, Software engineering challenges of deep learning, in: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2018, pp. 50–59.

[14] M. Agarwal, S. Goel, Expert system and it's requirement engineering process, in: International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), IEEE, 2014, pp. 1–4.

[15] F. Khomh, B. Adams, J. Cheng, M. Fokaefs, G. Antoniol, Software engineering for machine-learning applications: The road ahead, IEEE Software 35 (5) (2018) 81–84.

[16] H. Belani, M. Vukovic, Ž. Car, Requirements engineering challenges in building AI-based complex systems, in: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), IEEE, 2019, pp. 252–255.

[17] W. Ertel, Introduction to artificial intelligence, Springer, 2018.

[18] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer, Automated checking of conformance to requirements templates using natural language processing, IEEE transactions on Software Engineering 41 (10) (2015) 944–968.

[19] S. Abualhaija, C. Arora, M. Sabetzadeh, L. C. Briand, M. Traynor, Automated demarcation of requirements in textual specifications: a machine learning-based approach, Empirical Software Engineering 25 (6) (2020) 5454–5497.

[20] S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, L. C. Briand, Using domain-specific corpora for improved handling of ambiguity in requirements, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, 2021, pp. 1485–1497.

[21] K. Zamani, D. Zowghi, C. Arora, Machine learning in requirements engineering: A mapping study, in: 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), IEEE, 2021, pp. 116–125.

[22] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering.

[23] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and software technology 64 (2015) 1–18.

[24] K. Ahmad, M. Bano, M. Abdelrazek, C. Arora, J. Grundy, What's up with requirements engineering for artificial intelligence systems?, IEEE 29th International Requirements Engineering Conference (RE).

[25] D. Amyot, G. Mussbacher, S. Ghanavati, J. Kealey, GRL modeling and analysis with jUCMNav, iStar 766 (2011) 160–162.

[26] Sirius Eclipse, Modeling tool, https://www.eclipse.org/sirius/.

[27] J. Dick, E. Hull, K. Jackson, Requirements engineering, Springer, 2017.

[28] L. S. Wheatcraft, M. J. Ryan, Communicating requirements–effectively!, in: INCOSE International Symposium, Vol. 28, Wiley Online Library, 2018, pp. 716–732.

[29] B. Nuseibeh, S. Easterbrook, Requirements engineering: a roadmap, in: Proceedings of the Conference on the Future of Software Engineering, 2000, pp. 35–46.

[30] G. Koelsch, Requirements writing for system engineering, Springer, 2016.

[31] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, S. Shamshirband, A systematic literature review on agile requirements engineering practices and challenges, Computers in human behavior 51 (2015) 915–929.

[32] B. C. Hu, R. Salay, K. Czarnecki, M. Rahimi, G. Selim, M. Chechik, Towards requirements specification for machine-learned perception based on human performance, in: Workshop on Artificial Intelligence for Requirements Engineering (AIRE), IEEE, 2020.

[33] A. Lapouchnian, Goal-oriented requirements engineering: An overview of the current research, University of Toronto 32.

[34] D. Amyot, G. Mussbacher, User requirements notation: the first ten years, the next ten years, JSW 6 (5) (2011) 747–768.

[35] E. Gonçalves, M. A. de Oliveira, I. Monteiro, J. Castro, J. Araújo, Understanding what is important in iStar extension proposals: the viewpoint of researchers, Requirements Engineering 24 (1) (2019) 55–84.

[36] M. B. Duran, G. Mussbacher, Reusability in goal modeling: a systematic literature review, Information and Software Technology 110 (2019) 156–173.

[37] H. Kuwajima, H. Yasuoka, T. Nakae, Engineering problems in machine learning systems, Machine Learning (2020) 1–24.

[38] J. Bosch, H. H. Olsson, I. Crnkovic, Engineering ai systems: A research agenda, in: Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, IGI global, 2021, pp. 1–19.

[39] L. M. Cysneiros, M. Raffi, J. C. S. do Prado Leite, Software transparency as a key requirement for self-driving cars, in: 2018 IEEE 26th International Requirements Engineering Conference (RE), IEEE, 2018, pp. 382–387.

[40] F. Houdek, S. Schmerler, Automotive future and its impact on requirements engineering., in: REFSQ Workshops, 2017.

[41] L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, I. Crnkovic, A taxonomy of software engineering challenges for machine learning systems: An empirical investigation, in: International Conference on Agile Software Development, Springer, 2019, pp. 227–243.

[42] B. Kostova, S. Gurses, A. Wegmann, On the interplay between requirements, engineering, and artificial intelligence., in: REFSQ Workshops, 2020.

[43] L. Chazette, Mitigating challenges in the elicitation and analysis of transparency requirements, in: 2019 IEEE 27th International Requirements Engineering Conference (RE), IEEE, 2019, pp. 470–475.

[44] H. Kaindl, J. Ferdigg, Towards an extended requirements problem formulation for superintelligence safety, in: 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), IEEE, 2020, pp. 33–38.

[45] H.-M. Heyn, E. Knauss, A. P. Muhammad, O. Eriksson, J. Linder, P. Subbiah, S. K. Pradhan, S. Tungal, Requirement engineering challenges for ai-intense systems development, in: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), IEEE, 2021, pp. 89–96.

[46] M. Camilli, M. Felderer, A. Giusti, D. T. Matt, A. Perini, B. Russo, A. Susi, Risk-driven compliance assurance for collaborative ai systems: A vision paper, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2021, pp. 123–130.

[47] L. A. Odong, A. Perini, A. Susi, Requirements engineering for collaborative artificial intelligence systems: A literature survey, in: International Conference on Research Challenges in Information Science, Springer, 2022, pp. 409–425.

[48] D. Brugali, Non-functional requirements in robotic systems: Challenges and state of the art, in: 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR), IEEE, 2019, pp. 743–748.

[49] R. Guizzardi, G. Amaral, G. Guizzardi, J. Mylopoulos, Ethical requirements for AI systems, in: Canadian Conference on Artificial Intelligence, Springer, 2020, pp. 251–256.

[50] G. D. Crnkovic, B. Çürüklü, Robots: ethical by design, Ethics and Information Technology 14 (1) (2012) 61–71.

[51] M. Thinyane, L. Goldkind, A multi-aspectual requirements analysis for artificial intelligence for well-being, in: 2020 IEEE First International Workshop on Requirements Engineering for Well-Being, Aging, and Health (REWBAH), IEEE, 2020, pp. 11–18.

[52] M.-L. Nguyen, T. Phung, D.-H. Ly, H.-L. Truong, Holistic explainability requirements for end-to-end machine learning in iot cloud systems, in: 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), IEEE, 2021, pp. 188–194.

[53] H. Felzmann, E. F. Villaronga, C. Lutz, A. Tamò-Larrieux, Transparency you can trust: Transparency requirements for artificial intelligence between legal norms and contextual concerns, Big Data & Society 6 (1) (2019) 2053951719860542.

[54] T. M. Fagbola, S. C. Thakur, Towards the development of artificial intelligence-based systems: Human-centered functional requirements and open problems, in: 2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), IEEE, 2019, pp. 200–204.

[55] S. Martínez-Fernández, J. Bogner, X. Franch, M. Oriol, J. Siebert, A. Trendowicz, A. M. Vollmer, S. Wagner, Software engineering for ai-based systems: A survey, arXiv preprint arXiv:2105.01984.

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

[57] M. Bano, D. Zowghi, N. Ikram, Systematic reviews in requirements engineering: A tertiary study, in: 2014 IEEE 4th International Workshop on Empirical Requirements Engineering (EmpiRE), IEEE, 2014, pp. 9–16.

[58] T. Ambreen, N. Ikram, M. Usman, M. Niazi, Empirical research in requirements engineering: trends and opportunities, Requirements Engineering 23 (1) (2018) 63–95.

[59] G. D. Hager, R. Bryant, E. Horvitz, M. Mataric, V. Honavar, Advances in artificial intelligence require progress across all of computer science, arXiv preprint arXiv:1707.04352.

[60] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg, S. Kalyanakrishnan, E. Kamar, S. Kraus, et al., Artificial intelligence and life in 2030: the one hundred year study on artificial intelligence.

[61] J. Bosch, H. H. Olsson, I. Crnkovic, It takes three to tango: Requirement, outcome/data, and AI driven development., in: SiBW, 2018, pp. 177–192.

[62] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th international conference on evaluation and assessment in software engineering, 2014, pp. 1–10.

[63] M. A. Köhl, K. Baum, M. Langer, D. Oster, T. Speith, D. Bohlender, Explainability as a non-functional requirement, in: 2019 IEEE 27th International Requirements Engineering Conference (RE), IEEE, 2019, pp. 363–368.

[64] F. B. Aydemir, F. Dalpiaz, A roadmap for ethics-aware software engineering, in: 2018 IEEE/ACM International Workshop on Software Fairness (FairWare), IEEE, 2018, pp. 15–21.

[65] S. Nalchigar, E. Yu, K. Keshavjee, Modeling machine learning requirements from three perspectives: a case report from the healthcare domain, Requirements Engineering 26 (2) (2021) 237–254.

[66] A. Vogelsang, M. Borg, Requirements engineering for machine learning: Perspectives from data scientists, IEEE International Requirements Engineering Conference Workshops.

[67] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[68] P. A. Glasow, Fundamentals of survey research methodology, Retrieved January 18 (2005) 2013.

[69] G. G. Gable, Integrating case study and survey research methods: an example in information systems, European journal of information systems 3 (2) (1994) 112–126.

[70] W. Tellis, Application of a case study methodology, The qualitative report 3 (3) (1997) 1–19.

[71] C. B. Meyer, A case in case study methodology, Field methods 13 (4) (2001) 329–352.

[72] C. Wohlin, Case study research in software engineering—it is a case, and it is a study, but is it a case study?, Information and Software Technology 133 (2021) 106514.

[73] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: Guide to advanced empirical software engineering, Springer, 2008, pp. 285–311.

[74] M. Bonfe, F. Boriero, R. Dodi, P. Fiorini, A. Morandi, R. Muradore, L. Pasquale, A. Sanna, C. Secchi, Towards automated surgical robotics: A requirements engineering approach, in: 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), IEEE, 2012, pp. 56–61.

[75] K. Gruber, J. Huemer, A. Zimmermann, R. Maschotta, Integrated description of functional and non-functional requirements for automotive systems design using SysML, in: 2017 7th IEEE International Conference on System Engineering and Technology (ICSET), IEEE, 2017, pp. 27–31.

[76] K. Sandkuhl, Putting AI into context-method support for the introduction of artificial intelligence into organizations, in: 2019 IEEE 21st Conference on Business Informatics (CBI), Vol. 1, IEEE, 2019, pp. 157–164.

[77] M. Rahimi, J. L. Guo, S. Kokaly, M. Chechik, Toward requirements specification for machine-learned components, in: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), IEEE, 2019, pp. 241–244.

[78] J. Jakob, K. Kugele, J. Tick, Defining camera-based traffic scenarios and use cases for the visually impaired by means of expert interviews, in: 2017 IEEE 14th International Scientific Conference on Informatics, IEEE, 2017, pp. 128–133.

[79] C. E. Tuncali, G. Fainekos, D. Prokhorov, H. Ito, J. Kapinski, Requirements-driven test generation for autonomous vehicles with machine learning components, IEEE Transactions on Intelligent Vehicles 5 (2) (2019) 265–280.

[80] J. S. Becker, Partial consistency for requirement engineering with traffic sequence charts., in: Software Engineering (Workshops), 2020.

[81] C. Shin, S. Rho, H. Lee, W. Rhee, Data requirements for applying machine learning to energy disaggregation, Energies 12 (9) (2019) 1696.

[82] G. Dimitrakopoulos, E. Kavakli, P. Loucopoulos, D. Anagnostopoulos, T. Zographos, A capability-oriented modelling and simulation approach for autonomous vehicle management, Simulation Modelling Practice and Theory 91 (2019) 28–47.

[83] D. Weihrauch, P. A. Schindler, W. Sihn, A conceptual model for developing a smart process control system, Procedia CIRP 67 (2018) 386–391.

[84] S. Fenn, A. Mendes, D. M. Budden, Addressing the non-functional requirements of computer vision systems: a case study, Machine Vision and Applications 27 (1) (2016) 77–86.

[85] K. Neace, R. Roncace, P. Fomin, Goal model analysis of autonomy requirements for unmanned aircraft systems, Requirements Engineering 23 (4) (2018) 509–555.

[86] J. Lockerbie, N. A. Maiden, Using a requirements modelling language to co-design intelligent support for people living with dementia., in: REFSQ Workshops, 2020.

[87] K. Nakamichi, K. Ohashi, I. Namba, R. Yamamoto, M. Aoyama, L. Joeckel, J. Siebert, J. Heidrich, Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation, in: 2020 IEEE 28th International Requirements Engineering Conference (RE), IEEE, 2020, pp. 260–270.

[88] H. Challa, N. Niu, R. Johnson, Faulty requirements made valuable: On the role of data quality in deep learning, in: 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), IEEE, 2020, pp. 61–69.

[89] B. Ries, N. Guelfi, B. Jahic, An MDE method for improving deep learning dataset requirements engineering using alloy and UML, in: Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development, SCITEPRESS, 2021, pp. 41–52.

[90] K. Olmos-Sánchez, J. Rodas-Osollo, Helping organizations manage the innovation process to join the cognitive era, in: 2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT), IEEE, 2020, pp. 1–10.

[91] H. Samin, L. H. G. Paucar, N. Bencomo, P. Sawyer, Towards priority-awareness in autonomous intelligent systems, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 1328–1337.

[92] D. Cirqueira, D. Nedbal, M. Helfert, M. Bezbradica, Scenario-based requirements elicitation for user-centric explainable ai, in: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, 2020, pp. 321–341.

[93] C. Ntakolia, G. Dimas, D. K. Iakovidis, User-centered system design for assisted navigation of visually impaired individuals in outdoor cultural environments, Universal Access in the Information Society (2020) 1–26.

[94] T. A. Schoonderwoerd, W. Jorritsma, M. A. Neerincx, K. van den Bosch, Human-centered xai: Developing design patterns for explanations of clinical decision support systems, International Journal of Human-Computer Studies (2021) 102684.

[95] M. N. Islam, S. R. Khan, N. N. Islam, M. Rezwan-A-Rownok, S. R. Zaman, S. R. Zaman, A mobile application for mental health care during covid-19 pandemic: Development and usability evaluation with system usability scale, in: International Conference on Computational Intelligence in Information System, Springer, 2021, pp. 33–42.

[96] J. Cleland-Huang, A. Agrawal, M. N. A. Islam, E. Tsai, M. Van Speybroeck, M. Vierhauser, Requirements-driven configuration of emergency response missions with small aerial vehicles, in: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A-Volume A, 2020, pp. 1–12.

[97] M. Hall, D. Harborne, R. Tomsett, V. Galetic, S. Quintana-Amate, A. Nottle, A. Preece, A systematic method to understand requirements for explainable ai (xai) systems, in: Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China, Vol. 11, 2019.

[98] K. M. Habibullah, J. Horkoff, Non-functional requirements for machine learning: Understanding current use and challenges in industry, IEEE 29th International Requirements Engineering Conference (RE).

[99] L. Rivero, C. Portela, J. Boaro, P. Santos, V. Rego, G. Braz Junior, A. Paiva, E. Alves, M. Oliveira, R. Moraes, et al., Lessons learned from applying requirements and design techniques in the development of a machine learning system for predicting lawsuits against power companies, in: International Conference on Human-Computer Interaction, Springer, 2021, pp. 227–243.

[100] J. Horkoff, Non-functional requirements for machine learning: Challenges and new directions, in: 2019 IEEE 27th International Requirements Engineering Conference (RE), IEEE, 2019, pp. 386–391.

[101] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, Functional requirements and design issues for a socially assistive robot for elderly people with mild cognitive impairments, in: 2013 IEEE RO-MAN, IEEE, 2013, pp. 768–773.

[102] H. H. Altarturi, K.-Y. Ng, M. I. H. Ninggal, A. S. A. Nazri, A. A. Abd Ghani, A requirement engineering model for big data software, in: 2017 IEEE Conference on Big Data and Analytics (ICBDA), IEEE, 2017, pp. 111–117.

[103] J. K. Ang, S. B. Leong, C. F. Lee, U. K. Yusof, Requirement engineering techniques in developing expert systems, in: 2011 IEEE Symposium on Computers & Informatics, IEEE, 2011, pp. 640–645.

[104] F. Ishikawa, Y. Matsuno, Evidence-driven requirements engineering for uncertainty of machine learning-based systems, in: 2020 IEEE 28th International Requirements Engineering Conference (RE), IEEE, 2020, pp. 346–351.

[105] H. Kuwajima, F. Ishikawa, Adapting SQuaRE for quality assessment of artificial intelligence systems, in: 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE, 2019, pp. 13–18.

[106] G. Amaral, R. Guizzardi, G. Guizzardi, J. Mylopoulos, Ontology-based modeling and analysis of trustworthiness requirements: Preliminary results, in: International Conference on Conceptual Modeling, Springer, 2020, pp. 342–352.

[107] A. Agrawal, J. Cleland-Huang, J.-P. Steghöfer, Model-driven requirements for humans-on-the-loop multi-uav missions, in: 2020 IEEE Tenth International Model-Driven Requirements Engineering (MoDRE), IEEE, 2020, pp. 1–10.

[108] D. Clauer, J. Fottner, E. Rauch, M. Prüglmeier, Usage of autonomous mobile robots outdoors-an axiomatic design approach, Procedia CIRP 96 (2021) 242–247.

[109] P. Khatamino, M. B. Camli, B. Öztekin, U. Gozumoglu, E. Tortumlu, H. M. Gezer, An nlp-based chatbot to facilitate re activities: An experience paper on human resources application., in: REFSQ Workshops, 2021.

[110] M. Schwammberger, A quest of self-explainability: When causal diagrams meet autonomous urban traffic manoeuvres, in: 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), IEEE, 2021, pp. 195–199.

[111] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative research in psychology 3 (2) (2006) 77–101.

[112] C. Urquhart, Grounded theory for qualitative research: A practical guide, Sage, 2012.

[113] J. Bach, J. Langner, S. Otten, M. Holzäpfel, E. Sax, Data-driven development, a complementing approach for automotive systems engineering, in: 2017 IEEE International Systems Engineering Symposium (ISSE), IEEE, 2017, pp. 1–6.

[114] S. Nalchigar, E. Yu, Business-driven data analytics: a conceptual modeling framework, Data & Knowledge Engineering 117 (2018) 359–372.

[115] L. Baresi, L. Pasquale, P. Spoletini, Fuzzy goals for requirements-driven adaptation, in: 2010 18th IEEE international requirements engineering conference, IEEE, 2010, pp. 125–134.

[116] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, S. Sankaranarayanan, Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications, in: Lectures on Runtime Verification, Springer, 2018, pp. 135–175.

[117] M. M. Glymour, S. Greenland, Causal diagrams, Modern epidemiology 3 (2008) 183–209.

[118] J. Pearl, Causal diagrams for empirical research, Biometrika 82 (4) (1995) 669–688.

[119] S. Greenland, J. Pearl, J. M. Robins, Causal diagrams for epidemiologic research, Epidemiology (1999) 37–48.

[120] M. Blumreiter, J. Greenyer, F. J. C. Garcia, V. Klös, M. Schwammberger, C. Sommer, A. Vogelsang, A. Wortmann, Towards self-explainable cyber-physical systems, in: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, 2019, pp. 543–548.

[121] M. Fowler, Domain-specific languages, Pearson Education, 2010.

[122] E. Vassev, M. Hinchey, Autonomy requirements engineering, in: Autonomy Requirements Engineering for Space Missions, Springer, 2014, pp. 105–172.

[123] V. Viyović, M. Maksimović, B. Perisić, Sirius: A rapid development of dsm graphical editor, in: IEEE 18th International Conference on Intelligent Engineering Systems INES 2014, IEEE, 2014, pp. 233–238.

[124] E. Vassev, M. Hinchey, On the autonomy requirements for space missions, in: 16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013), IEEE, 2013, pp. 1–10.

[125] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, Bdd100k: A diverse driving dataset for heterogeneous multitask learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2636–2645.

[126] J. DiMatteo, D. M. Berry, K. Czarnecki, Requirements for monitoring inattention of the responsible human in an autonomous vehicle: The recall and precision tradeoff., in: REFSQ Workshops, 2020.

[127] X. Zhou, Y. Jin, H. Zhang, S. Li, X. Huang, A map of threats to validity of systematic literature reviews in software engineering, in: 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), IEEE, 2016, pp. 153–160.

[128] Google Research, The people + AI guidebook, [online] Available at: https://research.google/teams/brain/pair/ [Accessed 1 April 2020] (2019).

[129] F. Wang, J. Liu, Networked wireless sensor data collection: issues, challenges, and approaches, IEEE Communications Surveys & Tutorials 13 (4) (2010) 673–687.

[130] O. Wang, B. Cheng, T. Hoang, C. Arora, X. Liu, Virtual reality enabled human-centric requirements engineering, in: 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW), IEEE, 2021, pp. 159–164.

[131] M. K. Hong, A. Fourney, D. DeBellis, S. Amershi, Planning for natural language failures with the ai playbook, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–11.

[132] Q. Z. Chen, T. Schnabel, B. Nushi, S. Amershi, Hint: Integration testing for ai-based features with humans in the loop, in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 549–565.

[133] J. Krause, A. Perer, K. Ng, Interacting with predictions: Visual inspection of black-box machine learning models, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 5686–5697.

[134] D. M. Berry, Requirements engineering for artificial intelligence: What is a requirements specification for an artificial intelligence?, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2022, pp. 19–25.

[135] L. M. Cysneiros, J. C. S. do Prado Leite, Non-functional requirements orienting the development of socially responsible software, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2020, pp. 335–342.

[136] H.-L. E. G. on Artificial Intelligence, Ethics guidelines for trustworthy AIDocument can be accessed at: https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

[137] M. M. De Graaf, B. F. Malle, How people explain action (and autonomous intelligent systems should too), in: 2017 AAAI Fall Symposium Series, 2017.

[138] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al., Guidelines for human-AI interaction, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019, pp. 1–13.

[139] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38.

[140] T. Miller, P. Howe, L. Sonenberg, Explainable AI: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences, IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI), 36–42, URL http://people.eng.unimelb.edu.au/tmiller/pubs/explanation-inmates.pdf.

[141] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (5) (2018) 1–42.

[142] D. Wang, Q. Yang, A. Abdul, B. Y. Lim, Designing theory-driven user-centric explainable AI, in: Proceedings of the 2019 CHI conference on human factors in computing systems, 2019, pp. 1–15.

[143] S. Amershi, M. Cakmak, W. B. Knox, T. Kulesza, Power to the people: The role of humans in interactive machine learning, Ai Magazine 35 (4) (2014) 105–120.

[144] M. Maguire, Methods to support human-centred design, International journal of human-computer studies 55 (4) (2001) 587–634.

[145] A. Schmidt, Interactive human centered artificial intelligence: a definition and research challenges, in: Proceedings of the International Conference on Advanced Visual Interfaces, 2020, pp. 1–4.

[146] J. C. Grundy, Impact of end user human aspects on software engineering., in: ENASE, 2021, pp. 9–20.

[147] B. Shneiderman, Human-centered ai, Issues in Science and Technology 37 (2) (2021) 56–61.

[148] Carolyn Y. Johnson, Racial bias in a medical algorithm favors white patients over sicker black patients, [online] https://www.washingtonpost.com/health/2019/10/24/racial-bias-medical-algorithm-favors-white-patients-over-sicker-black-patients// [Accessed 1 Aug 2022].

[149] N. Lee, A. Madotto, P. Fung, Exploring social bias in chatbots using stereotype knowledge., in: WNLP@ ACL, 2019, pp. 177–180.

[150] T. Miller, S. Pedell, L. Sterling, F. Vetere, S. Howard, Understanding socially oriented roles and goals through motivational modelling, Journal of Systems and Software 85 (9) (2012) 2160–2170.

[151] M. Burnett, A. Peters, C. Hill, N. Elarief, Finding gender-inclusiveness software issues with GenderMag: a field investigation, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 2586–2598.

[152] M. Vorvoreanu, L. Zhang, Y.-H. Huang, C. Hilderbrand, Z. Steine-Hanson, M. Burnett, from gender biases to gender-inclusive design: An empirical investigation, in: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI'19). ACM, New York, NY, USA. https://doi. org/10.1145/3290605.3300283, 2019.

[153] J. McIntosh, X. Du, Z. Wu, G. Truong, Q. Ly, R. How, S. Viswanathan, T. Kanij, Evaluating age bias in e-commerce, in: 2021 IEEE/ACM 13th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, 2021, pp. 31–40.

[154] J. Grundy, K. Mouzakis, R. Vasa, A. Cain, M. Curumsing, M. Abdelrazek, N. Fernando, Supporting diverse challenges of ageing with digital enhanced living solutions., Studies in health technology and informatics 246 (2018) 75–90.

[155] Microsoft, The hax toolkit project, [online] https://www.microsoft.com/en-us/research/project/hax-toolkit/ [Accessed 1 June 2022].

[156] Apple Developer, Human interface guidelines, [online] https://developer.apple.com/design/human-interface-guidelines/technologies/machine-learning/introduction [Accessed 1 May 2020].