Automated Software Engineering (2026) 33:31
https://doi.org/10.1007/s10515-025-00578-6

Artificial intelligence and machine learning in enhancing software project management processes: A systematic literature review

Usama Ali[1] · Mehwish Naseer[1]

Received: 24 January 2025 / Accepted: 3 November 2025

Abstract
The growing complexity of software systems and the increasing demand for global software development necessitate innovative approaches to enhance project management, quality assurance, and risk mitigation. In this study, Artificial Intelligence (AI) and Machine Learning (ML) techniques are examined for overcoming problems in software project management, notably effort estimation, scheduling, resource allocation, risk management, and defect prediction. By systematically reviewing the literature, we show that AI/ML models like Support Vector Machines, neural networks, and ensemble learning can enhance estimation accuracy, maximize resource utilization, and reduce risks. Furthermore, the practical benefits and challenges of implementing an AI/ML system into a real-world system are discussed using real-world case studies, which include data quality and integration issues, and the interpretability of the model. In addition, advanced models, such as graph convolutional networks and deep neural networks, hold great promise as a defect prediction and bug severity classifier. The focus of this research is to leverage the transformative capabilities of AI/ML toward defect-free, efficient, and customer-centric software development. Finally, it suggests future research interests, including integrating explanation model AI, managing data in a better way, and implementing the scalable hybrid approach to meet the newer needs of the industry.

Keywords Artificial Intelligence · Machine Learning · Software Project Management · Risk Management · Effort Estimation · Defect Prediction

✉ Mehwish Naseer
mehwish.naseer@ceme.nust.edu.pk

Usama Ali
usamaalieme40@gmail.com

1 Computer & Software Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), 44080 Islamabad, Pakistan

1 Introduction

1.1 Background

Software Project Management (SPM) is a fundamental aspect of software engineering that directly impacts the success or failure of software development projects. It includes the tasks of coordination, resource allocation, goal setting, progress monitoring, and making sure the project is completed in a timely fashion and within budget, meeting all other standards it was required to meet (El Bajta et al. 2018). As software projects by nature are complex, with evolving requirements and many uncertainties, effective SPM is important. The main objective of any SPM approach is to reduce risks, maximize resource allocation, and deliver on time (Marinho et al. 2014). Despite using different methods of software project management including Agile, Scrum, and Waterfall tools, these projects are still lagging when it comes to completion within the stipulated time and budget, with demands for even more features and low-quality (Andrei et al. 2019).

The recent two decades have seen great advances in Artificial Intelligence (AI) and Machine Learning (ML), which are now only intensifying and becoming a more critical part of many fields, including software engineering (Petrillo et al. 2019). Creating such systems is called AI because it mimics human intelligence by teaching machines to take up jobs that usually are manned by humans, including solving problems, making decisions, and understanding languages (Jarrahi 2018). ML is a subset of AI that deals with developing algorithms that result in a system that learns from data and continues to improve without explicit programming. Artificial intelligence and machine-learning techniques have produced significant advances in diverse fields, including healthcare, finance, and marketing. Helm et al. (2020). Over the past few years, they have begun to trickle into software project management to provide solutions to problems that project managers have been battling with for many years (Hashfi and Raharjo 2023).

SPM can be addressed by multiple challenges using AI and ML techniques. For example, AI systems can automatically perform any repetitive tasks such as resource allocation, schedule optimization, risk analysis, etc which save time and reduce human errors (Ogunbukola 2024). By analyzing historical project data, ML algorithms can aid in better decision-making by predicting potential risks, solving effort estimating and bid adjustment, and efficient resource allocation problems (Wan et al. 2020). Moreover, these technologies support better decision-making by helping project managers to make data-driven decisions in real time. Although AI and ML have the potential to penetrate and improve SPM, their implementation is still in its infancy, and few studies offer a general description of the landscape (Perifanis and Kitsios 2023).

The introduction of AI and ML into the software development lifecycle doesn't just provide new ways of improving prediction accuracy or cutting costs, but it also helps with delivering the software on time. Yet, it is difficult to integrate these advanced technologies into actual world project management workflows. The purpose of this review is to synthesize systematically the existing literature on the use of AI and ML in SPM, illustrating their applications, advantages, disadvantages, and research opportunities.

1.2 Problem statement

While traditional SPM methodologies have made significant progress, they still face inherent limitations due to the unpredictable nature of software projects (Leong et al. 2023). One of the most critical challenges that confront project managers is the area of time and cost estimation, resource allocation, and risk management. The reliance upon the human judgment and experience of conventional methods often results in errors and inefficiencies (Conboy and Carroll 2019). In particular, traditional methods of effort estimation (e.g., function point analysis, expert judgment) are unreliable in large-scale projects where requirements are dynamic and evolving. Project scheduling and risk management are also frequently not sophisticated enough to adjust to a scope change or unexpected challenge (Munialo and Muketha 2016).

AI and ML are great for addressing these limitations by automated processes, learning from past data, and predicting things that can help in making better decisions (Channe 2024). Cost and effort estimation can be improved by the use of machine learning models, such as decision trees, support vector machines, and neural networks, which analyze patterns in past projects and apply them to new projects (Rodríguez Sánchez et al. 2023). Genetic algorithms or reinforcement learning-based models detect optimal job scheduling and resource allocation considering needed constraints and project goals (Zhou et al. 2024). Moreover, ML can contribute to risk management by using historical project data to identify emerging risks as well as early warning to project managers (Kalogiannidis et al. 2024).

Although AI and ML provide great benefits, issues like access and quality of data, model interpretability, and organizational resistance to change limit their complete roll-out for use in SPM. Existing project management tools lack integration with many AI/ML applications; thus, it's difficult for organizations to adopt them in a unified way. Additionally, training of AI and ML models is often reliant upon huge amounts of high-quality data, which is neither straightforwardly available nor guaranteed in many real-world projects (Mohammad and Chirchir 2024). To address these issues, this systematic literature review (SLR) on SPM with AI and ML pulls together existing research on the topic and describes the benefits and limitations of the technology, and how it can be integrated into future software projects.

1.3 Research objectives

This study aims to systematically analyze the existing literature on AI and ML in SPM. The specific objectives of the research are as follows:

1. To identify the various AI and ML techniques used throughout different phases of SPM, such as effort estimation, scheduling, risk management, and quality assurance.

2. To evaluate how AI/ML techniques help to solve the key challenges that software development project managers currently deal with.
3. To measure the limitations and challenges associated with integrating AI and ML to traditional SPM practices.
4. To highlight future research directions and suggest ways to overcome the barriers to adopting AI and ML in SPM.

1.4 Research contributions

The research questions guiding this review are:

1. RQ1: What are the AI and ML techniques applied to SPM and which phases of SPM do these techniques target?
2. RQ2: What are the best AI and ML techniques to back the solutions to the traditional SPM challenges like cost estimation, scheduling, risk management, and resource allocation?
3. RQ3: What limitations and challenges exist in the application of AI/ML in SPM, and how can these be overcome?
4. RQ4: What future trends and research opportunities might be framed regarding the integration of AI/ML in SPM, and how they could influence the field in the upcoming years?

1.5 Organization of the paper

The structure of this paper is organized to provide a comprehensive analysis of the role of Artificial Intelligence (AI) and Machine Learning (ML) in enhancing software project management (SPM) processes. Chapter 2 begins with a detailed review of the related work, in which we first review the past research that exploits the use of AI or ML applications in SPM, put emphasis on existing studies, and analyze the major contributions and limitations of the earlier research. Section 3 presents the research methodology for this systematic literature review (SLR), including the study selection criteria, data extraction, and analysis steps to evaluate the gathered literature. In Section 4, we present the results of the review highlighting the application of different AI and ML techniques in various steps of SPM (effort estimation, scheduling, risk management, and quality assurance). Furthermore, this section presents an evaluation of the effectiveness of these techniques in dealing with project management problems. In Section 5, we discuss future research directions, identifying certain areas for follow-up work and how the use of AI and ML in the future will further impact the field. Section 6 of the paper finally summarizes the findings of the thesis, provides insights to the academia and industry practitioners, and finally, suggests practical recommendations for the implementation of AI and ML into the software project management processes.

2 Related work

During the last decade, researchers have conducted studies on the application of AI and ML in software engineering with a varied emphasis on different stages of the software development lifecycle such as design, testing, and maintenance. There are a few studies that have looked at AI applications over software engineering as a whole, which gives a sense of which tasks AI can help with Collins et al. (2021); Barenkamp et al. (2020). Although these studies are valuable, because they are often framed in terms of technical applications, they leave the managerial side of software development relatively unexplored.

One of the notable reviews is by Kostopoulos et.al (2024), which specifically addresses AI applications in decision support Systems. This study focuses on expert systems, decision support systems, and AI tools for project scheduling. Although it touches on a few feature areas, it is primarily concerned with more mature forms of AI including a set of rule-based systems and expert systems, which have been well surpassed by more current techniques such as deep learning and reinforcement learning (Kostopoulos et al. 2024).

Also important is the contribution of Chirra and Raza (2019), which discusses machine learning techniques in software cost estimation, a critical activity in SPM. They analyze different ML models used to estimate effort, including regression models and neural networks. The study is a valuable contribution to the cost estimation literature, but by focusing on only one aspect of SPM it fails to explore any other major components of project management (i.e. scheduling, risk management, quality assurance) and leaves a gap in the wider SPM context (Chirra and Reza 2019).

2.1 AI and ML in software project management

Specific applications of AI and ML in different project management phases of software projects have been explored in several studies showing how these techniques can improve the project processes' efficiency (Chirra and Reza 2019). For example, machine learning has shown a huge jump in using models for effort estimation versus traditional estimation. Regression trees, support vector machines, and other random forests are the algorithms that have been used to predict project effort by what they have learned about the existing project data (Lavingia et al. 2024). ML techniques provide more accuracy than expert judgment and function point analysis, especially in big projects with a lot of complexity – scenarios in which traditional methods choke (Wen et al. 2012).

The good old challenges of project scheduling have been partially approached by utilizing simple AI approaches like genetic algorithms, ant colony optimization, and particle swarm optimization to optimize task scheduling and resource allocation (Liu et al. 2014). However, these algorithms can allow for different kinds of constraints and uncertainty and produce more efficient schedules than traditional methods, in some cases adaptively compensating for changes in the level of project scope or resource availability (Santos et al. 2023).

Risk management too has seen significant application of AI. Machine learning models can analyze historical data to identify patterns that alert of potential risks and issues, proactively alerting organizations to risk mitigation strategies (Božić 2023). It is used to predict risks like cost overruns, delays, or quality issues before they appear in the project using ML models such as decision trees, clustering algorithms, and neural networks. The predictive capability of the model, along with the speed of its analysis, empowers project managers to take corrective actions early, while still minimizing the impact on timelines and budgets (Opeyemi Abayomi Odejide and Tolulope Esther Edunjobi 2024).

While the potential applications of AI and ML in SPM are promising, several hurdles must first be met for the widespread adoption of such techniques. The major issue with the availability and the quality of data is one. In software projects, historical data can often be incomplete or inconsistent; many AI/ML models require large volumes of high-quality data to be effective (Mohammed et al. 2022). Moreover, the integration of AI and ML tools into legacy SPM workflows can be challenging for organizations with legacy systems, or those who are hesitant to adopt new technologies (Mohammad and Chirchir 2024). In addition, AI/ML models are not very interpretable, many AI/ML models are 'black boxes' hindering project managers from understanding how predictions are made (Rudin 2019).

2.2 Contributions of this SLR

This systematic literature review builds on existing studies by offering a more comprehensive analysis of AI and ML techniques applied to software project management across all project phases. In contrast to previous reviews focusing on specific areas such as effort estimation or scheduling, the present study offers a broader perspective, including crucial areas such as risk management and monitoring, and quality assurance. Furthermore, it explores the pros and cons of using AI and ML in SPM to provide directions for future research and share its practical insight with those in the industry. It also marks important gaps in the existing literature, including the existence of AI/ML tools that can be integrated with existing SPM systems, the need for data quality, and a requirement for the use of more interpretable AI models.

3 Methodology

In this chapter, we offer a comprehensive explanation of the systematic methodology employed in the performing of the systematic literature review (SLR) of Artificial Intelligence (AI) and Machine Learning (ML) use in Software Project Management (SPM) improvement. It is intended to be applied to make the review thorough, reproducible, and bias-free. In this chapter, the review protocol is explained, the selection and screening process are described, and methods for extracting and synthesizing data from the reviewed studies are outlined.

3.1 Categorization framework for software-project-management phases

Best-practice guidance for software projects is codified in the PMBOK® Guide (7th ed.) (Institute 2021), ISO 21500:2021 “Project, programme and portfolio management” (Standardization 2021), and the IEEE-CS SWEBOK v4.0 knowledge-area model (Mohammad Zarour 2024). Across these references four management domains appear consistently: (i) effort estimation, (ii) schedule & resource allocation, (iii) risk management, and (iv) quality assurance & testing. Each domain functions as a discrete but interlocking subsystem for planning, executing and controlling software work, collectively spanning PMBOK process groups “Planning,” “Executing,” and “Monitoring & Controlling,” ISO 21500 management subjects “Resources,” “Time,” “Risk,” and “Quality,” and SWEBOK knowledge areas “Software Engineering Management,” “Software Quality,” and “Software Testing” (Mahdi et al. 2021). The evidence synthesised in this review is therefore organised around those four domains. Effort estimation aligns with PMBOK Scope/Resource Management and SWEBOK Cost Estimation; scheduling & resource allocation corresponds to PMBOK Schedule Management and ISO 21500 Time Management; risk management mirrors both PMBOK and PRINCE2 “Risk” themes; and quality assurance & testing maps directly to PMBOK Quality Management and the SWEBOK Software Testing and Software Quality areas (Leone Young 2024).

3.2 Systematic review protocol

The systematic literature review was conducted following the PRISMA (preferred reporting items for systematic reviews and meta-analyses) framework to describe a standardized approach to finding, assessing, and synthesizing research evidence. The PRISMA framework provides a structured overview of what the reviewer did so that future researchers can replicate this (Page et al. 2021).

The main goal of this SLR was to answer major research questions about the effectiveness of AI/ML techniques in SPM, challenges they encounter when such techniques are applied, and directions for future research in this area. To focus on the latest advancements and trends of AI/ML, the 2019–2024 time period has been selected which is relevant since the pace of development and adoption of AI/ML technologies has accelerated in recent years.

More appropriate than other methodologies to the present study as it allows the aggregation of findings obtained from different studies with an elimination of subjective bias. A structured protocol is used to identify gaps in the literature, consolidate existing knowledge, and provide direction for future research. The employment of PRISMA ensured a logical and systematic approach to study identification, selection, and synthesis (Sauer and Seuring 2023).

3.3 Data sources and search strategy

Using multiple well-established academic databases to retrieve the relevant studies was done to ensure the comprehensiveness of the review. The databases used for this work were IEEE Xplore, ACM Digital Library, Scopus, Web of Science, and SpringerLink, which were selected as rich repositories of peer-reviewed research in computer science, software engineering, and relevant disciplines (Cavacini 2014).

An overall search strategy was formulated to comprise a diverse set of studies focused on AI/ML applications to SPM. To ensure broad enough search strings to include compellingly assorted studies without delving into unconnected returns, the writers drew the strings to be as specific as possible. The following examples illustrate the approach:

"Artificial Intelligence" OR "AI" AND "Software Project Management" OR "SPM"

"Machine Learning" OR "ML" AND "Effort Estimation" OR "Risk Management"

Boolean operators, such as AND and OR, were employed to combine keywords effectively, ensuring that all relevant studies were retrieved. To account for differences in terminology in the literature, synonyms for key terms were included. For instance, when searching for papers dealing explicitly with SPM, the term applied, 'Software project scheduling,' was used for papers that treat this aspect of SPM but do not specifically use the term 'SPM.'

Specific filters were applied to further refine the search. To limit the time frame of publications to recent advances, these include restricting the publication time frame to 2018–2024. As this is the predominant language for academic research in the field only articles written int English were included. Only peer-reviewed journal articles and conference papers were included and prioritized for their methodological rigor, editorials, grey literature, and non-peer-reviewed sources were excluded.

To guarantee reproducibility we have documented the exact Boolean queries, field tags, date of final search (12 February 2024), and initial hit counts for each database:

IEEE Xplore (n = 62)

(((("software project management" OR "software-project scheduling") NEAR/5 (artificial intelligence OR machine learning)))

AND (Publication_Year:2019-2024) AND (Document_Type:Conference OR Journal)

ACM DL (n = 49)

title.abs.keywords:("software project management" OR "SPM")

AND title.abs.keywords:("artificial intelligence" OR "machine learning")

AND pub_year:[2019 TO 2024]

Scopus (n = 78):

TITLE-ABS-KEY("software project management" OR "software project scheduling")

AND ("artificial intelligence" OR "machine learning")

AND PUBYEAR>2018

AND (LIMIT-TO(DOCTYPE,"ar") OR LIMIT-TO(DOCTYPE,"cp"))

Web of Science (n = 31)

TS=("software project management" OR "software project scheduling")

AND TS=("artificial intelligence" OR "machine learning")

AND PY=(2019-2024)

Refined by: DOCUMENT TYPES = (Article OR Proceeding Paper)

SpringerLink (n = 17)

("software project management" OR "software project scheduling")
AND ("artificial intelligence" OR "machine learning")
Filter: Year = 2019-2024; Content Type = Conference Paper OR Journal Article

Duplicates (identified via title/DOI matching in Covidence) were removed before screening, leaving 237 unique records. To avoid a superficial analysis, these studies spanned a broad set of AI/ML techniques applied to various aspects of SPM.

We have also used Snowballing techniques (both backward and forward) to enhance the comprehensiveness of literature retrieval and mitigate potential biases from keyword-based searches.

- **Backward Snowballing**: This method involves reviewing the reference lists of key articles to identify earlier studies they cited, which may have been missed during the initial keyword-based search. This approach is valuable for uncovering relevant studies that could otherwise be overlooked, ensuring that the literature review is more exhaustive.
- **Forward Snowballing**: In this technique, we looked at newer studies that cite the key articles (using tools such as Google Scholar or Scopus). This method helps to identify more recent developments and emerging research, ensuring that the literature search remains up to date.

3.4 Study selection

The selection process was designed to filter the initial pool of 237 papers and identify the most relevant and high-quality studies for inclusion in the review. This process was conducted in two stages: the application of inclusion and exclusion criteria, followed by a detailed screening process.

3.4.1 Inclusion criteria

To ensure that every study directly informs our research questions and can be meaningfully compared, we applied the following eligibility rules:

- Peer-reviewed journal article or conference paper (English).
- Published 2018 – 2024.
- Empirical study that implements or evaluates an AI or ML technique.
- Technique applied to at least one of the four SPM domains defined in 3.1.
- Reports quantitative or qualitative outcomes relevant to project-management effectiveness (e.g., estimation accuracy, schedule gain, risk-detection rate, defect-detection precision).

In addition, to further clarify the scope and enhance the coherence of our inclusion criteria, we have explicitly mapped them to the PICOC framework (Population, Intervention, Comparison, Outcome, and Context) as follows (Mengist et al. 2020):

- **Population**: Studies focusing on software project management (SPM) across various industries, specifically those that apply AI/ML techniques in the software development process.

- **Intervention**: Studies that employ AI/ML techniques, such as machine learning algorithms (e.g., regression models, neural networks) and optimization methods (e.g., genetic algorithms, particle swarm optimization).
- **Comparison**: Studies that compare AI/ML methods with traditional project management approaches (e.g., expert judgment, function point analysis).
- **Outcome**: Studies reporting quantitative or qualitative outcomes related to project management effectiveness, such as improved estimation accuracy, better risk detection, and more efficient resource allocation.
- **Context**: Studies related to various types of software projects where AI/ML techniques are applied across phases such as scheduling, effort estimation, resource allocation, risk management, and quality assurance.

3.4.2 Exclusion criteria

Conversely, the following attributes led to automatic exclusion because such papers cannot contribute reliable evidence to our synthesis:

- Grey literature, editorials, book chapters, theses, patents.
- Studies focused solely on general software engineering with no project-management angle.
- Pure simulation or theory papers with no implementation or evaluation.
- Non-English publications or those outside the 2018–2024 window.
- Duplicates or extended versions of work already captured (the most complete version retained).

After these criteria were applied, the selection process was followed by the initial screening of titles and abstracts. After a preliminary review of this initial pool of 237 papers, the authors retained 160 papers. The full-text review process then excluded 110 further papers, which did not meet the inclusion criteria. Finally, we had a final dataset of 50 high-quality studies that directly answered the research questions and gave a great insight into the use of AI/ML in SPM (Fig. 1).

3.5 Screening process

Two independent reviewers (Author UA and Author MN) screened all records in two passes.

Phase 1 – Title/Abstract screening

- We examined the 237 de-duplicated records in Covidence and classified each as *Include*, *Exclude*, or *Uncertain* against the bullet-point criteria in 3.4.
- Agreement was measured with Cohen's $\varkappa$. Raw concordance was 218/237 (92 %); $\varkappa$ = 0.88, indicating "almost perfect" agreement (McHugh 2012).
- Conflicts (n = 19) were resolved through discussion and consensus between the two reviewers. After this phase, 160 records advanced to full-text review.**Phase 2 – Full-text screening**

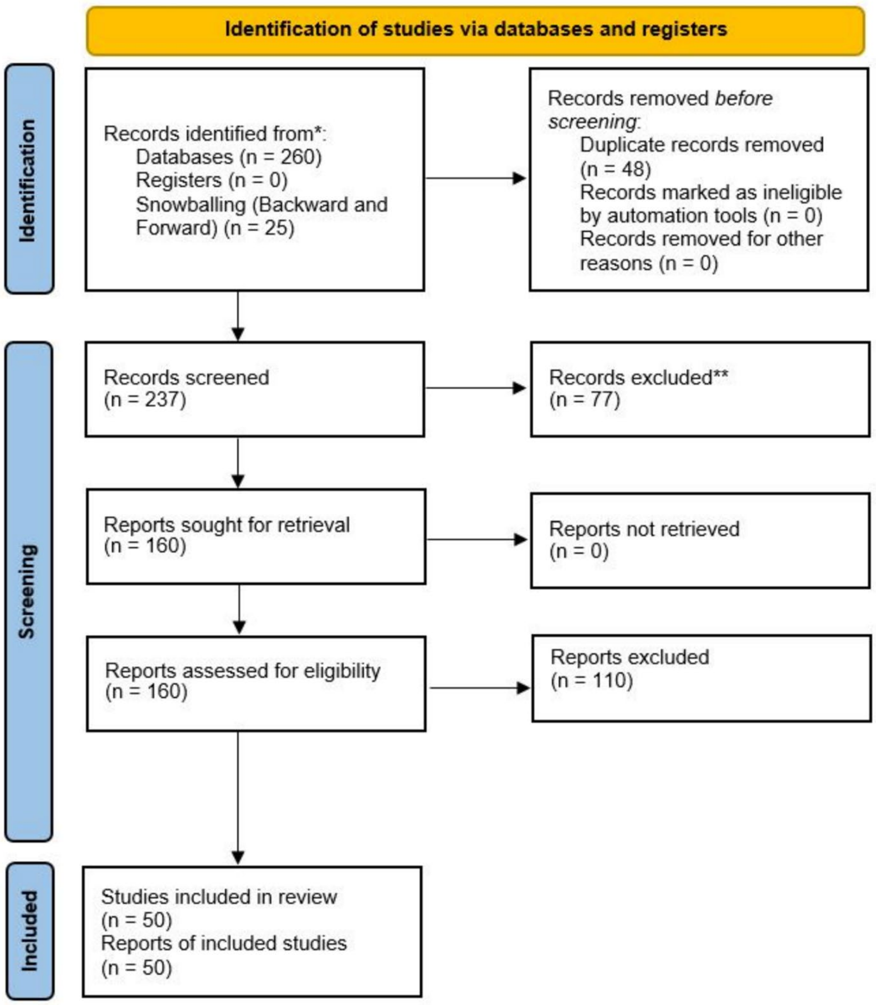
Identification of studies via databases and registers
Identification
Records identified from*:
Databases (n = 260)
Registers (n = 0)
Snowballing (Backward and Forward) (n = 25)
Records removed *before screening*:
Duplicate records removed (n = 48)
Records marked as ineligible by automation tools (n = 0)
Records removed for other reasons (n = 0)
Screening
Records screened (n = 237)
Records excluded** (n = 77)
Reports sought for retrieval (n = 160)
Reports not retrieved (n = 0)
Reports assessed for eligibility (n = 160)
Reports excluded (n = 110)
Included
Studies included in review (n = 50)
Reports of included studies (n = 50)

Fig. 1 Study Selection Framework PRISMA 2020 Flowchart (Source: McHugh (2021))

- We independently assessed the 160 PDFs.
- Agreement: 137/160 concordant decisions (86 %); $\varkappa = 0.83$ ("almost perfect").
- Sixteen disagreements were discussed and resolved unanimously; seven borderline papers were further reviewed and adjudicated by both reviewers to reach a final decision.
- Fifty papers satisfied *all* eligibility rules and were retained for quality appraisal (3.6).

3.6 Quality assessment

Each of the 50 retained studies underwent a structured appraisal designed to gauge methodological soundness and practical usefulness. Two reviewers (UA and MN), after a brief calibration exercise on five pilot papers, **independently** scored every study against four dimensions [35]:

1. **Relevance to the research questions** – explicit application of AI/ML to at least one SPM domain.
2. **Methodological rigour** – transparency of data collection, appropriateness of AI/ML technique, and validity of evaluation measures.
3. **Clarity of findings** – completeness of results reporting, logic of interpretation, and linkage between evidence and conclusions.
4. **Practical applicability** – extent to which outcomes can be generalised to real-world SPM settings.

Each dimension was rated **0 = low, 1 = medium, 2 = high** (maximum composite score = 8). Raw concordance between reviewers was 43/50 (86 %); **Cohen's** $\varkappa$ **= 0.79**, indicating *substantial* agreement. Seven discrepant ratings were resolved through discussion, with no need for third-party adjudication. Consistent with our protocol, only studies scoring greater than or equal to 4 (medium or high quality) were taken forward to synthesis.

3.7 Extraction of data

A standardised extraction template (Excel, pre-piloted on three papers) was used to capture the following fields for each eligible study:

- bibliographic details (authors, year, venue),
- study context (industry/academic, country, project scale),
- AI/ML technique(s) evaluated,
- SPM phase(s) targeted (per 3.1 framework),
- datasets or tools employed, outcome metrics (e.g., MRE, precision-recall, schedule gain), and
- authors' key conclusions and limitations [36].**Dual extraction and reliability.** Two reviewers (UA and MN) independently completed the template for all 50 studies. Concordance on categorical items (technique, SPM phase, venue) was **Cohen's** $\varkappa$ **= 0.82**; agreement on continuous outcomes (accuracy, effort-error) yielded **ICC [2,1] = 0.87**–both indicating strong reliability. Conflicts (5.6 % of cells) were discussed and reconciled by consensus; an adjudicator was not required. The final used studies, along with the data extraction template, study selection log, and other relevant artifacts, are now stored in the Zenodo repository (DOI: doi.org/10.5281/zenodo.17253628).

Synthesis approach. Because the studies report heterogeneous designs and metrics, a qualitative narrative synthesis was undertaken. Extracted data were first thematically grouped by AI/ML technique (e.g., tree-based learners, neural networks, optimisation heuristics) within each of the four SPM domains. We then compared:

1. reported effectiveness (accuracy gains, risk-detection uplift, etc.),
2. contextual moderators (data quality, project size, domain), and
3. stated implementation challenges.

Recurring patterns–such as the superior MRE of ensemble models for effort estimation or the data-quality bottleneck in risk-prediction studies–were coded, tabulated and summarised.

The synthesis also highlighted gaps in the literature and opportunities for future research. For example, there were some studies related to effort estimation and scheduling, and on the other side, fewer studies that were related to risk management and quality assurance to be explored in the future. These findings are the result of an extensive analysis of the potential that AI/ML has to revolutionize SPM phases and serve as a basis for the subsequent chapters in this paper.

3.8 Limitations and threats to validity

Despite the structured protocol and dual-reviewer safeguards adopted in this study, several residual factors should be borne in mind when interpreting the findings:

- We queried five major English-language databases; relevant studies published in other languages or stored in smaller repositories may have been overlooked.
- Limiting the corpus to peer-reviewed journal and conference papers can tilt the evidence base toward positive or novel results and under-represent null findings.
- Although two authors independently screened, appraised, and extracted all records–*dual-reviewer screening ($\varkappa$ = 0.88) partially mitigates selection bias*–some judgment error is unavoidable.
- Many primary investigations rely on small academic or synthetic datasets; results may not translate directly to large, industry-scale software projects.
- The 2019–2024 window captures the most recent AI/ML surge but excludes earlier foundational work and may date quickly as new large-language-model tooling emerges.

Taken together, these limitations frame the review as a rigorous yet time-bounded snapshot of current evidence. Future updates–ideally incorporating additional languages, grey literature, and large industrial case studies–will help to consolidate and extend the insights presented here.

4 Results and analysis

This chapter presents the key findings from the reviewed studies on the application of Artificial Intelligence (AI) and Machine Learning (ML) in Software Project Management (SPM). The studies show many AI/ML techniques that have been used in each phase of software project management–in effort estimation, scheduling, resource allocation, risk management, and quality assurance. These technologies have been shown to enhance accuracy, efficiency, and decision making; for the better of project outcomes. Yet, the adoption of these carries along challenges regarding data quality, model interpretability, provisioning these in the current systems, and scaling explained (Shah 2019).

This section is structured so that it first provides a description of the most common AI/ML techniques used in SPM that can be grouped by type, e.g., machine learning algorithm, optimization technique, expert system, and natural language processing (NLP). It will then delve into the effectiveness of those approaches in key SPM phases, backed up by case studies and examples of how the use of AI and ML is revolutionizing conventional approaches to project management. Challenges associated with each phase of SPM will be discussed along with how AI/ML can address those challenges, as well as the obstacles organizations have in adopting AI/ML to address those use cases.

Finally, a synthesis of the major findings, emerging trends, and areas for further research that will explore the future directions of AI/ML integration in SPM is presented in the chapter. This is an attempt to give a complete understanding of what current status of AI/ML stands in software project management and where we are lagging in enabling their full potential.

4.1 Overview of AI and ML techniques applied in SPM

Artificial Intelligence (AI) and Machine Learning (ML) techniques have been applied to Software Project Management (SPM) to improve planning and scheduling, risk management, as well as performance assurance over the last couple of years. These techniques can enhance project outcomes by providing better predictions, allotment of resources in a better way, and making better decisions (Hossain et al. 2024). The AI and ML techniques commonly applied in SPM can be broadly categorized into the following groups: machine learning algorithms, optimization techniques, expert systems and decision support systems, natural language processing (NLP), and hybrid models (Rane et al. 2024).

4.1.1 Machine learning algorithms

Many AI applications in SPM are backed by machine learning algorithms, like in the fields of effort estimation, scheduling, and risk management. Important ML techniques used are regression models, decision trees, Support Vector Machines (SVM), random forests, and neural networks. Specifically, regression models and decision trees were extensively leveraged for effort estimation by estimating the duration of projects and resource requirements based on historical data (Alfaifi and Aksoy 2023). In scheduling and risk prediction, support vector machines and random forests are commonly used to analyze large datasets, finding patterns to help decision-making in projects (Pospieszny et al. 2018).

In recent years, neural networks have proven effective at enhancing the accuracy of effort estimation models and revealing critical risks in large-scale projects [39].

The estimation errors were significantly reduced and resource allocation improved by the use of these algorithms [42]. In the study conducted by Nabeel (2024), he found that ML algorithms will increase project planning accuracy and reduce the overallocation and underutilization of resources (Nabeel 2024).

4.1.2 Optimization techniques

SPM applies many of the optimization techniques to schedule and allocate resources for solving complex scheduling and resource allocation problems. Genetic algorithms, particle swarm optimization, ant colony optimization, and reinforcement learning are among popular optimization methods (Chen and Zhang 2013). Various constraint-driven scheduling and resource allocation problems, for example considering team availability, skill level, and project timeline, are often solved by applying genetic algorithms and particle synchronization optimization (Nakra 2023). By exploring large solution spaces and discovering approximate (near-optimal) solutions rather than traditional techniques, these techniques have been shown to yield more efficient solutions (Nabeel 2024).

With the widespread use of reinforcement learning, which is based on the concept that an agent learns optimal actions through trial and error, reinforcement learning has been increasingly used to optimize resource allocation in dynamic environments where project requirements change over time (Koulinas et al. 2018). These methods have been shown to produce effective resource allocation strategies in response to evolving project needs and improve decision-making by continuously improving the strategy (Jia et al. 2022).

4.1.3 Expert systems and decision support systems

In applicable, expert systems and decision support systems (DSS) are deemed vital in the SPM because they offer engineering project managers recommendations or insights based on certain rules or expert knowledge. Uncertainty and decision-making in uncertain, changing, or complex project environments are usually solved by fuzzy logic and rule-based systems (Taboada et al. 2023). These systems help make good decisions in situations in which traditional decision-making techniques can fail, due to problems with incomplete knowledge or uncertainty (Taherdoost and Mitra Madanchian 2023).

Risk management and project planning are particular examples where decisions rely on historical knowledge and experience and expert systems are very helpful in these cases (Shamim 2024). The incorporation of expert knowledge into decision-making processes through AI-powered expert systems can drastically increase project efficiency and decrease risk (Taherdoost and Madanchian 2023).

4.1.4 Natural Language Processing (NLP)

Another AI technique, which is more and more being used in the SPM, is Natural Language Processing (NLP), particularly for NLP tasks used on text data, e.g. project status reports, bug tracking, and communication analysis. It follows that NLP techniques including text mining and sentiment analysis are used to derive insights from unstructured project data making it possible for project managers to gain real-time project status and identify potential issues in good time . For example, NLP algorithms can analyze communication patterns so that the company can predict what may be delayed or when the team no longer is satisfied, which is important for the success of the project (Wirtz et al. 2020).

Moreover, NLP helps automate administrative work, including management of project documentation and coming up with usual communication, thus providing project managers with more time to occupy themselves with other strategic activities (Di Giuda et al. 2019). Improvements in both productivity and monitoring of projects have been achieved in several case studies as a result of this capability (Ajiga et al. 2024).

4.1.5 Hybrid models

Combining multiple AI/ML techniques improves the performance of project management systems termed hybrid models. Models of these types are normally a combination of machine learning algorithms with optimization techniques or decision support systems to provide results of better robustness across several phases of SPM (Taboada et al. 2023). For example, hybrid methods in which neural networks are used alongside a genetic algorithm for project scheduling optimization, and a prediction of possible risks (Hourri and Alaa 2021). Integrated systems of such kind can help consider project management as a whole, that is the decisions taken in one phase be in line with the objectives of other phases.

Akumba et al. (2023) implemented a hybrid model was to for estimating software project costs by merging machine learning's prediction power with reinforcement learning's optimization power. Through this integration, it gave much better results (Akumba et al. 2023).

4.1.6 Techniques for specific phases of SPM

The application of AI/ML techniques in SPM varies depending on the phase of the project. ML algorithms, namely regression models and decision trees, in effort estimation, are used for predicting the effort needed for software development tasks based on historical data (Alfaifi and Aksoy 2023). For scheduling and resource allocation, optimization techniques like genetic algorithms, and reinforcement learning are used to make sure resources are allocated efficiently, in cases where projects become dynamic (Zhou et al. 2024). Machine learning algorithms such as decision trees and clustering are used for risk management, through analyzing historical project data to discover and mitigate risks (Secundo et al. 2023). In quality assurance and testing, AI / ML techniques, such as deep learning and anomaly detection, predict defects and test process automation to achieve faster and more accurate testing output (Ramadan et al. 2024).

Overall, we see great promise in including AI and ML techniques across various phases of SPM to improve project efficiency, mitigate risk, and increase project output accuracy. However data quality, model interpretability, and integrating these new technologies with already well-established project management tools still are big hurdles for broad adoption.

4.2 AI/ML in effort estimation

Effort estimation is a crucial task in Software Project Management (SPM), as accurate estimations can significantly influence project planning, resource allocation, and scheduling (Tam et al. 2020). This domain has started to look increasingly to the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques, as they promise to improve estimation accuracy, reduce human bias, and improve project predictability. A large number of studies have investigated the application of different AI and ML methods including regression models, support vector machine (SVM), decision trees, and neural networks to predict software development efforts. In this section, key findings from the literature on applying AI/ML in effort estimation are summarized, their effectiveness is evaluated, real-world case studies are presented, and challenges inherent in the techniques are discussed.

4.2.1 Key findings

AI and ML algorithms have been successfully applied in effort estimation to predict the amount of time, cost, and resources required to complete a project. Linear regression and support vector regression are well-known as regression-based models as they can model the relationship between the input variables (e.g., size, complexity, team experience) and output variables (e.g., effort, duration) (Bargam et al. 2024). For instance, Raju et. al (2024) used Support Vector Machines (SVM) to estimate software project effort and showed that SVM achieved better accuracy and stability than commonly employed estimation techniques like expert judgment and function point analysis (Raju et al. 2024). Several studies have used regression models to predict effort with varying degrees of success based on the quality of input data used and the particular feature selection included (Alfaifi and Aksoy 2023).

Another popular ML technique that seems to work for effort estimation is decision trees. These models function by separating information into subsets depending on particular parameters, offering a hierarchical structure that is easy to decipher and apply (Pospieszny et al. 2018). Decision trees were found to provide better scalability and adaptability compared to traditional expert-based approaches, and a good fit for software projects that are large-scale and complex. With machine learning-based decision trees, managers can make more accurate effort predictions based on historical project data, rather than basing them off of human error and subjectivity (Mahdi et al. 2021).

It is shown that neural networks, which can learn complex patterns from large datasets with better accuracy, are even more promising to improve the estimation accuracy. Effort estimation models can incorporate multiple, crosscutting factors that influence project effort using multilayered neural networks for producing precise and reliable predictions (Şengüneş and Öztürk 2023). The study by Sengunes and Ozturk (2023) has demonstrated that these techniques work best when we have a wealth of historical project data at our disposal, which is only beneficial for organizations that have accumulated a large body of project records. Rijwani and Jain (2022) applied neural networks to predict efforts in software development projects and that has shown better accuracy than the traditional estimation methods and is a more reliable technique for effort prediction (Rijwani and Jain 2022).

Coding of all effort-estimation papers shows a clear chronology: early-period studies (2019–2020) rely almost exclusively on single-algorithm approaches–typically regression or plain SVM–whereas the more recent papers (2022–2024) overwhelmingly adopt either ensemble trees or hybrid set-ups that blend neural networks with optimization heuristics. This shift is not confined to one or two examples: virtually every post-2021 study in the corpus reports combining at least two techniques or coupling the learner with feature-selection or boosting, confirming that "hybridization" has become the dominant design choice for effort prediction in the contemporary literature.

4.2.2 Effectiveness of AI/ML in improving estimation accuracy

The primary benefit of AI and ML in effort estimation lies in their ability to improve the accuracy and reliability of predictions. Standard techniques for effort estimation, such as expert judgments, analogy-based methods, and function point analysis, can suffer from human error and are only as good as the available data quality (Ritu and Bhambri 2023). Unlike AI/ML techniques can process huge amounts of historical data and learn from previous projects to make better predictions. Reduced estimation errors are possible, resource planning is improved and project scheduling becomes more effective.

The studies have repeatedly shown that the prediction accuracy of AI/ML techniques is better than deprecated methods. For instance, in the study where Pospieszny (2018) compared the use of ML algorithms, such as SVM and ensemble methods with function point analysis to estimate the effort, ML algorithms proved to be much more accurate compared to function point analysis which is a mostly inaccurate effort estimation technique (Pospieszny et al. 2018). Furthermore, the incorporation of ensemble methods, including random forests and boosting, has been noted to improve prediction performance by merging the outputs of several models to counteract variance and bias (Uddin et al. 2024).

Additionally, ML-based models have been shown to scale better than traditional solutions. On the other hand, methods used in the past rely usually on domain-specific knowledge and require (similarly to the use case of this thesis) a fair amount of manual effort, whereas ML algorithms can handle a large part of the estimation process 'automatically' and thus better scale to large projects. Additionally, these models are more adaptive to changing project requirements and hence are suitable for use in dynamic and complex software development environments (Taye 2023).

4.2.3 Case Studies/Examples

There are strong real-world case studies on how AI and ML can help improve effort estimation in software projects. For example, in Pospieszny et.al (2024) machine learning algorithms were used to propagate the effort required during a software project. Based on the results, ML models including neural networks demonstrated significantly better performance than traditional estimation methods in terms of the prediction of project duration and resource requirements (Pospieszny et al. 2018).

Also, the study by Ajiga et al. (2024) studied support vector machines for effort estimation in high-tech companies. The researchers discovered that SVM models produced a major reduction in estimation errors and provided the possibility of precise resource allocation, thus enhancing project planning and performance (Ajiga et al. 2024). An example of another study is Hossain et al. (2024) who applied machine learning techniques in the construction industry to predict effort and schedules on projects. Finally, the study concluded that machine learning algorithms are generally more reliable than traditional methods for estimating projects, including using decision trees and regression models to provide better outcomes and fewer delays for project success (Hossain et al. 2024).

4.2.4 Comparative insights

Across the effort-estimation literature, studies converge on the idea that **algorithm choice should follow data conditions**. When historical-project data are plentiful and richly featured, multilayer neural networks consistently edge out simpler approaches, as shown by Şengüneş and Öztürk (2023) and Rijwani and Jain (2022). Where datasets are smaller or features more homogeneous, Support Vector Machines and regularised regression remain competitive–Raju et al. (2024) even report SVM outperforming expert judgement and function-point analysis. Decision trees offer an intermediate option: they scale well to complex, large-scale projects and provide interpretable rule sets that managers can inspect.

Methodologically, papers differ in the cost-driver schemes they adopt (function points versus size-and-complexity mixes) and in how much feature engineering precedes modelling. Work that integrates extensive feature-selection or ensemble techniques (e.g., random forests and boosting) generally reports further accuracy gains by reducing variance and bias. Taken together, the evidence suggests starting with lightweight regression or SVM models when data are limited, then progressing to tree ensembles or neural nets as organisational data maturity grows–while also investing in explainable-AI add-ons to counter the "black-box" reservations often noted in practice.

4.2.5 Challenges in AI/ML-based effort estimation

Although promising results have been obtained, several challenges prevent the widespread use of AI and ML for effort estimating. The key issues include the quality and availability of historical project data. To successfully train models, the ML algorithms require large high-quality datasets. Organizations sometimes do not have enough data, or difficulty with their data, for example, incomplete data or inconsistent data (Aldoseri et al. 2023). Data pre-processing and Feature selection are also important for the success of ML models, poor programmation of data can lead to faulty predictions.

Model interpretability and explainability are the other challenges. Therefore, many ML techniques, particularly deep learning models, behave like ‘black boxes’ and project managers are not able to understand how the model arrived at a specific estimate (Bhavsar et al. 2019). The lack of transparency here can be a big barrier to trust and adoption, where if project managers don’t understand the model they’re relying on, they’re reluctant to do so. To solve this, the latest research is dedicated to creating explainable AI (XAI) models that attempt to give more details and interpretability in ML-based decision-making tasks (Patil 2024). Integration with existing project management tools is a huge challenge at last. Legacy systems are often used for project management and integrating an AI/ML model with these systems can be technologically complicated and expensive. AI-based tools must integrate seamlessly with existing workflows to be adopted and used effectively.

Accurate and reliable effort estimation in software is an important performance baseline for software projects. Traditionally, machine learning algorithms–such as regression models, decision trees, support vector machines, or neural networks–have proved more accurate and have better scalability than traditional estimation methods. What’s working so far, however, comes with challenges: data quality, model interpretability, and integration with legacy systems must be overcome to push adoption further. However, since AI and ML technologies are still evolving, their contribution to effort estimation will likely play an even greater role in helping project managers and organizations.

4.3 AI/ML in scheduling and resource allocation

Software Project Management (SPM) includes scheduling and resource allocation, which directly impact the length of a project, the resources used, and the resulting project success. The management of diverse resources (e.g., human, financial, and technological) across multiple tasks in dynamic environments is very challenging. Scheduling and resource allocation tasks have been shown much promise in turning to AI and Machine Learning (ML) techniques for automating and optimizing these tasks (Chiang and Lin 2020). In this section, we discuss how AI/ML has been applied in these areas and their effectiveness, present example case studies, and note the challenges involved.

4.3.1 Key findings

AI and ML implementation in scheduling and resource allocation is mainly concerned with optimal task assignment, conflict-free resource assignment, and project execution optimization. This domain has explored several AI and ML techniques such as genetic algorithm (GAs), particle swarm optimization (PSO), ant colony optimization (ACO), and reinforcement learning (RL). The best alternative is to analyze complex scheduling problems and optimize scheduling based on multi-resource assignments with project constraints, such as deadlines, budget, and limited resource availability.

A large body of literature exists on Genetic algorithms (GAs) which are inspired by the process of natural selection and have been used widely for optimizing project schedules. Specifically these algorithms iteratively evolve solutions by selecting the 'best' performing individuals (i.e., task schedules) based on a fitness function that assesses solution quality. Due to their ability to solve large, complex tasks and success in applying software project management tasks, like task sequencing and resource leveling, GAs have been used in projects (Nakra 2023). Another population-based optimization technique, namely PSO, is used which again mimics the movement of particles in a swarm to reach the optimum solution (Gad 2022). In SPM, PSO has been used to optimize task scheduling in such a way that makespan (total project duration) is minimized and resource allocation is optimized (Anbarkhan and Rakrouki 2023).

Dynamic resource allocation is being tackled with a promising technique: reinforcement learning (RL), especially in projects where resource availability changes as the project progresses and the requirements of the tasks change too. RL agents learn what is the optimal action, but by interacting with the environment, and seeing how their decisions are evaluated. RL applies to SPM and can be applied to dynamically resource allocating taking into account real-time project data like task progress, availability of resources, and dependencies between tasks (Nabeel 2024). Because they are specifically effective in the control of projects subject to unforeseeable or continually changing conditions, these techniques provide greater flexibility and responsiveness in scheduling and allocation of resources.

Across the scheduling subset, classical evolutionary methods (GA, PSO, ACO) remain foundational, but a growing share of publications from 2022 onward embed these heuristics inside composite frameworks–either hybrid GA-PSO schedulers or GA initialisers followed by reinforcement-learning refinements. Reinforcement learning on its own is still rare, yet its adoption curve is steep: while absent in the 2019 material, it appears in multiple 2023–2024 studies and is routinely paired with an optimisation component, underscoring an incremental but systematic move toward hybrid, multi-stage scheduling pipelines.

4.3.2 Effectiveness of AI/ML in scheduling and resource allocation

AI and ML techniques have demonstrated considerable effectiveness in improving scheduling accuracy, reducing delays, and optimizing resource utilization in software projects. These techniques can be superior to traditional scheduling techniques, such as the critical path method (CPM) and the program evaluation and review technique (PERT) when applied to complex projects with many dependencies and scarce resources. Hossain et al. (2024) in their study applied genetic algorithms to optimize software project scheduling showing that genetic algorithms can also be used to find the best project scheduling by minimizing the project's total duration and by improving resource capacity utilization (Hossain et al. 2024).

AI and ML are especially effective in machine learning and artificial intelligence in resource allocation projects that have dynamic constraints over resources. Traditional scheduling methods do not consider real-time changes in resource availability, task priority, and project scope. On the other hand, AI/ML-based approaches especially reinforcement learning, enable project managers to change resource allocation dynamically subject to changes in conditions. Overall project performance is seen to be improved by minimizing downtime, and by making resources available only when necessary (Shamim 2024). In addition, AI-based models can rapidly detect resource bottlenecks and redistribute resources to prevent delay, thereby shortening project timelines and decreasing costs (Hossain et al. 2024).

4.3.3 Case Studies/Examples

It has also been shown, through case studies, that AI and ML can be used to solve scheduling and resource allocation problems. For example, the case study conducted by Shao et al. (2023), used a hybrid AI approach of genetic algorithms and particle swarm optimization to optimize project scheduling in a software development company. Based on the results, there was a reduction in the project duration and resource utilization. AI-based techniques allowed the company to deal with complex task dependency more efficiently by ensuring that critical tasks were done on time while maximizing resource resource usage (Shao et al. 2023).

Nevertheless, it is another case study by Liu et al. (2018) that explored the application of reinforcement learning for dynamic resource allocation in an IT firm. The authors showed that the reinforcement learning model was able to update the resource allocations in real-time based on project requirements that changed over time, for example, a team member unexpectedly leaving or a new task becomes available. This dynamic scheduling approach enhances the firm's ability to meet project deadlines with reduced resource conflicts which has led to better project outcomes (Liu et al. 2018).

4.3.4 Comparative Insights

The scheduling corpus shows clear divisions along algorithmic lines. Genetic Algorithms and Particle Swarm Optimisation dominate initial project-schedule generation, particularly for problems where task dependencies and resource constraints are well known in advance. Reinforcement learning is favoured for environments that evolve during execution; Liu et al. (2018) demonstrate how an RL agent can reallocate staff dynamically when tasks or personnel availability change unexpectedly. Hybrid strategies that combine GA with PSO, or optimisation with rule-based overlays, appear when authors seek both initial schedule quality and ongoing adaptability.

Data-quality issues and computational load emerge as shared constraints. Scheduling models trained on incomplete task-dependency information or inaccurate resource-availability data can propagate errors that no optimiser can overcome. Likewise, optimisation techniques grow more resource-hungry as project size increases, posing adoption barriers for smaller firms. These methodological contrasts indicate a pragmatic two-step approach: apply GA/PSO for baseline scheduling, then layer a lighter RL or rules engine for day-to-day adjustments, balancing solution quality with computational practicality.

4.3.5 Challenges in AI/ML-based scheduling and resource allocation

Despite the promising results, the application of AI and ML in scheduling and resource allocation presents several challenges. The main obstacle is the computational complexity of the optimization algorithm, especially when such problems are highly complex and large. However, genetic algorithms, particle swarm optimization, and reinforcement learning utilize large resources since a large dataset is required to process big datasets to get an optimum solution (Wang et al. 2022). Small organizations or projects with little computational capacity can make this especially hard.

Another problem is that we need high-quality input data. Historical project data is what AI and ML models are fairly dependent on for training and learning. Incomplete or inaccurate data can result in suboptimal solutions to the scheduling and resource allocation models (N. 2024). For example, wrong information regarding resource availability, and task dependency could result in poor scheduling decisions that afterward can impact project timeline and resource utilization.

Another challenge is integrating with existing project management tools. Combining AI/ML models with current tools usually needs significant changes to the project management software, which could be expensive and lengthy. Furthermore, the use of AI-based scheduling and resource allocation models necessitates the availability of individuals with knowledge and aptitude of how to comprehend, interpret, and make the best use of such models' outputs which creates yet another level of complexity when implementing these models (Mohammad and Chirchir 2024).

Finally, the adaptability of AI and ML models for constraints in a real-world project is another challenge. AI/ML models aim to optimize project schedules, however, real projects are subject to unforeseen changes in project scope, budget constraints, and team performance issues which can aid or spoil the working of the model. Consequently, AI/ML models need to be monitored continuously, and adjusted, so that their solutions remain optimal when conditions are varied (Reddy 2024).

Some of the AI & ML techniques namely, genetic algorithms, particle swarm optimization, and reinforcement learning have evidenced a great assist in improving the scheduling & resource allocation in a software project. This can do better than conventional techniques in task scheduling, shortening project duration while maximizing resource utilization. Challenges including computational complexity, data quality, and incorporation with currently existing systems need to be resolved to realize the full potential of AI/ML for scheduling and resource allocation. With the advancement of AI technologies, the use of AI technologies in software project management will become wider, which will make it possible to manage the biggest software projects even more efficiently and flexibly.

4.4 AI/ML in risk management

Risk management is a critical aspect of Software Project Management (SPM), aiming to identify, assess, and mitigate potential risks that may impact project success. The combination of AI and Machine Learning (ML) techniques has demonstrated capabilities of enhancing risk management by enabling the prediction and detection of risks as well as mitigating them proactively and more effectively. In this section, we review the use of AI/ML in risk management, asses its effectiveness, present some case studies, and discuss the challenges of their use.

4.4.1 Key findings

Risk management in software projects has been widely scoped to apply AI and ML techniques in risk prediction, risk detection, and risk mitigation. Decision trees, clustering algorithms, and even neural networks are often used by machine learning algorithms to predict potential risks by using them to analyze historical project data and discover patterns in historical data that might signal impending risks. Early in the project lifecycle, these models can detect these risks (for example: cost overruns, scope creep, schedule delays) and allow project managers to take corrective action before the risks materialize (Nabeel 2024).

This is the problem that decision trees (widely used in risk prediction) solve by creating a model that shows us what our possible outcomes look like if we take one path or another. In the SPM framework, trees are also used to evaluate the probabilities of risks happening (e.g. delays or budget overruns) from historical data and project characteristics (Hussain et al. 2023).

Another technique that we have applied in risk management is neural networks, which are also a powerful technique in ML. These are models that can learn those complex relationships between different project variables as well as they can find hidden patterns that traditional statistical methods might overlook. Neural networks that predict project risks with significant accuracy given large amounts of datasets of historical project information, for example, are suited for large-scale or high-complexity software projects (Alatawi et al. 2023).

AI-based expert systems are becoming popular in risk management along with machine learning models. Rules-based reasoning in expert systems mimics human expertise in decisions making. Risk assessment tools that utilize these systems have been employed to automate the process of identifying potential risks and recommending mitigation strategies driven by pre-defined rules and historical data (Miah et al. 2021).

The risk-oriented corpus reveals a split between interpretable tree/Bayesian models and deep-learning predictors, but an important nuance is that recent work often merges the two: expert-system or Bayesian front ends generate causal features that are then fed into neural or ensemble classifiers. Almost all papers published after 2021 mention some form of causal-feature engineering or rule-based overlay, indicating that the field is not simply chasing higher accuracy but is converging on hybrid, explainable configurations in response to stakeholder trust requirements.

4.4.2 Effectiveness of AI/ML in risk management

AI and ML techniques have demonstrated considerable effectiveness in improving risk management processes in software projects. AI/ML being used for risk management is also a major advantage as it enables early risk prediction during the project's lifecycle and consequently can be used to develop proactive risk mitigation strategies. Software development is an area where early risk detection is especially important because delays, budget overruns, and scope changes constitute significant threats that can lead to project failure (Schmidt 2023). AI/ML models can only succeed when the historical project data contains exploitable patterns of risk indicators that subtle project managers might miss in real time, but which project managers (or stakeholders) can use to alert them to early warning signals, even if the early warning system is not always reliable.

For example, Antic (2024) used machine learning to predict the possibility of software project cost overruns and schedule delays using machine learning models. These models were able to predict the risks with high accuracy and hence project managers could take pre-emptive measures to address these issues before they progress. Additionally, these AI/ML models shortened the time and cost of manual risk assessment considerably, and the process became more controlled and error-prone to human error (Antić et al. 2023).

Moreover, while AI/ML helps the decision-making process in risk management, it gives actionable insights to project managers. For example, an AI-based expert system can suggest which mitigation approaches for a particular project, are different from others, based on the project's risk profile. Furthermore, it increases the effectiveness of risk mitigation, and the overall project success rate is increased (Nabeel 2024). Furthermore, unlike deterministic models, AI/ML models can automatically revise risk estimates with new data entering the picture, such that risk management approaches remain current throughout the life cycle of a project (Khodabakhshian et al. 2023).

4.4.3 Case studies and examples

Real-world case studies have demonstrated the successful application of AI and ML in risk management. Benldris et al. put forward an improved model for the risk analysis of software development projects that uses BNs with causality constraints (BNCC). They showed that the suggested model, when together with expert knowledge, can identify causal structures that are consistent with the experts' information and that it provides better prediction accuracy than other algorithms such as logistic regression, Naive Bayes, and generic BNs. They introduced the first-ever framework for assessing the risk causality of software projects for risk assessment and a model for managing risk factors in software projects anchored on BNCC theory following their research (BenIdris et al. 2020).

Mahfoodh and Obediat (2020) proposed a new technique for risk estimation that can help the internal stakeholders of software development analyze the current risks of software risks in the expectation of a quantitative value for software risks. To set the tone of the risk it was calculated using historical quantitative data on software bugs and compared with current and forecasted bug-fix times, duplicate bug records, and the priority rank of the software component. Machine learning was used to determine the risk value on a Mozilla Core dataset (Networking: A value for risk level for certain software faults and the HTTP software component assigned, was predicted using the Tensorflow tool. This approach was used in approximating the overall risk to range between 0.274 and 0.84, and maximum prediction accuracy of 0.35. From this study, the researchers found a high correlation between risks estimated from bug-fix time and risks arising from the replication of bug reports (Mahfoodh and Obediat 2020).

A study was conducted by Iftikhar et. al (2021) on risk prediction using ANN. There is an increasing call for software development across the globe due to a lack of qualified software professionals in one geographical region or even country. It has resulted in the current trend of global software development, in which competent and skillful personnel around the world offer their services to complete specific projects. Thus, organizations can deal with IT challenges and develop software components by attracting the best software developers around the world. However, there is the issue of synthesing these globally distributed skills and tools into effective solutions to real-world issues. Risk management is an important field that provides solutions regarding the problems arising before software professionals in this environment of competition. This study has the primary objective of predicting risks concerning time, cost and resources for distributed teams engaged in GSD projects. For this purpose, the Levenberg–Marquardt algorithm, Bayesian Regularization, and Scaled Conjugate Gradient methods of neural networks are applied to forecast the effect of these risks. After demonstrating the steps used to establish the algorithms, a comparison analysis is performed to determine the highest level of accuracy in risk prediction. The findings of the present study indicate that Bayesian Regularization achieved higher accuracy than the other algorithms for MSE validation measures. This suggests that it performs better when predicting risks relating to temporal aspects of projects, cost estimates, and resource usage in contexts of development teams geographically dispersed globally (Iftikhar et al. 2021).

4.4.4 Comparative insights

Results across risk-management studies emphasise that model transparency often trumps raw predictive power. Decision-tree ensembles and expert-system rules are repeatedly valued for the clear rationales they give project managers, even when neural networks achieve slightly higher accuracy on large datasets (Iftikhar et al. 2021). Benldris et al. show that Bayesian networks enhanced with causality constraints strike a workable balance, outperforming baseline classifiers while preserving interpretable causal links (BenIdris et al. 2020).

Contradictory findings mainly stem from domain context. Antić et al. (2023) reports strong success predicting cost overruns in civil-engineering software, whereas Mahfoodh & Obediat (2020) note modest accuracy when forecasting bug-related risks in an open-source environment–differences likely tied to risk-label granularity and data volume (Mahfoodh and Obediat 2020). A cross-cutting lesson is that risk-prediction models are most effective when they incorporate domain-specific variables and remain intelligible to decision-makers, making explainable techniques a recuran research priority.

4.4.5 Challenges in AI/ML-based risk management

Risk management processes in software projects have proved to benefit from AI and ML, which provide them with enhanced capabilities for predicting risks, detecting them, and mitigating them. Decision trees, clustering algorithms, neural networks, and expert systems, all have proven their usefulness in improving risk management outcomes, for example, early detection of risk and better developing of decisions. However such technologies face challenges of data quality issues, model interpretability, and its integration with existing systems to be successfully adopted and used. However, the improvement of AI and ML will be constantly achieved, and their potential in risk management can be seen to be continuously increasing for software project management, which allows for more proactive, efficient, and effective risk management.

4.5 AI/ML in quality assurance and testing

Quality assurance (QA) and software testing are crucial to ensuring that software products meet the required standards and function as intended. Artificial Intelligence (AI) and Machine Learning (ML) can be integrated into QA and testing processes improving defect detection, optimizing test case generation, and automating repetitive tasks, thereby increasing its efficiency and reducing the turn around time. In this section, the use of AI/ML in software quality assurance and testing is explored, on key findings and Efficacy, case studies, and Challenges.

4.5.1 Key findings

AI and ML techniques are increasingly being applied to various aspects of QA and software testing. With these techniques, we can automate manual testing tasks, improve defect prediction, increase test coverage, and support easier generation of test cases, among others. AI is certainly one of the most popular QA automation use cases through automated testing where ML algorithms can be trained to identify defects in the code by using historical data and previous bug reports. It renders unnecessary the need for human testers to manually hunt for errors, making testing faster and more accurate, all at the same time (Islam et al. 2023).

Another area where AI/ML techniques have had a lot of promise is defect prediction. By studying historical project data, an AI model can forecast the chance of flaws arising for certain modules or bits inside the codebase. Commonly used machine learning algorithms, decision trees, and neural networks are applied to classify areas of software areas that are most likely to contain defects based on features including complexity, code churn, and developer experience. Identifying high-risk areas early allows teams to focus testing efforts better leading to a better final product (Pachouly et al. 2022).

Along with defect prediction, test case generation and optimization are also driven by the use of AI and ML. However, traditional test case generation methods are based on predefined scenarios and human intuition, often leading to higher construction costs and a higher probability of errors. However, it is possible to use an AI-powered system to automatically generate test cases based on software specifications, requirements, and user stories without manual effort and to achieve more comprehensive coverage. Additionally, ML models can arrange test cases on their chances of spotting bugs and can also stream line the testing process and minimize unnecessary test runs (Azlaan 2024).

Early QA studies in our dataset favour standalone classifiers optimised by meta-heuristics, yet the latest work demonstrably layers multiple techniques–e.g., graph convolution plus weighted loss, or clustering-assisted feature selection preceding defect classifiers. The upward trend is pronounced in bug-severity and priority prediction, where deep-text or graph models are now routinely augmented with optimisation wrappers or semantic clustering. Taken together, the literature shows a steady transition from single-algorithm defect predictors to multi-component, hybrid test-automation stacks.

4.5.2 Effectiveness of AI/ML in quality assurance and testing

The effectiveness of AI and ML in improving QA and software testing outcomes is well-documented in various studies. One of the primary benefits of these technologies is their ability to improve defect detection rates. Manual inspection and predefined testing scenarios that are traditionally used in testing can fail to pick defects in these subtle places while testing in such large and complex codebases. Although AI/ML Models analyze numerous data and search for patterns that are not easily apparent, they enhance the defect detection process.

Furthermore, it has been proven that AI/ML helps boost the testing workflow. In a traditional testing practice, several test cases having to run across many software modules can take too much time and money. With AI-driven systems, however, this process can be optimized by creating an automated system for test case generation, prioritized testing of tests, and no redundancy is present (Khan et al. 2024).

Another area in which AI/ML has proven effective is speed in testing. They can learn from the example of other software in real time and detect bugs and vulnerabilities during the software development process by analyzing the software in real-time. The continuous testing approach also referred to as continuous integration or continuous testing helps teams to catch bugs early so that they can be fixed before they escalate, hence the release frequency is increased and the software quality improved (Kolawole et al. 2024).

In addition, AI/ML is flexible enough to adapt to changing software environments. When facing new features, ongoing data gathering opportunities can teach AI/ML models the new data, refine their predictions, and also help with test case generation. By being suitable for this kind of adaptability, the testing process will still work even while the software is developed through continuous changes.

4.5.3 Case studies and examples

Real-world case studies demonstrate the practical benefits of AI and ML in QA and software testing. A case study of the use of machine learning models in a large-frame software development project for defect prediction was one such example. In this study, authors analyzed the use of various well-established machine learning (ML) techniques, as well as data-driven approaches to optimizing ML using an openly available data set. The objective of the research is to improve the model performance in terms of accuracy and precision; by citing additional improvements in accuracy in previous studies, limitations were identified within these studies. These improvements were achieved using K-means clustering to group class labels sensibly. Then, selected features were classified using classification models, and Particle Swarm Optimization (PSO) was applied to optimize the performance of ML models. The performance of the model was evaluated using metrics on precision, accuracy, recall, F measure, error rates, and confusion matrix. Findings showed that both standard ML and optimized ML models gave a good performance, where trained Support Vector Machines (SVM) and optimized SVM models were found to perform the best, reaching accuracy as high as 99% and 99.80% respectively. Other models also showed very accurate results, as Naïve Bayes (NB), Optimized NB (OV) achieved 93.90, 93.80, Random Forest (RF), and Optimized RF (OV) got 98.70 & 99.50, and the ensemble methods interestingly turned out with 97.60. In comparison with previous research, these results imply significant progress, accomplishing the goal of maximizing model accuracy (Khalid et al. 2023).

Recent work in bug severity prediction has progressed from the use of common computational approaches and prediction rules to the use of deep learning and natural language processing techniques. Now Ramay et al. (2019) proposed a method harnessing deep neural networks with modeling of bug reports as NLP problems, then scoring to capture the emotional context of the bug reporters (Ramay et al. 2019). Choetkiertikul et al. (2021) propose a predictive model that automatically extracts semantic features from issue reports using long short-term memory networks and combines them with traditional textual similarity features. We evaluate this model against a dataset of 142,205 issues from 11 large projects achieving 66.31% accuracy (Choetkiertikul et al. 2021). A new approach called Bug Priority Prediction using Graph Convolutional Networks (GCN) with a weighted loss as proposed by Fang et al. (2021). However, this system constructs a heterogeneous text graph for bug reports to allow GCN to extract the semantic word relationships and train the model by a weighted loss function. This method was tested on bug priority prediction for four open-source projects: Eclipse, Netbeans, the GNU Compiler Collection, and Mozilla (Fang et al. 2021).

4.5.4 Comparative insights

In QA and testing, deep-learning and graph-based approaches lead for defect prediction and bug-severity classification–as illustrated by Fang et al. (GCN) and Ramay et al. (deep neural networks) (Fang et al. 2021; Ramay et al. 2019). Yet traditional classifiers boosted by meta-heuristics (e.g., PSO-optimised SVMs) still perform strongly when paired with rigorous feature selection, suggesting that optimisation of simpler models can offset the need for heavy deep-learning pipelines. Flexibility and continuous-testing speed are noted benefits of AI-assisted test-case generation, but authors also highlight integration hurdles with existing CI/CD flows.

Methodological differences appear in how tools handle unstructured artefacts such as bug-report text. Studies leveraging natural-language techniques (Choetkiertikul et al., 2021) report gains in semantic insight but require substantial preprocessing (Choetkiertikul et al. 2021). Conversely, approaches centred on code metrics alone integrate more smoothly but risk missing context-rich defect patterns. Collectively, the evidence indicates that teams should match AI technique to artefact type and pipeline constraints–deploying sophisticated models where text or graph context is available, and leaner optimised classifiers where integration speed is paramount.

4.5.5 Challenges in AI/ML-based quality assurance and testing

The quality assurance and software testing practices are getting hugely improved with the help of AI, and ML technologies due to better defect detection, redundant task automation and optimized test case generation. These are technologies that have proved to efficiently achieve these goals: they improve the efficiency of testing, reduce defect rates, and have dynamic flexible nature to respond to changes in the software environment. Though however AI/ML has potential in QA and testing, there are still some challenges including the one of data quality, model interpretability and this is much integrating with legacy systems. Nonetheless, AI solutions retain potential to provide meaningful ways of shortening testing cycle, enhancing software quality and driving down time to market.

4.6 Summary of key findings and trends

In this section, we summarize the key findings from the reviewed studies on AI and ML applications in Software Project Management (SPM). We organize the studies into tables, categorizing them based on their focus areas, AI/ML techniques, effectiveness, challenges, and examples from the research papers. This overview allows for a clearer understanding of the trends and gaps in the current literature, highlighting the impact of AI and ML in various phases of SPM.

The synthesis draws on all 50 studies retained after quality appraisal (3.6). To keep the main text concise each domain-specific table (Tables 1, 2, 3 and 4) shows only those studies that

1. reported at least one quantitative outcome for the domain in focus.

Studies whose findings span several domains are discussed narratively in Sections 4.2–4.5. This approach preserves readability while ensuring that every included study is represented either tabularly or textually.

5 Recommendations and future work

In this section, we provide recommendations based on the findings from the previous chapter and suggest directions for future research. The objective of these recommendations is to resolve challenges hindered by the adoption of Artificial Intelligence (AI) and Machine Learning (ML) in Software Project Management (SPM) and to encourage effective incorporation of these technologies in project management practices. This proposed future work is aimed at helping researchers, practitioners, and organizations make better use of AI and ML techniques applied in the area of SPM.

Table 1 Key Findings from the Research Papers

Study	Focus Area	AI/ML Techniques	Key Findings	Challenges
Nabeel (2024)	Resource Allocation, Risk Mitigation, Decision-Making	Machine learning, AI enhanced systems, big data analysis	Better use of resources, risk mitigation, and AI-enabled decision-making across multiple industries.	Data quality, model interpretability, tool integration.
Taboada et al. (2023)	Emerging AI in Project Management	Various AI techniques, including decision trees and optimization algorithms	Key gaps in AI adoption, for example, low adoption of deep learning and lack of standardization.	Data security, sustainability, AI adoption barriers.
Uddin et al. (2024)	Project Analytics, Performance	SVM, Random Forest, Neural Networks	Focus on ML/DL algorithms in performance related to cost, time and safety and risk management.	Lack of robust datasets for training models.
Pospieszny et al (2018)	Effort and Duration Estimation	SVM, Neural Networks, Ensemble Learning	Traditional methods of estimating project effort and duration are outperformed by AI models.	Outlier data and feature selection challenges.
Bhavsar et al. (2019)	Automation, Risk Management	AI, ML, Deep Learning, NLP, Speech Recognition	Focus on AI-driven BPR for automating software project management tasks.	AI integration in existing workflows.
Nakra et al. (2023)	Task Allocation, Resource Optimization	Reinforcement Learning, Bayesian Networks, NLP	AI models help to predict best resource distribution and task management.	High data requirements, computational complexity.
Ajiga et al. (2024)	Code Generation, Bug Detection, Testing	Tools for the usage of AI, ML models for bug tracking, testing, CI/CD pipeline.	Software development productivity, quality, and efficiency was significantly improved by AI tools.	A high initial investment, complex models.
Hossain et al. (2024)	Project Efficiency, Planning	Scheduling, resource allocation, risk management with help of AI tools	AI integration resulted in better scheduling of the projects, lesser time to production, and better resource utilization.	Resistance to change, integration barriers.

5.1 Emerging trends and research gaps

Based on the findings summarized in the tables, several trends and gaps in the research on AI and ML in Software Project Management (SPM) emerge. One key trend is the increasing adoption of hybrid models that combine machine learning with optimization techniques for cross-phase applications (e.g., effort estimation, scheduling, and resource allocation). These models have shown significant promise in improving overall project efficiency. However, data quality and model interpretability remain persistent challenges, limiting the scalability and practical adoption of AI/ML in real-world SPM contexts (Azevedo et al. 2024). A closely related–and increasingly urgent–issue concerns legacy-system compatibility. Widely used platforms such as MS Project, Jira, and Primavera store schedules, costs, and risk logs in proprietary formats with limited real-time APIs, creating a practical barrier to embedding AI modules. Studies that attempt in-situ deployment report significant ad-hoc effort: bespoke scripts to convert CSV/XML exports, batch processing outside the

Table 2 AI/ML Techniques and Their Application Areas in SPM

AI/ML Technique	Application Areas	Impact on SPM	Studies Cited
Machine Learning Algorithms	Effort estimation, scheduling, risk management, resource allocation	Improved prediction accuracy, reduced errors, optimized decisions.	Taboada et al. (2023), Hossain et al. (2024)
Optimization Techniques (GA, PSO, ACO)	Scheduling, resource allocation, risk management	Enhanced scheduling efficiency and resource optimization.	Nakra et al. (2023), Bhavsar et al. (2019)
Expert Systems	Decision support, risk management	Improved decision-making with rule-based models and real-time data.	Taboada et al. (2023), Ajiga et al. (2024)
Natural Language Processing (NLP)	Text mining for project status, communication, bug tracking	Streamlined communication, faster bug identification.	Bhavsar et al. (2019), Nakra et al. (2023)
Hybrid Models	Across multiple phases (estimation, scheduling, risk management)	Enhanced accuracy, better coordination, and performance.	Nabeel (2024), Ajiga et al. (2024), Taboada et al. (2023)

Table 3 Effectiveness of AI/ML in Different Phases of SPM

Phase of SPM	AI/ML Techniques Used	Effectiveness	Studies Cited
Effort Estimation	Regression models, SVM, Decision Trees	Improved accuracy and reduced errors in estimation.	Pospieszny (2018), Uddin et al. (2024)
Scheduling and Resource Allocation	Genetic Algorithms, PSO, Reinforcement Learning	Optimized resource distribution and reduced delays.	Nakra et al. (2023), Nabeel (2024)
Risk Management	Decision Trees, Neural Networks, Clustering Algorithms	Early detection of risks, better mitigation strategies.	Bhavsar et al. (2019), Taboada et al. (2023)
Quality Assurance and Testing	Deep Learning, Anomaly Detection, Predictive Models	Enhanced defect detection, reduced testing time, and improved quality.	Ajiga et al. (2024), Nakra et al. (2023)

live environment, and manual re-injection of predictions. Promising remedies are beginning to surface: lightweight REST middleware that converts legacy exports into an open JSON schema; micro-service "sidecars" or plug-ins that surface AI insights directly inside the familiar dashboard; phased "shadow-mode" pilots in which AI

Table 4 Challenges and Gaps in AI/ML Adoption in SPM

Challenge	Impact on SPM	Studies Cited
Data Quality and Availability	Inaccurate predictions as well as reduced model effectiveness are caused by poor data quality.	Uddin et al. (2024), Pospieszny (2018)
Model Interpretability	Lack of ability to comprehend the decision of AI models gives rise to untrustworthy AI model decision-making.	Taboada et al. (2023), Hossain et al. (2024)
Integration with Existing Systems	Some difficulties with integration of AI models into legacy project management tools are observed.	Bhavsar et al. (2019), Ajiga et al. (2024)
Scalability	Scaling AI models for creation of big or more complicated projects.	Nakra et al. (2023), Nabeel (2024)
Resistance to Change	Delaying the benefits in the AI tools due to organizational resistance to adopting them.	Hossain et al. (2024), Nabeel (2024)

recommendations are shown but not yet enforced; and explainable-AI panels that display feature influences to foster user trust. Systematic research into these patterns–coupled with open data-exchange standards–could transform integration from a bespoke hurdle into a repeatable practice, ensuring that emerging AI capabilities can slot smoothly into day-to-day project-management workflows. Additionally, further exploration of deep learning techniques for risk management and quality assurance could offer valuable insights for improving project outcomes in complex software development environments.

It is important to note that these integration solutions, particularly the hybrid models, are often cumulative and applicable across multiple SPM phases. For instance, hybrid models combining machine learning with optimization techniques are increasingly used for tasks spanning effort estimation, scheduling, and resource allocation. These models provide a more robust framework that adapts to multiple project management domains. On the other hand, some integration solutions, such as those for risk management or quality assurance, are more domain-specific. Techniques like deep learning and anomaly detection are typically applied to specific tasks such as defect prediction or risk assessment and are not commonly generalized across other project management phases.

5.2 Overview of AI and ML techniques applied in SPM

Based on the key findings from the literature, the following recommendations are provided to practitioners involved in software project management:

1. Adopt Hybrid Models for Enhanced Accuracy: Several studies pointed out the advantage of combining several AI/ML techniques such as machine learning with optimization algorithms, to produce better project management results. In

this respect, it is important to encourage practitioners to explore hybrid models, which combine the usage of machine learning algorithms (SVM, Neural Networks) with optimization techniques (genetic algorithms, particle swarm optimization) (Azevedo et al. 2024). In doing so, they hold promise to improve effort estimation, scheduling, and resource allocation.

2. Focus on Data Quality and Availability: Data quality remains a major barrier to the successful application of AI/ML in SPM. Improving data collection and management processes is the key that practitioners need to prioritize, to ensure clean, consistent, and relevant data is available for training AI models (Rangineni 2023). The reason is particularly for risk management and project estimation, as poor data quality can result in inaccurate predictions and suboptimal decision-making.
3. Invest in Explainable AI: Given that AI/ML models will become more ubiquitous in decision-making, demand for interpretable and explainable models is important to get project managers' and stakeholders' trust. Adoption of transparency in the AI decision-making process will be key for organizations to use such models, especially for complex tasks such as risk management and scheduling (Bhavsar et al. 2019).
4. Integrate AI into Existing Project Management Systems: To increase the AI/ML adoption in SPM, practitioners should investigate the means to integrate with their existing project management software AI tools. The addition of AI-enhanced features to present systems can automate repetitive tasks, enhance estimation accuracy, and optimize resource allocation. By integrating this, AI will not be seen as a substitute for existing tools but as an improvement to add efficiency and decision-making powers.

5.3 Overview of AI and ML techniques applied in SPM

While significant progress has been made in applying AI and ML to SPM, several opportunities for further research exist:

1. Focus on Deep Learning for Complex Problem-Solving: Deep learning models are underutilized in SPM but machine learning models like decision trees and regression have been extensively explored. Future work can investigate deep learning techniques, like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to solve challenging problems in risk management, effort estimation, and quality assurance (Taye 2023).
2. Address the Gap in Agile Project Management: The majority of the studies discussed in this systematic literature review focused on traditional project management methodologies while few were applied to agile project management and AI/ML have been applied in both. As future work, research should be directed toward how AI/ML can most rigorously be incorporated in agile practices (such as sprint planning, task prioritization, and resource allocation) and how challenges that are particular to agile projects should be faced (Al-Fraihat et al. 2024).
3. Improve Risk Management Models with AI: AI/ML has already proved to be a promising technique to predict risks in software projects, but there is still a long

way to go. Possible future research involves designing more sophisticated models accounting for a broader set of risk factors, e.g., a project's complexity, team behavior, or external environment factors (e.g., market changes). Such models could produce early warning signals of these potential risks and recommend mitigating strategies before damage occurs.

4. Research on AI Integration with Legacy Systems: Integrating AI/ML models with legacy systems in organizations remains a key challenge. The researchers need to look at strategies and frameworks that will overcome technical and organizational barriers to AI adoption, such as the development of middleware and APIs that are easy-to-use applications to connect AI models to older project management systems.

6 Conclusion

The application of Artificial Intelligence (AI) and Machine Learning (ML) in Software Project Management (SPM) has the potential to significantly enhance the efficiency, accuracy, and effectiveness of project management processes. This systematic literature review has reviewed the different AI and ML techniques used in different phases of SPM comprehensively including effort estimation, scheduling, resource allocation, and risk and quality assurance phases. Based on current research, main findings, and potential trends, this paper aims to discuss the current AI/ML adoption in SPM and reveal the important directions to focus on.

Several critical findings emerged from the reviewed literature. First, AI and ML techniques, including machine learning algorithms (e.g., regression models, support vector machines, neural networks), optimization techniques (e.g., genetic algorithms, particle swarm optimization), and expert systems, have been successfully applied across various stages of software project management. It has been proved that these techniques can increase the predictability of effort estimates, enhance the process of resource allocation and scheduling, and help with risk control testing and quality assurance.

In addition, firstly, integrated approaches that utilize several AI/ML mechanisms have demonstrated increased effectiveness of project management processes. These models offer more powerful solutions relying on the capabilities of different algorithms to yield more precise predictions and efficient project implementation. Another factor that has been pointed out as perhaps the primary reason for AI to become entrenched in contemporary project management is if bearing upon the existent tools.

However, some barriers prevent the effective application of AI/ML in SPM even today. Challenges that include data propriety, model explicability, and compatibility with existing frameworks are some of the obstacles interfering with the utilization of AI-fostered solutions. Accelerate, more research is required on applying AI/ML for the agile project management and development of an integrated real-time decision support environment.

This review contributes to the field by consolidating the various AI/ML techniques and their applications in SPM into a comprehensive framework. The division and description of the techniques give the review a better view of the efficacy of the different methods in particular aspects of project management. In addition, it covers recent trends including the growing application of reinforcement learning for resource allocation and the increasing demand for xAI models that are vital for providing transparency in computing decisions.

The study also identifies the researched gaps in the existing literature, including the limited application of AI in agile project management and the increased complexity of large, sophisticated projects. These gaps provide a research agenda for future work to continue to build the theory of the subject matter.

While this review provides valuable insights, it is not without limitations. The present review was restricted to the most recent articles based on several years of publication, and including studies from various industries or PM domains could add more value to the results. Moreover, future works might explore the ethical, organizational, and social consequences of AI/ML in SPM as well as the effects of these technologies on the decision-making remit of project managers.

Furthermore, because AI and ML technologies are actively developing, it is important to consider new developments, e.g. quantum computing or federated learning, which might contain the solutions for the complicated issues of project management.

Author Contributions Conceptualization, U.A. and M.N.; methodology, U.A. ; software, U.A.; validation, M.N.; formal analysis, M.N.; resources, U.A.; data curation, M.N.; writing original draft preparation, U.A.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Ajiga, D., Okeleke, P.A., Folorunsho, S.O., Ezeigweneme, C.: Enhancing software development practices with AI insights in high-tech companies. Computer Science & IT Research Journal. **5**(8), 1897–1919 (2024)

Akumba, B., Ogala, E., Agaji, I., Blamah, N.V.: A Hybrid Machine Learning Method for Estimating Software Project Cost. https://www.researchgate.net/publication/379956201_A_Hybrid_Machine_Learning_Method_for_Estimating_Software_Project_Cost (2023)

Alatawi, M.N., Alyahyan, S., Hussain, S., Alshammari, A., Aldaeej, A.A., Alali, I.K., Alwageed, H.S.: A Data-Driven artificial neural network approach to software project risk assessment. IET Software **2023**, 1–19 (2023)

Aldoseri, A., Al-Khalifa, K.N., Hamouda, A.M.: Re-Thinking data strategy and integration for artificial intelligence: Concepts, opportunities, and challenges. Appl. Sci. **13**(12), 7082 (2023)

Alfaifi, I.J., Aksoy, P.M.S.: Impact of machine learning on IT project management. Journal of Image Processing and Intelligent Remote Sensing. **41**, 31–38 (2023)

Al-Fraihat, D., Sharrab, Y., Al-Ghuwairi, A.-R., Alzabut, H., Beshara, M., Algarni, A.: Utilizing machine learning algorithms for task allocation in distributed agile software development. Heliyon. **10**(21), 39926 (2024)

Anbarkhan, S.H., Rakrouki, M.A.: An enhanced PSO algorithm for scheduling workflow tasks in cloud computing. Electronics **12**(12), 2580 (2023)

Andrei, B.-A., Casu-Pop, A.-C., Gheorghe, S.-C., Boiangiu, C.-A.: A STUDY ON USING WATERFALL AND AGILE METHODS IN SOFTWARE PROJECT MANAGEMENT. https://www.researchgate.net/publication/333968900_A_STUDY_ON_USING_WATERFALL_AND_AGILE_METHODS_IN_SOFTWARE_PROJECT_MANAGEMENT (2019)

Antić, Katarina, S.: Implementing artificial intelligence tools for risk management in software projects. Tehnika. **78**(6), 735–742 (2023)

Azevedo, B.F., Rocha, A.M.A.C., Pereira, A.I.: Hybrid approaches to optimization and machine learning methods: a systematic literature review. Mach. Learn. **113**(7), 4055–4097 (2024)

Azlaan, H.: Leveraging AI and ML to predict defects, prioritize testing efforts, and optimize test coverage in complex.. https://www.researchgate.net/publication/383464151_Leveraging_AI_and_ML_to_predict_defects_prioritize_testing_efforts_and_optimize_test_coverage_in_complex_software_systems (2024)

Barenkamp, M., Rebstadt, J., Thomas, O.: Applications of AI in classical software engineering. AI Perspectives. **2**(1), 1 (2020)

Bargam, B., Boudhar, A., Kinnard, C., Bouamri, H., Nifa, K., Chehbouni, A.: Evaluation of the support vector regression (SVR) and the random forest (RF) models accuracy for streamflow prediction under a data-scarce basin in morocco. Discover Applied Sciences. **6**(6) (2024)

BenIdris, M., Ammar, H., Dzielski, D., Benamer, W.H.: Prioritizing Software Components Risk. ACM (2020)

Bhavsar, K., Shah, D.V., Gopalan, D.S.: Business process reengineering: A scope of automation in software project management using artificial intelligence. International Journal of Engineering and Advanced Technology. **9**(2), 3589–3594 (2019)

Božić, V.: THE ROLE OF ARTIFICIAL INTELLIGENCE IN RISK MANAGEMENT. https://www.researchgate.net/publication/370005124_THE_ROLE_OF_ARTIFICIAL_INTELLIGENCE_IN_RISK_MANAGEMENT (2023)

Cavacini, A.: What is the best database for computer science journal articles? Scientometrics **102**(3), 2059–2071 (2014)

Channe, P.S.: The Impact of AI on Economic Forecasting and Policy-Making: Opportunities and Challenges for Future.. https://www.researchgate.net/publication/382827405_The_Impact_of_AI_on_Economic_Forecasting_and_Policy-Making_Opportunities_and_Challenges_for_Future_Economic_Stability_and_Growth (2024)

Chen, W.-N., Zhang, J.: Ant colony optimization for software project scheduling and staffing with an Event-Based scheduler. IEEE Trans. Software Eng. **39**(1), 1–17 (2013)

Chiang, H.Y., Lin, B.M.T.: A decision model for human resource allocation in project management of software development. IEEE Access. **8**, 38073–38081 (2020)

Chirra, S.M.R., Reza, H.: A survey on software cost estimation techniques. J. Softw. Eng. Appl. **12**(06), 226–248 (2019)

Choetkiertikul, M., Dam, H.K., Tran, T., Pham, T., Ragkhitwetsagul, C., Ghose, A.: Automatically recommending components for issue reports using deep learning. Empir. Softw. Eng. **26**(2), 14 (2021)

Collins, C., Dennehy, D., Conboy, K., Mikalef, P.: Artificial intelligence in information systems research: A systematic literature review and research agenda. Int. J. Inf. Manage. **60**, 102383 (2021)

Conboy, K., Carroll, N.: Implementing Large-Scale agile frameworks: Challenges and recommendations. IEEE Softw. **36**(2), 44–50 (2019)

Di Giuda, G.M., Locatelli, M., Schievano, M., Pellegrini, L., Pattini, G., Giana, P.E., Seghezzi, E.: Natural language processing for information and project management. In: Research for Development, pp. 95–102. Springer, Cham (2019)

El Bajta, M., Idri, A., Ros, J.N., Fernandez-Aleman, J.L., Gea, J.M., Garcia, F., Toval, A.: Software project management approaches for global software development: a systematic mapping study. Tsinghua Science and Technology. **23**(6), 690–714 (2018)

Fang, S., Tan, Y.-S., Zhang, T., Xu, Z., Liu, H.: Effective prediction of Bug-Fixing priority via weighted graph convolutional networks. IEEE Trans. Reliab. **70**(2), 563–574 (2021)

Gad, A.G.: Particle swarm optimization algorithm and its applications: A systematic review. Archives of Computational Methods in Engineering. **29**(5), 2531–2561 (2022)

Hashfi, M.I., Raharjo, T.: Exploring the challenges and impacts of artificial intelligence implementation in project management: A systematic literature review. International Journal of Advanced Computer Science and Applications. **14**(9) (2023)

Helm, J.M., Swiergosz, A.M., Haeberle, H.S., Karnuta, J.M., Schaffer, J.L., Krebs, V.E., Spitzer, A.I., Ramkumar, P.N.: Machine learning and artificial intelligence: Definitions, applications, and future directions. Curr. Rev. Musculoskelet. Med. **13**(1), 69–76 (2020)

Hossain, M.Z., Hasan, L., Dewan, M.A., Monira, N.A.: The impact of artificial intelligence on project management efficiency. International journal of management information systems and data science. **1**(05), 1–18 (2024)

Hourri, M., Alaa, N.: HYBRIDIZATION OF NEURAL NETWORKS AND GENETIC ALGORITHMS FOR BETTER CLASSIFICATION. https://www.researchgate.net/publication/357270864_HYBRIDIZATION_OF_NEURAL_NETWORKS_AND_GENETIC_ALGORITHMS_FOR_BETTER_CLASSIFICATION (2021)

Hussain, T., Rashid, T., Abbas, M.: Present role of artificial intelligence in software project management and in the future. SSRN Electronic Journal. (2023)

Iftikhar, A., Alam, M., Ahmed, R., Musa, S., Su'ud, M.M.: Risk prediction by using artificial neural network in global software development. Computational Intelligence and Neuroscience. **2021**(1) (2021)

Institute, P.M.: A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and the Standard for Project Management (CHINESE). Project Management Institute, (2021)

Islam, M., Khan, F., Alam, S., Hasan, M.: Artificial Intelligence in Software Testing: A Systematic Review, vol. 9. IEEE (2023)

Jarrahi, M.H.: Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. Bus. Horiz. **61**(4), 577–586 (2018)

Jia, J., Lai, Y., Yang, Z., Li, L.: The optimal strategy of enterprise key resource allocation and utilization in collaborative innovation project based on evolutionary game. Mathematics. **10**(3), 400 (2022)

Kalogiannidis, S., Kalfas, D., Papaevangelou, O., Giannarakis, G., Chatzitheodoridis, F.: The role of artificial intelligence technology in predictive risk assessment for business continuity: A case study of greece. Risks. **12**(2), 19 (2024)

Khalid, A., Badshah, G., Ayub, N., Shiraz, M., Ghouse, M.: Software defect prediction analysis using machine learning techniques. Sustainability. **15**(6), 5517 (2023)

Khan, S.A., Oshin, N.T., Nizam, M., Ahmed, I., Musfique, M.M., Hasan, M.: AI-Based software testing. In: Lecture Notes in Networks and Systems, pp. 323–334. Springer, Singapore (2024)

Khodabakhshian, A., Puolitaival, T., Kestle, L.: Deterministic and probabilistic risk management approaches in construction projects: A systematic literature review and comparative analysis. Buildings **13**(5), 1312 (2023)

Kolawole, I., Osilaja, A.M., Essien, V.E.: Leveraging artificial intelligence for automated testing and quality assurance in software development lifecycles. International Journal of Research Publication and Reviews. **5**(12), 4386–4401 (2024)

Kostopoulos, G., Davrazos, G., Kotsiantis, S.: Explainable artificial Intelligence-Based decision support systems: A recent review. Electronics **13**(14), 2842 (2024)

Koulinas, G., Xanthopoulos, A., Kiatipis, A., Koulouriotis, D.: A Summary Of Using Reinforcement Learning Strategies For Treating Project And Production Management Problems. IEEE (2018)

Lavingia, K., Patel, R., Patel, V., Lavingia, A.: Software effort estimation using machine learning algorithms. Scalable Computing: Practice and Experience. **25**(2), 1276–1285 (2024)

Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L., Ramasamy, R.K.: Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. Sustainability. **15**(2), 1121 (2023)

Liu, C.-Y., Zou, C.-M., Wu, P.: A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing. IEEE (2014)

Liu, H., Liu, S., Zheng, K.: A reinforcement Learning-Based resource allocation scheme for cloud robotics. IEEE Access. **6**, 17215–17222 (2018)

Mahdi, M.N., Zabil, M.H.M., Ahmad, A.R., Ismail, R.: Software project management using machine learning technique-a review. Appl. Sci. **11**(11), 5183 (2021)

Mahfoodh, H., Obediat, Q.: Software Risk Estimation Through Bug Reports Analysis and Bug-fix Time Predictions. IEEE (2020)

Marinho, M., Sampaio, S., Lima, T., Moura, H.d.: A systematic review of uncertainties in software project management. International Journal of Software Engineering & Applications. **5**(6), 1–21 (2014)

McHugh, M.L.: Interrater reliability: the kappa statistic. Biochemia Medica. **22**(3), 276–282 (2012)

Mengist, W., Soromessa, T., Legese, G.: Method for conducting systematic literature review and meta-analysis for environmental science research. MethodsX. **7**, 100777 (2020)

Miah, M.S., Pasupuleti, M.B., Adusumalli, H.P.: The nexus between the machine learning techniques and software project estimation. Global Disclosure of Economics and Business. **10**(1), 37–44 (2021)

Mohammad Zarour, M.A. Mamdouh Alenezi: A framework to evaluate software engineering program using swebok version 4. Journal of Communications Software and Systems. **21**(1), 66–78 (2024)

Mohammad, A., Chirchir, B.: Challenges of integrating artificial intelligence in software project planning: A systematic literature review. Digital. **4**(3), 555–571 (2024)

Mohammad, A., Chirchir, B.: Challenges of integrating artificial intelligence in software project planning: A systematic literature review. Digital. **4**(3), 555–571 (2024)

Mohammed, S., Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., Patzlaff, H., Naumann, F., Harmouch, H.: The Effects of Data Quality on Machine Learning Performance. https://arxiv.org/abs/2207.14529 (2022)

Munialo, S.W., Muketha, G.M.: A review of agile software effort estimation methods. International Journal of Computer Applications Technology and Research. **5**(9), 612–618 (2016)

Nabeel, M.Z.: AI-Enhanced project management systems for optimizing resource allocation and risk mitigation. Asian Journal of Multidisciplinary Research & Review. **5**(5), 53–91 (2024)

Nakra, V.: Enhancing software project management and task allocation with AI and machine learning. International Journal on Recent and Innovation Trends in Computing and Communication. **11**(11), 1171–1178 (2023)

N., N.: A COMPREHENSIVE REVIEW OF AI'S DEPENDENCE ON DATA. https://www.researchgate.net/publication/380005413_A_COMPREHENSIVE_REVIEW_OF_AI'S_DEPENDENCE_ON_DATA (2024)

Ogunbukola, M.: The Impact of Artificial Intelligence on Project Management: Enhancing Efficiency, Risk Mitigation, and.. https://www.researchgate.net/publication/384266056_The_Impact_of_Artificial_Intelligence_on_Project_Management_Enhancing_Efficiency_Risk_Mitigation_and_Decision-Making_in_Complex_Projects (2024)

Opeyemi Abayomi Odejide: Tolulope Esther Edunjobi: AI IN PROJECT MANAGEMENT: EXPLORING THEORETICAL MODELS FOR DECISION-MAKING AND RISK MANAGEMENT. Engineering Science & Technology Journal. **5**(3), 1072–1085 (2024)

Pachouly, J., Ahirrao, S., Kotecha, K., Selvachandran, G., Abraham, A.: A systematic literature review on software defect prediction using artificial intelligence: Datasets, data validation methods, approaches, and tools. Eng. Appl. Artif. Intell. **111**, 104773 (2022)

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Systematic Reviews. **10**(1) (2021)

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I.: The prisma 2020 statement: an updated guideline for reporting systematic reviews. BMJ. (2021)

Patil, D.: Explainable Artificial Intelligence (XAI): Enhancing transparency and trust in machine learning models. https://www.researchgate.net/publication/385629166_Explainable_Artificial_Intelligence_XAI_Enhancing_transparency_and_trust_in_machine_learning_models (2024)

Perifanis, N.-A., Kitsios, F.: Investigating the influence of artificial intelligence on business value in the digital era of strategy: A literature review. Information **14**(2), 85 (2023)

Petrillo, A., Travaglioni, M., De Felice, F., Cioffi, R., Piscitelli, G.: Artificial intelligence and machine learning applications in smart production: Progress, trends and directions. Technical report, MDPI AG (December (2019)

Pospieszny, P., Czarnacka-Chrobot, B., Kobylinski, A.: An effective approach for software project effort and duration estimation with machine learning algorithms. J. Syst. Softw. **137**, 184–196 (2018)

Raju, K.K., Kumar, G.K., Rao, G.J., Vital, T.P.: Improved Software Effort Estimation Through Machine Learning Challenges Applications and Feature.. https://www.researchgate.net/publication/384537559_Improved_Software_Effort_Estimation_Through_Machine_Learning_Challenges_Applications_and_Feature_Importance_Analysis (2024)

Ramadan, A., Yasin, H., Pektas, B.: The Role of Artificial Intelligence and Machine Learning in Software Testing. https://arxiv.org/abs/2409.02693 (2024)

Ramay, W.Y., Umer, Q., Yin, X.C., Zhu, C., Illahi, I.: Deep neural Network-Based severity prediction of bug reports. IEEE Access. **7**, 46846–46857 (2019)

Rane, N.L., Mallick, S.K., Kaya, Ö., Rane, J.: Techniques and optimization algorithms in machine learning: A review. In: Applied Machine Learning and Deep Learning: Architectures and Techniques. Deep Science Publishing, (2024)

Rangineni, S.: An analysis of data quality requirements for machine learning development pipelines frameworks. International Journal of Computer Trends and Technology. **71**(8), 16–27 (2023)

Reddy, R.K.: The Role of Artificial Intelligence In Project Management For Software Engineering. https://www.researchgate.net/publication/383472569_The_Role_of_Artificial_Intelligence_In_Project_Management_For_Software_Engineering (2024)

Rijwani, P., Jain, S.: Software effort estimation development from neural networks to deep learning approaches. Journal of Cases on Information Technology. **24**(4), 1–16 (2022)

Ritu, Bhambri, P.: Software effort estimation with machine learning – a systematic literature review. Agile Software Development, 291–308 (2023)

Rodríguez Sánchez, E., Vázquez Santacruz, E.F., Cervantes Maceda, H.: Effort and cost estimation using decision tree techniques and story points in agile software development. Mathematics. **11**(6), 1477 (2023)

Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence. **1**(5), 206–215 (2019)

Santos, J.I., Pereda, M., Ahedo, V., Galán, J.M.: Explainable machine learning for project management control. Computers & Industrial Engineering. **180**, 109261 (2023)

Sauer, P.C., Seuring, S.: How to conduct systematic literature reviews in management research: a guide in 6 steps and 14 decisions. RMS **17**(5), 1899–1933 (2023)

Schmidt, J.: Mitigating risk of failure in information technology projects: Causes and mechanisms. Project Leadership and Society. **4**, 100097 (2023)

Secundo, G., Mele, G., Passiante, G., Ligorio, A.: How machine learning changes project risk management: a structured literature review and insights for organizational innovation. Eur. J. Innov. Manag. **27**(8), 2597–2622 (2023)

Şengüneş, B., Öztürk, N.: An artificial neural network model for project effort estimation. Systems. **11**(2), 91 (2023)

Shah, V.: Towards Efficient Software Engineering in the Era of AI and ML: Best Practices and Challenges. https://www.researchgate.net/publication/378395600_Towards_Efficient_Software_Engineering_in_the_Era_of_AI_and_ML_Best_Practices_and_Challenges (2019)

Shamim, M.M.I.: Artificial intelligence in project management: Enhancing efficiency and Decision-Making. GLOBAL MAINSTREAM JOURNAL. **1**(1), 1–6 (2024)

Shao, K., Fu, H., Wang, B.: An efficient combination of genetic algorithm and particle swarm optimization for scheduling Data-Intensive tasks in heterogeneous cloud computing. Electronics **12**(16), 3450 (2023)

Standardization, I.O.: ISO 21500:2021(E) - Project. Context and Concepts. International Organization for Standardization, Programme and Portfolio Management (2021)

Taboada, I., Daneshpajouh, A., Toledo, N., Vass, T.: Artificial intelligence enabled project management: A systematic literature review. Appl. Sci. **13**(8), 5014 (2023)

Taherdoost, H.: Mitra Madanchian: Decision making: Models, processes, techniques. Cloud Computing and Data Science **5**, 1–14 (2023)

Taherdoost, H., Madanchian, M.: Artificial intelligence and knowledge management: Impacts, benefits, and implementation. Computers. **12**(4), 72 (2023)

Tam, C., Moura, E.J.d.C., Oliveira, T., Varajão, J.: The factors influencing the success of on-going agile software development projects. International Journal of Project Management. **38**(3), 165–176 (2020)

Taye, M.M.: Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers. **12**(5), 91 (2023)

Taye, M.M.: Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. Computers. **12**(5), 91 (2023)

Uddin, S., Yan, S., Lu, H.: Machine learning and deep learning in project analytics: methods, applications and research trends. Production Planning & Control, 1–20 (2024)

Wan, Z., Xia, X., Lo, D., Murphy, G.C.: How does machine learning change software development practices? IEEE Trans. Software Eng. **47**(9), 1857–1871 (2020)

Wang, F., Wang, X., Sun, S.: A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. Inf. Sci. **602**, 298–312 (2022)

Wen, J., Li, S., Lin, Z., Hu, Y., Huang, C.: Systematic literature review of machine learning based software development effort estimation models. Inf. Softw. Technol. **54**(1), 41–59 (2012)

Wirtz, B.W., Weyerer, J.C., Sturm, B.J.: The dark sides of artificial intelligence: An integrated AI governance framework for public administration. Int. J. Public Adm. **43**(9), 818–829 (2020)

Young, L., Mansouri, M., Valerdi, R: Estimating the cost and effort of project management for enterprise resource planning systems development projects. 2024 IEEE International Symposium on Systems Engineering (ISSE). **1**(7) (2024)

Zhou, G., Tian, W., Buyya, R., Xue, R., Song, L.: Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions. Artificial Intelligence Review. **57**(5) (2024)

Zhou, G., Tian, W., Buyya, R., Xue, R., Song, L.: Deep reinforcement learning-based methods for resource scheduling in cloud computing: a review and future directions. Artif. Intell. Rev. **57**(5), 124 (2024)