Ingeniería Investigación y Tecnología
volumen XXV (número 3), julio-septiembre 2024 1-11
ISSN 2594-0732 FI-UNAM artículo arbitrado
Información del artículo: Recibido: 5 de abril de 2023, aceptado: 11 de abril de 2024

http://doi.org/10.22201/fi.25940732e.2024.25.3.017

A systematic literature review on machine learning applications for agile project management

Una revisión sistemática de la literatura sobre aplicaciones del aprendizaje automático para la gestión ágil de proyectos

Pérez-Castillo Yadira Jazmín
Instituto Politécnico Nacional, México
Centro de Investigación en Computación
E-mail: yperezc2022@cic.ipn.mx
https://orcid.org/0000-0001-9745-1964

Orantes-Jiménez Sandra Dinora
Instituto Politécnico Nacional, México
Centro de Investigación en Computación
E-mail: dinora@cic.ipn.mx
https://orcid.org/0000-0002-8626-5936

Letelier-Torres Patricio Orlando
Universitat Politècnica de València 0
E-mail: letelier@dsic.upv.es
https://orcid.org/0000-0002-3591-6663

Abstract

Since the rise of agile methods, it has become important to maintain their management and monitoring to succeed in the transformation process from a traditional approach to an agile one. Several authors have used Machine Learnig models to support prediction or estimation processes in the project management framework. However, there are current challenges and areas of opportunity in relation to Agile Project Management in combination with Machine Learning. Therefore, in this paper, we have conducted a Systematic Review of the Literature to understand the current state of Machine Learning applied to Agile Project Management, in order to identify which techniques are currently the most used and thus detect new areas of opportunity.
Keywords: Project management, agile approach, machine learning, systematic review, traditional approach.

Resumen

Desde el surgimiento de los métodos ágiles, se ha vuelto importante mantener su gestión y monitoreo para tener éxito en el proceso de transformación de un enfoque tradicional a uno ágil. Varios autores han utilizado modelos de aprendizaje automático para apoyar procesos de predicción o estimación en el marco de gestión de proyectos. Sin embargo, existen desafíos actuales y áreas de oportunidad en relación con la gestión de proyectos ágiles en combinación con el aprendizaje automático. Por lo tanto, en este documento, hemos realizado una revisión sistemática de la literatura para comprender el estado actual del aprendizaje automático aplicado a la gestión de proyectos ágiles, con el fin de identificar qué técnicas son actualmente las más utilizadas y así detectar nuevas áreas de oportunidad.
Descriptores: Gestión de proyectos, enfoque ágil, aprendizaje automático, revisión sistemática, enfoque tradicional.

http://doi.org/10.22201/fi.25940732e.2024.25.3.017

Introduction

The utility of agile methods has increased substantially in recent years. Many software companies now use agile methods for project development, including Scrum, which has become popular (Schwaber & Sutherland, 2020). Moreover, the *Project Management Institute* (PMI) has collaborated with the *Agile Alliance* (Sutherland, 2022) to create an *Agile Practice Guide* (Project Management Institute, 2017).

All this shows that Project Management (PM) and the use of methods have become an important part of a successful project (Imran & Soomro, 2022). According to *Project Management Body of Knowledge* (PMBOK) (Project Management Institute, 2017), three factors, including time, cost, and scope, are used to analyze the quality of work on a project. However, this is not enough in a real environment to determine the success of a project, as there are other elements that contribute to project success, such as collaboration among team members, satisfaction with the position in the projects, and others (Mamatha & Suma, 2021).

Machine Learning (ML) has become a fundamental tool for PM in a variety of fields, from medicine to engineering. By using ML techniques, project managers can make informed, data-driven decisions, which can improve efficiency and reduce project costs.

In other words, ML involves training an algorithm to "learn" from data and make decisions based on that information. Similarly, previous studies have shown that the algorithms and methods of ML can support the PM.

In a survey conducted by Mamatha & Suma (2021) it is emphasized that ML, in support of project management, will result in project progress, as the tasks of a project manager will generally be easier to handle. For example, automating routine tasks, assigning tasks, etc. This will allow project managers to spend more time on innovation and concentrate on increasing the productivity of teams.

Magaña & Fernández (2015) concluded that, in terms of accuracy, Artificial Intelligence tools outperform traditional tools, and Artificial Intelligence (AI) has been shown to be useful for controlling and monitoring projects.

In the Software Engineering (SE) domain, varieties of ML algorithms for prediction are used; some of them are to predict quality, effort, cost, risks, etc. Among these approaches, there are still challenges and research opportunities, particularly with *Agile Project Management* (APM).

This paper aims to conduct a *Systematic Literature Review* (SLR) to understand the state of the art of ML applied to PM when an agile approach is applied. Currently, most of the literature shows systematic reviews directed at a single prediction approach, such as the cost or effort of a project, in a traditional environment. For this reason, it is relevant to explore the use of these approaches in an agile context, which motivated the realization of this work.

This article includes abbreviated technical terms, shown in Table 9 in the appendix section. The paper is structured as follows: Section 2 provides a summary review and discussion of papers related to ML and PM in general. Section 3 explains the process of selecting and reviewing papers. Section 4 discusses the results of the investigation, and finally, Sections 5 and 6 provide conclusions and future work.

Background

Tracking in an agile approach involves reviewing and evaluating the product and the process of developing the product during the development stage. This is essential to obtain a product that is considered correct and properly managed, and is important because it allows everyone involved to know the status of the project at any time.

There are different techniques and tools that support the monitoring of the progress of a project. Current work focuses mainly on reviews of project effort, cost, risk, and defects using ML techniques and is mostly focused on traditional methods, such as Dos Santos *et al.* (2022) and Pachouly *et al.* (2022).

Other studies, such as those Sudarmaningtyas & Mohamed (2021), have focused on determining the effort estimation technique in an agile approach, showing results that ensure that the three most common techniques are ML (37 %), expert judgment (26 %), and algorithmic models (21 %). In addition, in works such as in Jadhav *et al.* (2022), text mining has been proposed to investigate trends in new models for cost and effort estimation.

In studies such as Sousa *et al.* (2021) and Tiwari *et al.* (2024), there is a notable emphasis on the increasing use of Machine Learning Algorithms (MLA) for risk assessment, particularly in supervised learning. The algorithms commonly used in ML include Decision Tree (DT), Naive Bayes (NB) classifiers, Neural Networks (NN), and Support Vector Machine (SVM). All of these works have contributed to demonstrating that most agile teams estimate software development effort using estimation techniques based on expert judgment, as in Fernández *et al.* (2020) and Srivastava *et al.* (2022)

Finally, Arora *et al.* (2020) based on the Scrum method, show that ML models clearly outperform nonmachine learning and traditional estimation techniques. While current research covers a wide area of opportunity, there is a lack of SLRs dedicated to understanding the current state of ML focused on project tracking in an agile context. These opportunities consider different prediction approaches such as effort, cost, and risks, as well as team characteristics, and tracking techniques such as the Burndown chart or the Kanban board.

For this reason, this SLR aims to provide information on the current state of the art of ML oriented towards APM, considering factors such as the actuality of the work, dataset, algorithms, techniques, and prediction approaches, as well as the tools used. The goal is to determine the most emphasized predictive measurable approaches, as well as common tracking variables such as Sprint duration and user story points, among others.

Method of investigation

To support the development of this SLR, a process consisting of 5 stages was established, which is described in the diagram of Figure 1, based on the methodology for SLR proposed by Kitchenham & Charters (2007).

The stages established for the execution of the SLR will be described in later sections. The process includes searching for the most relevant articles, selecting quality articles, extracting data, and obtaining results. Supporting research questions were established, and based on these, the inclusion and exclusion criteria for the SLR were reviewed. Additionally, the data source and the study selection process were described.

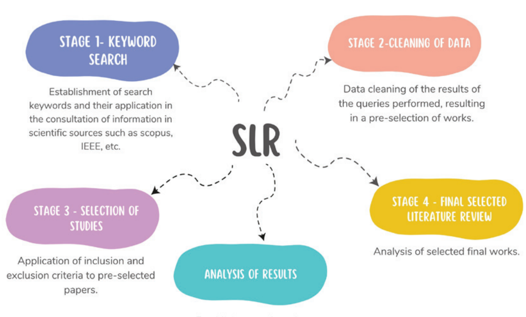

Figure 1. SLR Process

Research questions

The four proposed research questions are:

- **RQ1**: Can ML support APM?
- **RQ2**: Which variables, in relation to agile management, are the most considered?
- **RQ3**: What is the most used MLA for support PM in an agile environment?
- **RQ4**: What is the main predictive approach used within agile management andsupported by ML?

Search keywords

The keywords used as a search strategy to select the data that are shown in Table 1, with predominant terms including "*Agile*", "*Project Management*," and "*Machine Learning*," as these are considered the closest to the research objective.

Table 2 shows the sources consulted from scientific databases; they were chosen because they are freely accessible from the institution where this research was carried out. Table 3 shows the results of the initial search in the databases, considering the previously established search strings. The column *Total* shows the sum of the results of the queries per source, and at the end, a sub-total per query is displayed, leaving 3,099 records per result.

At this stage, only the selection criteria provided by the platforms have been applied, such as Proceedings of a Congress, Conference, Scientific Article, or book; in addition to subject areas and sub-areas, such as Computer Science, Social Sciences, Agricultural and Biological Sciences, Business, Management, etc., all of this depends on those offered by the platforms.

It is important to note that the research includes documents from 2017 to the present, in order to focus on the current advances in ML oriented to PM.

Table 1. Query strings

CLV	Query string
CC01	"Agile Software" AND "Machine Learning"
CC02	"Project Management" AND "Machine Learning"
CC03	"Agile Project Management" AND "Machine Learning"
CC04	"Agile Development" AND "Machine Learning"

Table 2. List of sources consulted

No.	Source	URL
01	Scopus	https://www.scopus.com/
02	IEEE explore	www.ieeeexplore.org
03	ACM digital library	https://dl.acm.org/
04	Science direct	www.sciencedirect.com
05	Springer	https://link.springer.com/

http://doi.org/10.22201/fi.25940732e.2024.25.3.017

Table 3. Initial search results

Source name	CC01	CC02	CC03	CC04	Work
Scopus	112	356	13	47	528
IEEE explore	27	345	3	19	394
ACM digital library	139	412	10	142	703
Science direct	176	833	31	126	1166
Springer	89	148	16	55	308
Total per search result	543	2094	73	389	3099

Selection criteria

After obtaining the information from the consulted databases in the search stage, the data was cleaned based on the title of the work. The cleaning process began by searching for duplicate articles, as well as works that were not aligned with the objective. Among them were research articles that discussed the incorporation of the agile or traditional approach in ML projects, which, according to the search, is of great interest but has nothing to do with the purpose of the research work proposed in this article.

Finally, once the information had been cleaned, we proceeded to classify the papers using the title and abstract in order to determine the most commonly used predictive approaches.

This classification includes common approaches such as efforts, risks, costs, etc. Table 4 shows the results obtained from the above-mentioned classification after having applied the corresponding filters in the data-cleaning stage.

Within the classification of *others*, which include articles of interest, related to advances in PM in general, in addition to works that remained empirical, as a proposal. This classification helped to carry out a pre-selection of papers and in the next stage; a more exhaustive review of each selected article was conducted, based on inclusion and exclusion criteria.

Table 4. Data cleaning results

Ranking	Work
RISKS	8
EFFORTS	25
COSTS	7
SLR	31
TEAM	2
MANAGEMENT	12
PLANNING	3
SUCCESS	5
DEFECTS	3
OTHER	76
Total	172

Inclusion and exclusion criteria

The inclusion and exclusion criteria are:

- The studies that maintain the agile approach as support for PM were considered in this work, discarding the research referring to the traditional approach.
- Papers classified as SRL were excluded, including reviews, comparative studies, and surveys.
- Works considered empirical were excluded, where only the use of ML as support for PM is proposed.

Discussion and analysis of results

After doing a search in several sources, and passing through the data cleaning process, 179 papers were obtained for review, to which inclusion and exclusion criteria were again applied, leaving 15 papers as the final selection. Table 5, shows the list of the 15 selected works in a final cutoff.

Table 5. Selected works in the state-of-the-art review

#	Work	Description
CVE01	Project tracking tool for scrum projects with machine learning support for cost estimation (2021)	This paper describes the design and implementation of a tool that supports various Scrum project-tracking activities, such as the creation of user stories, sprint tasks, and test cases. In addition, the tool supports Scrum project cost estimation based on your sprint tasks (Periyasamy & Chianelli, 2021)
CVE02	Task allocation in distributed agile software development using machine learning approach (2021)	The purpose of the work is to design and implement a method for distributed work allocation and it is based on machine learning (William *et al.*, 2021)
CVE03	A predictive model to estimate effort in a sprint using machine learning techniques (2021)	This paper presents a model to estimate and predict the effort in a Sprint using ML techniques considering several factors that affect a Sprint. The model has been evaluated using several regression algorithms such as linear regression, K-nearest neighbor, DT, polynomial kernel, radius basis function, and MLP. This model has produced more reliable estimates, with low error values and using the MLP algorithm (Ramessur & Nagowah, 2021)
CVE04	Machine learning-based estimation of story points in agile development: industrial experience and lessons learned (2021)	This paper evaluates a new generation machine learning technique, Deep-SE, (Choetkiertikul *et al.*, 2019) to estimate user history points in a project developed with agile methods (Abadeer & Sabetzadeh, 2021)
CVE05	An improved technique for software cost estimations in agile software development using soft computing techniques (2021)	This paper proposes a COCOMO model for soft- ware project cost estimation. It is based on ML for its predictions, using historical data from 57 different organizations representing the public and private sectors in Sudan (Bushra & Kadam, 2021).
CVE06	Machine learning application in LAPIS agile software development process (2020)	This paper considers the contributions of work teams to the continuous improvement process, with the goal of expanding opportunities for improvement, based on data. It also consider in- formation from retrospective meetings to support the proposed machine learning model, the information is obtained from the LAPIS process, an agile, improvement-oriented product delivery process developed by Logo Yazÿlÿm (Tekbulut *et al.*, 2020)
CVE07	The method of agile projects success evaluation using machine learning (2020)	The purpose of this study is to develop a unified method for measuring and predicting the success of agile IT projects based on the machine learning approach (Veido *et al.*, 2020)
CVE08	Machine learning models to predict agile methodology adoption (2020)	The main objective of this work is to use machine learning to develop predictive models for the adoption of the Scrum methodology, identifying a preliminary model with the highest prediction accuracy (Hanslo & Tanner, 2020)
CVE09	A predictive model to identify kanban teams at risk (2019)	In this paper, several models were built to demonstrate that selected variables could help identify teams at risk when the team uses a Kanban framework, resulting in better performance using the KNN algorithm with univariate feature selection (Shamshurin & Saltz, 2019)
CVE10	An ensemble-based model for predicting agile software development effort (2019)	It proposes a predictive model to estimate the effort required to develop user stories for agile software development projects by creating a MLA based on sets (Malgonde & Chari, 2019)
CVE11	An effort estimation support tool for agile software development: An empirical evaluation (2019)	This work proposed and evaluated a tool to support effort estimation. The tool uses historical data to build a predictive model using DT to provide effort estimates during the planning meeting (Dantas *et al.*, 2019).
CVE12	Predicting failures in agile software development through data analytics (2018)	This paper presents a method for predicting software failures in subsequent sprints in agile environments. This is obtained using analytical and statistical methods (Batarseh & Gonzalez, 2018)
CVE13	Bayesian network model for task effort estimation in agile software development (2017)	The model predicts the task effort and is independent of the agile methods used (Dragicevic *et al.*, 2017)
CVE14	Machine learning based approach for user story clustering in agile engineering (2023)	This work used the approach of machine learning clustering algorithms to organize groups of user stories, utilizing the K-means and K-medoids clustering algorithms. (Kumar *et al.*, 2023)
CVE15	Effort and cost estimation using decision tree techniques and story points in agile software development (2023)	This article uses hybrid models composed of algorithmic models and learning-oriented techniques as a method for estimating project-level effort in agile frameworks (Rodríguez *et al.*, 2023)

The analysis of the 15 papers selected revealed the following results:

- **Python, programming language of preference for authors**. The analysis showed that most of the authors chose Python as the programming language to implement the proposed ML models due to its accessibility in the use of the librariesit provides.
- **The predilection for a predictive approach to risk and effort**. Most authors focus on supporting PM in determining risks and efforts of a project, leaving aside some others predictive approaches that can be of great support to the Project Manager, such as the analysis of user histories. Figure 2 shows the graphical result of the analysis by prediction approach considered.

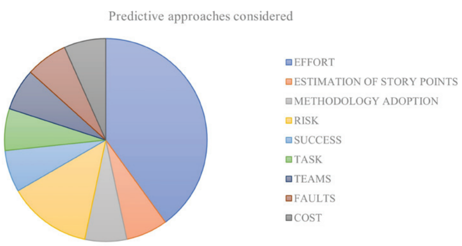

Figure 2. Results by prediction approach

- **The use of user stories as a main predictive variable**. Although some others are taken into consideration, such as COCOMO for cost estimation in conjunction with ML and the agile approach, most papers consider US related information to generate both dependent and independent variables.

  However, there was an absence in the use of other monitoring techniques for agile projects, which could well function as predictive variables, such as the Burn Down graph, which allows us to know the progress of work considering parameters such as status, execution times, among others. It should be noted that for a better analysis, a classification of variables was made, which is shown in Figure 3, which shows the graph resulting from the analysis of the classification of variables.

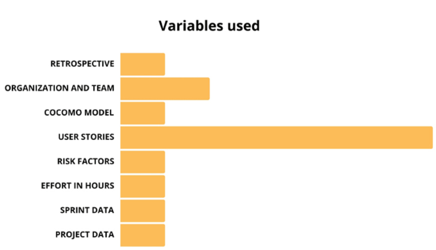

Figure 3. Results of variables considered

- **Predominance of the agile approach**. Most of the papers consider the agile approach in general, i.e., they do not focus on a single methodology, such as Scrum, or at least it is not mentioned. Rather, they maintain an agile context approach. Figure 4 shows the graphical result of the analysis by method considered.

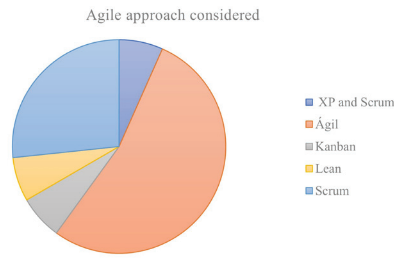

Figure 4. Results by method considered

- **MLA analysis and selection.** Most of the selected papers carried out previous comparative studies of certain MLAs to choose the most appropriate one or validate the proposed one. Among the most commonly, used algorithms for comparison are SVM, KNN, ANN, DT, RR, LR, and BN. It is important to mention that although a study does not indicate the process of a comparative analysis to choose the MLA to be used, it does mention and justify its choice.
- **Deficiency in requirements analysis with an agile and ML approach.** Within the general analysis of SLR, it was observed that there is not much literature on requirements analysis in an agile approach, which could lead to future work.
- **The topicality of the subject.** The use of ML to support PM is in use and has been visualized in the literature for some years; several authors have kept updated the use of new ML techniques to support PM. Table 6 shows the results of the analysis of works by year.

Table 6. Results by year

Year	Total works
2017	1
2018	1
2019	4
2020	3
2021	6
Total	15

The results of the analysis of the 15 selected papers are summarized in the comparative table in the Table 8 below and in Table 7 is shows a list of acronyms corresponding to the column headings, which is intended to support the reading of the comparative table.

ML has become an increasingly popular and effective tool in support of PM. This is because ML can help at work teams to make more decisions that are informed, identify patterns and trends in large data sets, and automate repetitive tasks. This study are shown the areas where ML is having a significant impact, some of which are in the prediction of project risks and problems.

MLA can analyze large amounts of real-time and historical data to identify patterns and trends that may indicate a potential project risk or problem. This allows project teams to take proactive steps to mitigate risks and fix problems before they affect the project.

Another area where ML is proving useful is in scheduling and resource allocation optimization, as in (William *et al.,* 2021).

MLA can analyze historical data and trends to identify patterns in resource utilization and task duration. This allows project teams to plan and allocate resources more effectively and efficiently, which can improve the quality of deliverables and reduce project costs.

In this study, it was also found that there are still challenges around APM, which provide opportunities for new lines of research. The lack of effective tools presents an opportunity for IA significantly improve APM. With the study, it was observed that ML could support APM, allowing the Project Manager to focus on improving the quality of project development, reducing data analysis time, such as task execution times, development times, etc.

In addition, it was observed that the most predominant variables are information related to the US, as well as sprint data, such as start and end time, and general team data. The main context in which ML is used to support PM is risk analysis and development effort.

We also found a wide variety of MLAs considered by the studies analyzed, both for comparing new and old models, some of them being RR, KNN, DT, SVM. During the development of SLR, it was emphasized that there are some empirical works, such as that mentioned in (Damet *al.,* 2019), which shows a proposal of IA as a support to PM. This type of work, despite not having been taken into consideration in the final selection (only proposals without further development) may be relevant in the future.

As a result of the SLR, it can be noted that the use of ML is of great contribution to APM, however, it is considered that is necessary to refine and unify the use of certain prediction approaches. For example of the above, several works only focus on one, such as costs or efforts, but it would be worthwhile to create an ML model that can consider different approaches by unifying variables. This could also be considered to generate new studies in IA and PM in general.

Table 7. Column abbreviations for comparative table

CLV	Key of the work, based on Table 5
MLA	ML algorithms used
MA	Agile method considered
CA	Comparison of algorithms
EP	Predictive approach considered

Table 8. Comparative table of selected papers in the SLR

CLV	Dataset	MLA	Variables	MA	CA	EP
CVE01	16 projects, 23,000 user stories. Projects came fromApache, Atlassian, Moodle and others	K- fold	US	Scrum	No	Effort
CVE02	Kaggle repository dataset,a platform that allows users to find and publish Datasets	NSL	Organization and team	Agile	Yes	Task
CVE03	Simulates a dataset consisting of 2100 records, performing a combination of intensity levels with 12 identified factors	MLP	US	Agile	Yes	Effort
CVE04	Collection of 4,727 created in Jira since its adoption at Privacy Analytics Inc. in January 2014 through October 2020	DL	US	Agile	Yes	Spe
CVE05	SEERA group dataset, located in the Middle East and North Africa region	NB	COCOMO model	Agile	Yes	Cost
CVE06	To validate the study, information from the retrospective of 6 sprints were used	NPL	SR	Lean	No	Teams
CVE07	2,000 agile projects	NN	Project data	Agile	No	Success
CVE08	Consider a set of 207 data used to train and test prediction models	MLR	Organization and team	Scrum	Yes	Adoption
CVE09	Dataset with information from 80 Kanban projects	KNN	US	Kanban	Yes	Risk
CVE10	503 stories from 24 ASD projects from 2012 to 2016	EP	US	Agile	Yes	Effort
CVE11	Consider 26 backlogs, with a total of 530 user stories and 1879 tasks	DT	Effort in hours	Scrum	No	Effort
CVE12	181 records collected from Scrum projects	RM	Sprint data	Scrum	No	Faults
CVE13	Database of 160 real agile project tasks	BN	US	Agile	Yes	Effort
CVE14	Not located	Clustering	US	Agile	Yes	Effort
CVE15	21 agile projects developed by six software houses from Pakistan	DR, RF, AdaBoost	US	Agile	Yes	Effort cost

Conclusions

ML and PM are two important disciplines that complement each other. ML involves the use of algorithms and models to learn from data and make accurate predictions, while PM focuses on planning, executing, and controlling projects to achieve specific objectives within a given time frame and budget. In the age of data, ML is increasingly used to improve PM. ML models can help project managers analyze large amounts of data, identify patterns, and make accurate predictions. For example, predictive analytics can help project managers predict the time it will take to complete a project or the total cost of a project. In addition, ML models can also help project managers identify potential risks and make more informed decisions.

Another important aspect is the use of ML in process automation, which can reduce the workload of the project team and enable greater efficiency in PM. Process automation can reduce the risk of human error and improve the quality of project deliverables.

On the other hand, the number of projects following an agile approach has increased significantly in recent years, not only in the software industry but also in other non-IT domains (Digital.ai, 2023). In addition, the success of ML in solving prediction problems has allowed for supporting new areas, such as software engineering, for some years now.

Therefore, the objective of this study was to carry out an SLR to determine the current state of ML focused on APM. To carry out this research, the process was established as follows: keyword search, data cleaning, selection of papers, a pre-selection, and finally, final analysis and selection of papers. As a result, 15 papers were selected after applying the previously established selection criteria.

From the study of the selected papers, it can be remarked that one of the main predictive variables used by the authors was the US, although others were also considered, such as Sprint data, in addition to team and organizational information. A variety of ML algorithms (MLAs) was also found to support the prediction of certain approaches, some of which were SVM, KNN, and DT, among others. In terms of prediction approaches, development efforts and project risks predominate.

Likewise, with the realization of this SLR, it was observed that the current literature contains information mainly focused on traditional management, and although there are works focused on APM, they are oriented to a certain prediction direction, such as costs or risks.

Therefore, and according to the gathered information for this work, it can be inferred that there is no current review that focuses on ML models as a support for APM, considering different prediction factors, which motivated the realization of this work. Finally, APM, in conjunction with ML, can support better PM, for example, by assigning tasks to teams that are geographically distributed; getting to know the team better through sentiment analysis, determining delivery times, predicting the direction of a Sprint, considering the US, or determining project risks, etc.

Lastly, it is important to keep in mind that ML is not a magic bullet for all PM problems. ML models require quality data to work properly, and project managers must ensure that the data used is accurate and representative. Furthermore, ML models are only a tool and do not replace human decision-making and PM expertise.

In summary, ML and PM can work together to improve efficiency and accuracy in PM. However, it is important for project managers to understand the limitations and opportunities of this technology and use it as a tool to improve decision-making and process automation.

Future work

The application of AI and ML in PM is a rapidly evolving field of research. Here are some examples of future work that could contribute to the advancement of this area:

1. *Development of risk prediction algorithms*: ML can help develop more accurate and efficient risk prediction models by analyzing historical data from similar projects to predict potential problems in the current project.
2. *Improving time and cost estimation:* ML techniques, such as regression models, could be used to predict task duration and resource cost, improving the accuracy of time and cost estimation in PM.
3. *Automating resource allocation:* ML can help automate resource allocation by analyzing the skills and availability of team members. For example, clustering algorithms could be developed to group team members according to their skills and assign tasks based on these groupings.
4. *Project quality analysis*: ML can assist in assessing project quality by developing classification models to identify projects more likely to meet established objectives and those with higher risks of failure.
5. *Improving change management:* ML can help improve change management by analyzing project change data and predicting its impact on the schedule and budget. For example, sentiment analysis algorithms could be developed to evaluate the response to change.

This work highlights the current focus on traditional project analysis in conjunction with ML, with few studies related to APM. It underscores new areas of opportunity for research in APM. While the field of management is broad, ML has the potential to significantly improve PM in an agile approach by reducing costs and increasing efficiency. Future work in this field could further explore the possibilities of this technology in APM.

Appendix

Table 9. List of abbreviations

AI	Artificial Intelligence
AM	Agile Methodologies
ANN	Artificial Neural Network
APM	Agile Project Management
ASD	Agile software development
BN	Bayesian Network
BP	Back Propagation
COCOMO	Constructive Cost Model
DL	Deep Learning
DT	Decision tree
EP	Ensemble Prediction
KNN	K-Nearest Neighbors
LAPIS	Logo Agile Process Improvement System
LR	Linear Regression
ML	Machine learning
MLA	Machine Learning Algorithms
MLP	Multi-Layer Perception
MLA	Machine Learning Algorithms
MLR	Multiple Linear Regression
MTBF	Mean Time between Failures
NB	Naive Bayes
NLP	Natural Language Processing
NSL	Neural Structured Learning
NN	Neural Networks
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
RR	Ridge Regression
SLR	Systematic literature review
SPE	Story Point Estimation
SR	Sprint Retrospective
SVM	Support Vector Machine
US	User Stories
WIP	Work in Progress

References

Abadeer, M., & Sabetzadeh, M. (2021). Machine learning-based estimation of story points in agile development: Industrial experience and lessons learned. *IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 106-115. https://doi.org/10.1109/REW53955.2021.00022

Arora, M., Verma, S., Kavita, & Chopra, S. (2020). A systematic literature review of machine learning estimation approaches in scrum projects. *Advances in Intelligent Systems and Computing*, 1040, 573-586. https://doi.org/10.1007/978-981-15-1451-7_59

Batarseh, F., & Gonzalez, A. (2018). Predicting failures in agile software development through data analytics. *Software Quality Journal*, 26(1), 49-66. https://doi.org/10.1007/s11219-015-9285-3

Bushra, Q., & Kadam, A. (2021). An improved technique for software cost estimations in agile software development using soft computing techniques. *IT in Industry*, 9(2). https://doi.org/10.1049/icp.2023.1541

Choetkiertikul, M., Dam, H., Tran, T., Pham, T., Ghose, A., & Menzies, T. (2019). A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*, 45(7), 637-656. https://doi.org/10.1109/TSE.2018.2792473

Dam, H., Tran, T., Grundy, J., Ghose, A., & Kamei, Y. (2019). Towards effective ai-powered agile project management. *IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER*), 41-44. https://doi.org/10.1109/ICSE-NIER.2019.00019

Dantas, E., Costa, A., Vinicius, M., Perkusich, M., Almeida, H., & Perkusich, A. (2019). An effort estimation support tool for agile software development: An empirical evaluation. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, 82-87. https://doi.org/10.18293/SEKE2019-141

Digital.ai. (2023). The 17th State of Agile Report 17th State of Agile. Retrieved from https://stateofagile.com/

Dos Santos, R. A., Vieira, D., Bravo, A., Suzuki, L., & Qudah, F. (2022). A systematic mapping study on the employment of neural networks on software engineering projects: Where to go next? *Journal of Software: Evolution and Process*, 34(3). https://doi.org/10.1002/smr.2402

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, 127, 109-119. https://doi.org/10.1016/j.jss.2017.01.027

Fernández-Diego, M., Méndez, E., González-Ladrón-De-Guevara, F., Abrahão, S., & Insfran, E. (2020). An update on effort estimation in agile software development: A systematic literature review. *IEEE Access*, 8, 166768-166800. https://doi.org/10.1109/ACCESS.2020.3021664

Hanslo, R., & Tanner, M. (2020). Machine learning models to predict agile methodology adoption. *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*, 697-704. https://doi.org/10.15439/2020F214

Imran, R., & Soomro, T. (2022, 2). Mapping of agile processes into project management knowledge areas and processes. *International Conference on Business Analytics for Technology and Security (ICBATS)*, 1-12. https://doi.org/10.1109/ICBATS54253.2022.9759013

Jadhav, A., Kaur, M., & Akter, F. (2022). Evolution of software development effort and cost estimation techniques: Five decades study using automated text mining approach. *Mathematical*

Problems in Engineering, 1-17. https://doi.org/10.1155/2022/5782587

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Kumar, B., Tiwari, U. K., & Dobhal, D. C. (2023). Machine learning based approach for user story clustering in agile engineering. *SN Computer Science*, 4(6), 763. https://doi.org/10.1007/s42979-023-02212-2

Magaña-Martínez, D., & Fernandez-Rodriguez, J. C. (2015). Artificial intelligence applied to project success: a literature review. *International Journal of Interactive Multimedia and Artificial Intelligence*, 3(5), 77. https://doi.org/10.9781/ijimai.2015.3510

Malgonde, O., & Chari, K. (2019). An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*, 24(2), 1017-1055. https://doi.org/10.1007/S10664-018-9647-0/TABLES/29

Mamatha, R., & Suma, K. G. (2021). Role of machine learning in software project management. *Journal of Physics: Conference Series*, 2040(1). https://doi.org/10.1088/1742-6596/2040/1/012038

Pachouly, J., Ahirrao, S., Kotecha, K., Selvachandran, G., & Abraham, A. (2022). A systematic literature review on software defect prediction using artificial intelligence: Datasets, data validation methods, approaches, and tools. *Engineering Applications of Artificial Intelligence*, 111, 104773. https://doi.org/10.1016/J.ENGAPPAI.2022.104773

Periyasamy, K., & Chianelli, J. (2021). A project tracking tool for scrum projects with machine learning support for cost estimation. *EPiC Series in Computing*, 76, 86-76. https://doi.org/10.29007/6vwh

Project Management Institute. (2017). *A guide to the project management body of knowledge* (PMBOK® guide). 6th ed. Project Management Institute.

Project Management Institute. (2017). *Agile practice guide.*

Rai, A. K., Agarwal, S., Khaliq, M., & Kumar, A. (2019). Quantitative analysis of development environment risk for agile software through machine learning. *International Journal of Recent Technology and Engineering*, 7(6), 83-89.

Ramessur, M. A., & Nagowah, S. D. (2021). A predictive model to estimate effort in a sprint using machine learning techniques. *International Journal of Information Technology*, 13(3), 1101-1110. https://doi.org/10.1007/s41870-021-00669-z

Rodríguez-Sánchez, E., Vázquez-Santacruz, E. F., & Cervantes-Maceda, H. (2023). Effort and cost estimation using decision tree techniques and story points in agile software development. *Mathematics*, 11(6), 1477. https://doi.org/10.3390/math11061477

Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego*. Retrieved on https://scrumguides.org/

Shamshurin, I., & Saltz, J. S. (2019). A predictive model to identify Kanban teams at risk. *Model Assisted Statistics and Applications*, 14(4), 321-335. https://doi.org/10.3233/MAS-190471

Sousa, A., Faria, J. P., & Mendes-Moreira, J. (2021). An analysis of the state of the art of machine learning for risk assessment in software projects. *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, 217-222. https://doi.org/10.18293/SEKE2021-097

Srivastava, P., Srivastava, N., Agarwal, R., & Singh, P. (2022). Estimation in agile software development using artificial intelligence. *Lecture Notes in Networks and System*, 376, 83-93. https://doi.org/10.1007/978-981-16-8826-3_8

Sudarmaningtyas, P., & Mohamed, R. (2021). A review article on software effort estimation in agile methodology. *Pertanika Journal of Science and Technology*, 29(2), 837-861. https://doi.org/10.47836/pjst.29.2.08

Sutherland, J. (2022, March 24). Manifesto for Agile Software Development. Retrieved on http://agilemanifesto.org/

Tekbulut, T., Canbaz, N., & Kaya, T. O. (2020). Machine learning application in LAPIS agile software development process. *Turkish National Software Engineering Symposium (UYMS)*, 1-6. https://doi.org/10.1109/UYMS50627.2020.9247069

Tiwari, S. P., Phonsa, G., & Malik, N. (2024). Estimation approaches of machine learning in scrum projects. *Lecture Notes in Networks and Systems*, 731, 103-111. https://doi.org/10.1007/978-981-99-4071-4_9

Veido, D., Misnevs, B., & Plotkin, A. (2020). The method of agile projects success evaluation using machine learning. *Lecture Notes in Networks and Systems*, 117, 478-487. https://doi.org/10.1007/978-3-030-44610-9_47/COVER/

William, P., Kumar, P., Chhabra, G. S., & Vengatesan, K. (2021). Task allocation in distributed agile software development using machine learning approach. Proceedings of IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications, CENTCON 2021, 168-172. Retrieved on https://doi.org/10.1109/CENTCON52345.2021.9688114

Cómo citar:

Pérez-Castillo, Y. J., Orantes-Jiménez, S. D., & Letelier-Torres, P. O. (2024). A systematic literature review on machine learning applications for agile project management. *Ingeniería Investigación y Tecnología*, 25 (03), 1-11. https://doi.org/10.22201/fi.25940732e.2024.25.3.017