AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)

Report for the course of Methods In Software Engineering

Software Project Management Using Machine Learning Technique-A Review

May 16, 2023

Prepared By:

MOMAHIDA-AN-NOOR (22-92685-3)
ABRAR AHMED KHAN (22-92681-3)
BORHAN UDDIN (22-92373-1)
MD. PIAL HASAN (23-92900-1)

Supervised by:
Dr. S.M. Hasan Mahmud
Assistant Professor, Dept. of Computer Science

Contents

Abstract

Project Management Planning and evaluation have great meaning for success in the performance of an entire project. It is a hectic process that involves planning, managing, and organizing resources to the expected results and objectives within the deadline and budget. Project managers need to analyze the main factors and criteria the project needs to succeed. It is the blueprint of the project that shouldn't be neglected. One of the main reasons a project fails for a software company is either the inexperience of the manager or lack of knowledge in the project management area. This costs a lot of money and time for the company, and in the worst-case scenario, they lose a potential client for rival companies. This article covers numerous articles on various machine learning applications for software project management. This study wishes to present an analysis that combines simulation models with machine-learning methods to determine the impact of key organizational elements on project success and to extract reliable analysis. In an effort to help readers better comprehend what we are attempting to discuss, we tried to divide the paper into a few sub-sections. The paper includes reviews on software project management, software platforms and their comparison, risk analysis, estimation, accuracy, limitations, and future work to solve the current needs. In conclusion, we tried to prove machine learning is very successful at minimizing project flaws and offers an additional method for effectively reducing likelihood chances and raising the software development performance ratio.

Keywords: Software project management, Machine learning technique, Assessment, Estimation, Risk assessment, Project failure, Management knowledge areas, Model prediction.

1 Introduction

The purpose of a well-established software company is to offer software products and profit from them. A project is a short-term endeavor that produces a distinct deliverable. Enhancing project performance and ensuring project success are concerns companies must overcome while managing software projects. Lack of knowledge, experience, tools, and methodologies during project execution is frequently the cause of project failure risk. Building prediction models using either a statistical approach, such as linear regression and correlation analysis, or machine learning (ML) techniques, such as Artificial Neural Networks (ANN), can be done utilizing the data saved in the projects' historical databases [1]. Project management objectives include initiating, planning, executing, regulating, and closing projects, as well as controlling the operations of the project team within the defined time, scope, budget, and quality standards to achieve all agreed-upon goals, and software project management refers to scheduling, planning, resource allocation, and execution. [2].

Considering the simple reality that project literature discusses a project's success and failure, there is a persistent debate on how project improvements should be measured and communicated. According to a review of the literature, significant progress has been made in the application of systems thinking as a holistic discipline that recognizes projects as interconnected technical and social factors producing behavior [3].

The main purpose of this paper is to reduce the cost, effort, and amount of time of an overall project. But it's not a simple task as each project varies depending on the criteria and what is the main concern of the customer. Every part of the project should be well specified at this phase because planning is a factor in how the project is carried out. Every factor that had an impact on the process was meant to be experimentally assessed and examined [4][5].

The study is structured in such a way that it covers the entire software management criteria while proceeding on to a literature review. The discussed scheduling techniques are then classified and compared. Later on, we discussed the different software platform and their frameworks of using machine learning with some case studies. The article outlines the current limitations of the applications and suggests ways to improve them in the future. Lastly, a conclusion was presented to finalize our discussion.

2 Background

The utilization of machine learning (ML) in software project management (SPM) has gained considerable attention in recent research. ML techniques provide powerful tools for analyzing large datasets and making data-driven decisions in various aspects of SPM.

Researchers have explored the application of ML to improve the project planning process. By analyzing historical project data and leveraging ML algorithms, project managers can make more accurate estimations of tasks, resource requirements, and project timelines. This enables better decision-making and resource allocation during the planning phase. ML is also used for software project risk management and failure prediction. ML models analyze project data to identify risk factors and predict the likelihood of project failures or quality issues. This allows project managers to proactively manage risks and allocate resources more effectively to mitigate potential issues. In terms of resource allocation and team management, ML techniques have been used to optimize resource assignments and facilitate workload balancing. By considering project requirements, individual skills, and team dynamics, ML algorithms suggest optimal resource allocations, enhancing overall project efficiency and team collaboration.

ML has proven valuable in monitoring project performance and providing decision support. Real-time monitoring using ML models helps project managers detect deviations from planned schedules, identify bottlenecks, and receive early warnings of potential problems. This enables timely decision-making and prompt corrective actions. Another area where ML has made an impact is software defect detection and quality assurance. ML models analyze code repositories to identify code odors and predict potential bugs. This allows project managers to focus their efforts on critical areas and allocate testing resources effectively.

3 Literature Review

Software project management is an indispensable aspect in the realm of project execution and delivery. The emergence of machine learning (ML) techniques has provided a new avenue for scholars to investigate their utility in augmenting software project management systems. The purpose of this literature review is to present a comprehensive survey of extant research in this field, with a specific emphasis on the integration of ML techniques in software project management systems.

The first paper conducted a comprehensive review that explored the potential of ML techniques in software project management. Their study encompassed various project management aspects, including project planning, resource allocation, risk management, and defect prediction [1]. The authors highlighted the advantages of ML algorithms in facilitating data-driven decision-making processes, which can significantly enhance software project management systems. The second paper extensively focused on the practical implementation of machine learning (ML) techniques in predicting potential failures in software project management knowledge domains [2]. The researchers employed a range of diverse ML algorithms, including decision trees, support vector machines, and neural networks, to identify crucial factors contributing to project failures, particularly in areas such as scope, time, and cost management. The study's outcomes offer vital insights into proactive management strategies that empower project managers to proficiently recognize and mitigate risks, ultimately leading to a significant improvement in project success rates. The third article presented a novel technique that merged artificial neural networks (ANNs) and systems thinking principles into software project management systems. Incorporating the complex interrelationships among project variables and adopting a comprehensive perspective, the objective of this study was to enhance competencies in risk assessment. The integration of ANNs facilitated a more precise risk evaluation and enlightened decision-making in software project management, culminating in superior risk management and project results [3]. The fourth paper explored the application of Machine Learning (ML) techniques in various aspects of software project management. The research analyzed the usage of ML techniques, such as decision trees, random forests and support vector machines, in various aspects of software project management such as effort estimation, defect prediction and project scheduling [4]. The results underscored the benefits of ML in augmenting project planning and decision-making processes. The utilization of ML techniques empowers project managers to leverage historical data, identify patterns, and generate precise estimates for required effort, predict potential defects, and optimize project schedules, ultimately enhancing project efficiency and success rates. The fifth paper presented a compelling case study that integrated machine learning (ML) algorithms and simulation methods to optimize software project management systems. By combining these techniques, the research aimed to enhance resource allocation and project scheduling. The study effectively demonstrated the potential of the integrated approach that utilizes simulation-based experiments and ML algorithms to achieve superior project outcomes and mitigate development risks [5]. The utilization of ML techniques enables project managers to optimize resource allocation, identify critical bottlenecks, and make informed decisions based on data to improve project schedules and overall project success.

The reviewed studies emphasize the benefits of ML in various aspects of project management, such as project planning, risk assessment, resource allocation, and failure prediction. The findings offer valuable insights for practitioners and researchers seeking to leverage ML to optimize software project management systems.

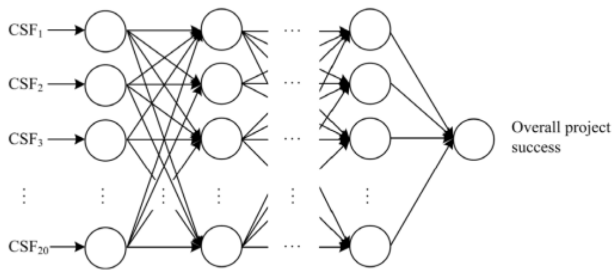

Figure 1: A MISO model [3]

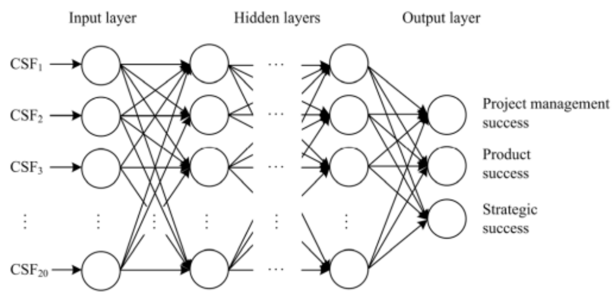

Figure 2: A MIMO model [3]

4 Methodology

This review paper adopts an artificial neural network (ANN) approach to analyze and predict project outcomes [3]. The methodology encompasses two ANN models, the Multiple-Input Single-Output (MISO) model and the Multiple-Input Multiple-Output (MIMO) model.

In figure 1 the MISO model comprises a multilayer neural network architecture with multiple inputs representing critical success factors (CSFs) and a single output representing overall project success. In this paper, the MISO model is used to predict the overall success of projects.

On the other hand, in figure 2 the MIMO model also utilizes a multilayer architecture, but with multiple inputs and multiple outputs. Each output of the MIMO model represents a specific dimension of project success, including project management success (PMS), product success (PS), and strategic success (SS). The MIMO model is employed to predict project success across these dimensions.

4.1 Data Collection

The list of the 27 performance measures grouped by project success dimensions is presented in Figure 3. These items were also measured on a five-

Project Management Success	Product Success	Strategic Success
Project Cost	Benefits	Return on investment
Project time	Satisfied objectives	Revenue measures
Full Scope	Satisfied project requirements	New business opportunities
Milestones	Stakeholder Satisfaction	New markets
Functionality	Client satisfaction	Derived products
Number of defects	Product/result usable	Competitive advantage
Use of resources	Product/result in use	New or expanded core competency/ capability
Agreed scope changes	Product/result useful	Improved processes
Change requests	Fulfilled expectations	Enhanced reputation

Figure 3: List of performance measures [3]

point Likert scale (1 is strongly disagree to 5 is strongly agree). To gather the necessary information, project managers were personally contacted and targeted as key informants. This approach allowed for additional clarification through follow-up questions, resulting in almost 100

4.2 MIMO and MISO model

To label the output variables for the MIMO model, success criteria are applied based on scores obtained for each success dimension. Scores above 4.00 are labeled as 1, indicating success, while scores below 4.00 are labeled as 0, indicating failure. For the MISO model, projects are assigned to clusters based on their success scores in all three dimensions. Projects with scores greater than 4.00 in at least two dimensions are labeled as 1, representing successful projects, while the remaining projects are labeled as 0, indicating failed projects.

The evaluation of input variables follows a two-step process. Initially, an independent sample t-test is conducted to compare the means of each input variable between successful and failed projects. Non-discriminative variables are identified based on their p-values. Subsequently, a correlation matrix is utilized to establish threshold values for maximum correlation and minimum significance. Relevant variables with the lowest p-values among highly correlated variables are retained.

4.3 ANN architectures (accuracy and processing time)

The comparison of training procedures, network designs, preprocessing techniques, transformation functions, and the use of the Leave-p-out (LPO) approach are just a few of the studies performed to optimize the ANN models. The models' efficiency is evaluated based on their accuracy and processing speed. Both the MIMO and MISO models are evaluated with two training algorithms: the Scaled Conjugate Gradient Algorithm (SCGA) and the Resilient Backpropagation Algorithm (RPA). For the MIMO model, RPA shows more accuracy, whereas for the MISO model, SCGA shows faster processing.

Different network architectures, ranging from one-layer to six-layer configurations, are evaluated. A one-layer network with 40 neurons in the hidden layer and a three-layer network with a 40/20/10 structure are the two architectures that execute the MIMO model the best. With the exception of the six-layer networks, which show reduced accuracy, there is no discernible variation in accuracy for the MISO model between the various network topologies.

In this paper [3] normalization techniques, specifically [0, 1] and [-1, 1] normalization, are applied. The results indicate that [1, 1] normalization yields better accuracy and processing time for the MIMO model. For the MISO model, [1, 1] normalization is exclusively utilized. Different combinations of transformation functions (tansig, purelin, and satlin) are tested for the hidden and output layers. The satlin-satlin and tansig-satlin combinations exhibit superior accuracy and processing time compared to the tansig-purelin combination for both the MIMO and MISO models.

Lastly, the LPO method is employed, where samples are sequentially or randomly removed from the dataset for testing purposes. The random LPO method yields slightly better accuracy for the MIMO.

4.4 Discussion

This paper study presented the performance of a model and a MIMO model for predicting the success or failure of software projects [3]. The following metrics were used for performance evaluation: Accuracy (ACC), False-positive ratio (FPR), Precision (PREC), Recall (REC), F-Score, and Area under the ROC curve (AUC).

For the MISO model, which predicts overall

	ACC	FPR	PRE C	REC	F-SCORE	AUC
MISO	0.83	0.13	0.87	0.88	0.88	0.84
MIMO	0.72	0.26	0.74	0.79	0.76	0.75
PMS	0.73	0.24	0.76	0.82	0.79	0.79
PS	0.80	0.13	0.87	0.87	0.87	0.81
SS	0.62	0.42	0.58	0.67	0.62	0.64

Figure 4: Binary classification tasks (MISO and MIMO).

project success vs. failure, the results showed an accuracy of 0.83 indicating a high capability of predicting project outcomes. The FPR was 0.13 indicating a low probability of false alerts for failed projects being incorrectly identified as successful. The precision was 0.87 indicating a high probability of correctly predicting successful projects. The recall was 0.88 indicating a high sensitivity in predicting positive classes. The F-Score was 0.88 demonstrating a balance between precision and recall. The AUC was 0.84, indicating overall high predictive quality.

For the MIMO model, which predicts success vs. failure in multiple project success dimensions, the average accuracy was 0.72. The model performed well in predicting success vs. failure, but there were significant differences in performance across different project success dimensions. The precision was 0.76 for PMS (Project Management Success) and 0.87 for PS (Project Success), indicating high probabilities of correctly predicting success in these dimensions. However, the precision for the SS (Stakeholder Satisfaction) dimension was unacceptably low. The average recall was 0.79, indicating reasonably good performance overall. The average AUC was 0.75, indicating the overall quality of the model's predictions. The FPR showed a high probability of incorrectly classifying failure as success in the SS dimension.

The artificial neural network (ANN) models utilized in this study were shown to be more accurate than standard linear models based on regression analysis when compared to previous approaches. ANN models do not assume a fixed model structure or use data distribution to describe nonlinear connections between events.

5 Benefits of ML in Software Project Management

The main concern of this paper is to improve project quality and decrease software project failure. The paper provides a comprehensive overview of the application of machine learning techniques in software project management. It contributes to the field of project management research by addressing new and old approaches to specify the complexity of software-project behavior and creating project-outcome prediction models. The study examines machine learning, software project management, and methodologies. It analyzes 5 different papers on the grasp of machine learning in software project management and categorizes them into four sections according to the methods.

The study shows that project risk assessment using machine learning is more effective in minimizing project losses, increasing the likelihood of project success, providing an alternative method for efficiently reducing project failure probabilities, increasing the output ratio for growth, and facilitating accuracy-based analysis of software fault prediction. It can help project managers become more familiar with the Project Management Knowledge Areas (PMKAs) and consider the company's conditions or project contexts before using them the basic themes in software project management can be gleaned from a variety of machine learning (ML) works. Finally, can contribute to reducing the amount of wariness and uncertainty in organizations.

6 Limitations

This paper has limitations in its narrow focus on machine learning techniques in software project management, overlooking alternative approaches for dynamic projects. It also lacks in-depth exploration and solutions for the challenges associated with implementing machine learning.

Additionally, the need for further research on the use of machine learning, its impact on project outcomes, and ethical implications is acknowledged but not thoroughly addressed. These limitations emphasize the importance of considering diverse approaches, addressing implementation challenges, and conducting comprehensive research to enhance project management practices.

7 Future Work

Future works in the field of software project management using ML techniques include exploring the use of hybrid models that combine different ML techniques to enhance SPM. This approach can potentially improve the accuracy and effectiveness of project management processes. Another area for future research is the application of ML techniques in predicting the success or failure of software projects. By analyzing historical project data and identifying patterns and factors that contribute to project outcomes, ML models can assist in the early detection of potential risks and provide recommendations for project improvement. Additionally, researchers can investigate the use of ML techniques in optimizing project scheduling and resource allocation. ML algorithms can improve project efficiency and mitigate risks by generating optimized schedules and identifying defects. It can also enhance software quality by analyzing code repositories and testing data.

Finally, the application of systems thinking principles in software project management is an interesting avenue for future research. 8 Conclusion Machine learning (ML) techniques have transformed software project management systems, offering automated solutions for effort estimation, resource allocation, risk management, and task assignment and scheduling. ML-based resource allocation techniques optimize resource utilization, leading to improved project outcomes and cost reduction. However, it is essential to acknowledge the significance of human expertise and judgment in software project management. Validating and interpreting ML results by project managers ensures effective decision-making aligned with project objectives. Further research is needed to address challenges in risk assessment and effort estimation using ML approaches. Scalable and efficient methods for handling multimedia data in real-time systems, considering resource constraints and network delays, require development. The combination of machine learning algorithms and human intelligence can be utilized to achieve maximal levels of project efficiency, precision, and success.

8 Conclusion

Machine learning (ML) techniques have transformed software project management systems, offering automated solutions for effort estimation, resource allocation, risk management, and task assignment and scheduling. ML algorithms, such as Artificial Neural Networks (ANN), Fuzzy Logic, Genetic Algorithms, and Regression Algorithms, play a vital role in enhancing project management practices. Effort estimation using ML provides accurate and timely predictions, aiding project planning and decision-making. ML-based resource allocation techniques optimize resource utilization, leading to improved project outcomes and cost reduction. Risk management benefits from ML models that identify and mitigate potential risks in software projects. ML algorithms contribute to optimized task assignment and scheduling, improving project efficiency. However, it is essential to acknowledge the significance of human expertise and judgment in software project management. Validating and interpreting ML results by project managers ensures effective decision-making aligned with project objectives. Further research is needed to address challenges in risk assessment and effort estimation using ML approaches. Scalable and efficient methods for handling multimedia data in real-time systems, considering resource constraints and network delays, require development. The integration of ML techniques empowers software project management systems, enabling organizations to optimize project outcomes. The combination of machine learning algorithms and human intelligence can be utilized to achieve maximal levels of project efficiency, precision, and success.

References

[1] Mahdi, M.N.; Mohamed Zabil, M.H.; Ahmad, A.R.; Ismail, R.; Yusoff, Y.; Cheng, L.K.; Azmi, M.S.B.M.; Natiq, H.; Happala Naidu, H. Software Project Management Using Machine Learning Technique—A Review. Appl. Sci. 2021, 11, 5183. https://doi.org/10.3390/app11115183.

[2] Gizatie Desalegn Taye1 • Yibelital Alemu Feleke2. Prediction of failures in the project management knowledge areas using a machine learning approach for software companies. https://doi.org/10.1007/s42452-022-05051-7.

[3] Z. M. Mitrović, A. M. Rakićević, D. č. Petrović, M. M. Mihić, J. Đ. Rakićević and E. T. Jelisić, "Systems Thinking in Software Projects-an Artificial Neural Network Approach," in IEEE Access, vol. 8, pp. 213619-213635, 2020, doi: 10.1109/ACCESS.2020.3040169.

[4] Pasupuleti, M. B. (2018) “The Application of Machine Learning Techniques in Software Project Management- An Examination”, ABC Journal of Advanced Research, 7(2), pp. 113–122. doi: 10.18034/abcjar.v7i2.626.

[5] Ayria Behdinian; Mohammad Amin Amani; Amir Aghsami; Khaje Nasir Toosi; Fariborz Jolai. An Integrating Machine Learning Algorithm and Simulation Method for Improving Software Project Management: A Case Study. DOI: 10.22070/JQEPO.2022.15346.1214.