International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 2, pp 42-49, February-2023

https://doi.org/10.47001/IRJIET/2023.702006

Software Risk Prediction Through the Use of Machine Learning: Review

[1]Rabab T. Mahmood, [2]Ibrahim A. Saleh

[1,2]Department of Software Engineering, Computer Science and Mathematics College, University of Mosul, Mosul, Iraq

Authors E-mail: [1]ruba.altaee1986@gmail.com, [2]i.hadedi@uomosul.edu.iq

***Abstract* - Software engineering and data science require strong programming skills. Software engineering focuses more on construction, functionality, and features, while software risk forecasting focuses more on data collection and analysis. A high level of system functionality is one of the basic needs of software development projects. One of the main characteristics that directly affects the effectiveness of software systems is the prediction of risks. Organizations can make decisions about potential solutions and improvements by using the ability to identify software systems risks through early recognition of expected failures. Inaccurate risk assessments may result in poor system performance and thus reveal its reliability. This research focuses on reviewing mechanisms for predicting early failure in software project risk assessment. Various ML machine learning techniques are used. The aim of the study is to review experience-based risk assessment models that use historical failure data from several past program projects as training data to accurately assess the risks of program initiatives. This study covers software project risk prediction models that are generally applied to all software projects throughout the software development process, helping advance the evolution of software systems.**

Keywords: Software Development Life Cycle (SDLC), Software Risk (SR), Machine Learning (ML), Risks Supervision (RM), Software Risk Management (SRM), Software Engineering (SE).

I. INTRODUCTION

The approach to software development is called software engineering [1] [2]. Unexpected events can always occur during the software life cycle (SDLC), which can lead to loss or failure of software development [3][4][5]. Incomplete and ambiguous information may lead to a large number of risks [6,7]. The most sensitive task in the SDLC is the risk prediction function which requires accurate completion [8]. To ensure project success, risk forecasting and oversight is an important stage in the development process [9][10]. No matter how much effort has been put into making sure that software projects are successful, many of them contain a high percentage of risk [11][12]. All aspects related to risk, such as cost, schedule, and quality [13], are included in the development cycle. Even if there is a single factor that has the potential to significantly affect the entire software development process, it should not be ignored [14].

Thus, a successful risk management model must be skilled in predicting risks and assessing how they will change as the project progresses [15]. Without risk management, the main risks will be ignored [16]. Because of this, risk analysis is important in SDLC, where risks are present and appropriate action is taken [17]. Moreover, given the increasing complexity of contemporary software systems, it is crucial to take appropriate steps to avoid project failure [18]. These risks may lead to project failure if not all are properly identified [2]. In SDLC, risk prediction and management should be a consistent activity, as it can save time and money if risks are predicted early and fixed early in the software development process [19].

It uses artificial intelligence (ML), which helps computers learn from experience and progress without explicit programming. In other words, the main goal of ML is to allow computers to learn independently, without the help of humans, and then adjust their behavior accordingly. In addition, machine learning makes it possible to process huge amounts of data. Stewardship of missing values in information architecture data structures has become a growing concern for others [20], which has also shown a growing interest in machine learning technology.

There have been other classification studies on regression. The methodologies and predictions for volatility forecasting as well as classification criteria are outlined in the paper [21]. Attributes that were used as indicators of literary volatility criteria were generated and prediction methods that were used to increase the accuracy of volatility risk requirements. Specifications with twists are crucial to software systems.

Because it is directly related to costs and overrun period. The SLR has been proposed in [22] as a formal repeated outcome framework. The study, like studies of other data sets, cannot determine the exact applicability of the data set arrangement. Another [21] discussed the use of machine learning techniques for program effort estimation. The systematic investigation showed the importance of ML techniques, scale measures, comparison data sets, evaluation procedures, etc.

The Fuzzy Logic Studies article [23] looked at hardware machine learning techniques to predict program risks. In addition, he discussed a variety of software projects and cost analyzes of the system business approach. The main conclusions were that the process and model should not be preferred by any other methodologies.

This study aims to conduct a comprehensive evaluation of the literature and hardware for Risk Prediction and Management Models (SRPM). Belief that the results of this study will shed light on SRPM research. A selection of the literature published within the last few years on the topic of SR prediction was examined.

II. SOFTWARE MANAGEMENT

The method involved in organizing, staffing, controlling, managing, and coordinating a product project is known as software product management (SPM) [25]. Many of the contracting supervisors today are using various SPM tools to supervise and support their project activities due to the huge number of SPM automations that are now available and rapidly evolving. Project arrangement, control and monitoring are crucial purposes of these developments. These tools offer a variety of benefits. To enhance implementation, project managers must choose an acceptable set of tools with the necessary capabilities among the various available devices [26].

Social performance management tools have been discussed extensively in the literature. Here, we present some typical tools that may be used in a variety of settings. In order to help software engineers, choose the best tool, we also introduce their features. A list of standard tools is provided below.

2.1 Software Risk Management System

Risks Management (RM) one of most significant subfields in project supervision. It is extremely important to project supervision. It eliminates the risk that could thwart the intended effects and outcomes. Small and medium-sized businesses (SME) adopt preventative actions to reduce risk [27]. SME will always require a well-planned risk supervision strategy to search for any unwanted or unanticipated errors in a project [27]. In the risk supervision process, there are five steps as shown in Figure (1) are illustrated for the Risk supervision process [28], Also Table (1) shows the SPM Tools.

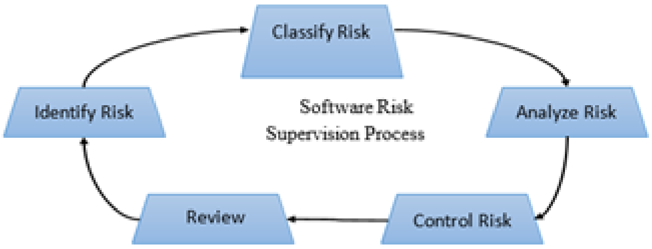

Figure 1: Risk Management process

Table 1: Software Product Management Tools [25][26][28][29]

Tool	Description
Primavera	One of the best For institution project supervision scalability, the Project Supervision Module is a complete, multi-project planning and control program built on SQL, Oracle, and SQL Server Express server databases
MS Project	Microsoft Project was created to help with project monitoring, automatically receiving and logging team member status reports and alerting the project manager if these are late or incomplete It eventually supplanted all other PC-based project supervision software
Gantt Project	Gantt Project is a cross-platform desktop application used for managing and scheduling projects. A multi-application platform for planning and project supervision is called Gantt Project. Running on any operating system, Gantt Project adheres to the task scheduling process outlined in Workflow 2, saves schedules in an XML file format that can be transported into HTML Web pages, and conducts scheduling activities by integrating with the workflow that has been implemented by the e-Hub.
Red mine	During the initial phase of the agile implementation in MIMOS, the Red mine project supervision web application solution was introduced, helping the team and capturing the requirements. All of the product backlogs are listed. A web-based supervision tool is called Red mine. Red mine lacks support for agile and scrum and is more of a general-purpose project supervision solution. User stories and tasks were submitted through a ticketing system.
Base Camp	It is a web-based application tool that can be accessed with just a web browser and an internet connection. A free one-

	active project plan with two write boards is available through Base camp, a project collaboration tool. It supports numerous projects and provides project storage space.
dotProject	It is a web-based tool for managing projects made to offer project planning and supervision capabilities. You must comprehend project supervision in order to comprehend dot Project.
Assembla	It is a web-based platform that enables employers to hire their PM candidates online. It depends on how much project work the project team needs to get done.

The Risk supervision process can be summarized in these steps.

- Determine the risk: The team's duty is to identify risks that could have an impact on the project. To do this, many strategies are utilized, the first of which is to keep a project risk register.
- Classify the risk: Various hazards are categorized based on their anticipated cost, likely impact, and likelihood of occurrence. For instance, the possibility of collecting repayments from the debtor is used to categorize credit risk.
- Analyze the risk: After identifying the risks, the next crucial stage is to examine the consequences of each risk, where the nature of the risk and how much it can influence the project's outcome are established. This data is included to the project risk registry as well.
- Control the risk: Risk analysis is followed by risk supervision. It is the process through which software companies assess risks and take steps to reduce or get rid of potential dangers. This is known as the risk control hierarchy, which states that the most effective control is always to eliminate the risk.
- Review risk control: It is done to make sure the imposed control measures are successful and efficient. It must be examined and amended to ensure that they function as intended and to identify whether any immediate corrective action is required.

2.2 Diminutions of Risk

By looking at risk sources, the risks in software evolution projects can be divided into time, money, and resource risks, or a mix of these [30]. The primary project supervision problems that could adversely affect one or more project performance characteristics in the GSD environment are project time, cost, and resource (see Figure 2).

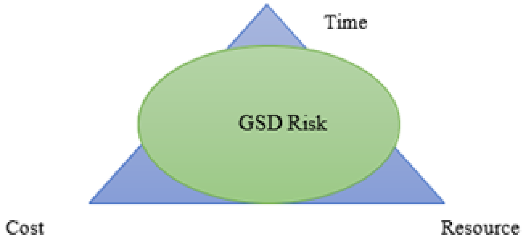

Figure 2: Three risk dimensions of GSD projects

2.3 Cost Related to Risk Factor

The project managers face two types of challenges when it comes to time-related risk factors:

- The number of adjustments needed during the project's execution.
- Time spent on unnecessary activities [31].

It is not an exaggeration to say that "the time is what defines the success of any project." By creating a clear project strategy and timeframe in advance and adhering to it, you may overcome these two difficulties easily. Effective time supervision requires a clearly defined project plan, and project managers should create such a plan and timeline to lower the two risk factors mentioned above.

2.4 Resource-Related Risk Factors

Along with time, another constraint that may be quantified is money. The cost structure for the project must be anticipated in advance as precisely as possible, just like the project's timeline [32]. Cost simply refers to the sum of money needed to start and complete a project. It is sometimes referred to as the project's budget. Knowing the cost structure in advance provides a benchmark against which to measure and track the project's real cost as it is being carried out. It enadepts the project manager to steer clear of any unanticipated expenses that might arise.

There are two different types of resources in a project: human and material. The availability of both should be taken into account by the project supervision. This restriction is strongly influenced by the pricing structure: the more money one has, the more resources one may purchase in terms of both tangible goods and higher caliber labor. The availability and accessibility issues in the market cannot be solved by money, thus a project manager should consider these issues while determining the timetadept and the pricing structure.

2.5 Risk Prediction

In light of the risk examination that includes preparation history and task attributes in the Product Improvement Life Cycle (SDLC), as shown in Figure (3), a product risk forecasting model was created. The model, known as the Atropos model, was accompanied by six main phases [33].

- Gather information through the interface.
- Program similarities.
- Store date and time accounts.
- Similarity by setting accounts.
- Suggest any potential risks.
- Risk management and verification.

The recommended approach uses the identified necessity as input into determining the degree of risk associated with the prerequisites, allowing the undertaking supervisor or area manager to prepare and reduce risks only at an earlier stage.

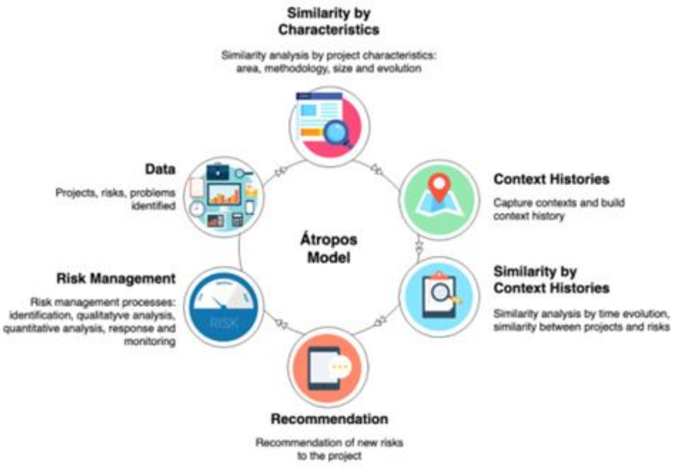
Similarity by Characteristics
Similarity analysis by project characteristics: area, methodology, size and evolution
Data
Projects, risks, problems identified
Context Histories
Capture contexts and build context history
Átropos Model
Risk Management
Risk management processes: identification, qualitatyve analysis, quantitative analysis, response and monitoring
RISK
Similarity by Context Histories
Similarity analysis by time evolution, similarity between projects and risks
Recommendation
Recommendation of new risks to the project

Figure 3: Atropos Six-Stage Model

2.6 Software Risk Evaluation

In light of the hazards encountered throughout the creation, supervision, and assessment of software, a rigorous and structured literature review was carried out. A review of the literature led to the discovery of 54 likely risk variadepts pertaining to the software evolution sector. Table (2) lists the identified risk variadepts and references.

Table 2: Fifty-four software evolution Risks

No	Risk factor	Ref.
1	Project requirements have changed	[34][35]
2	Absence of inefficient PM techniques	[36]
3	Unsuit adept task timings	[36]
4	Failing to supply resources	[36]
5	Scheduling and activity estimation errors	[37]
6	Incorrect planning	[38]
7	Low level of output	[36]
8	Insufficient supply	[35]
9	Poor quality	[39]
10	Failure to capture requirements	[39]
11	Project design that is improper	[36]
12	Inadequate system of rewards and motivation	[40]
13	Team members' lack of coordination and collaboration	[40]
14	Payment problem	[41]
15	Lack of dedication	[41]
16	Mistrust	[41]
17	Not clearly specified project milestones	[41]
18	Inadequate backing from the top	[41]
19	The project team has experienced frequent turnover.	[41]
20	A lack of specific abilities	[41]
21	Novice project manager	[41]
22	Insufficient dialogue	[41]
23	The wrong system is required	[41]
24	System requirements are unclear	[41]
25	Insufficient identification of the system requirement	[41]
26	Insufficiently inspiring attitude	[41]
27	Developing technologies	[41]
28	Changes to the organization during the project	[41]
29	Unstadept workplace situation	[41]
30	Organizational changes during the project	[42]
31	Components supplied are not sufficient	[42]
32	Including pointless features	[42]
33	Deadline sensitivity	[42]
34	Mistaken documents	[42]
35	Requirements document is not sent to the team	[25]
36	Lack of agreement on the requirement	[42]
37	Participants' diverse cultural backgrounds	[42]
38	Lack of a cooperative office setting	[42]
39	Greater number of locations	[42]
40	Governmental state	[42]
41	Social condition	[42]
42	Monetary state of the target market	[42]
43	Lack of motivation among developers	[42]
44	Absence of prior experience	[42]
45	Inadequate planning for the necessary resources	[42]
46	People develop	[42]
47	Information security issues	[42]
48	Project evolution is not sufficiently being tracked	[42]
49	Members of the evolution team with inadequate training	[42]
50	Process failure	[42]
51	Using modern technology	[42]
52	Inaccuracy in the project time estimation	[42]
53	Inadequate kill and knowledge	[42]
54	Improper control and leadership	[42]

ISSN (online): 2581-3048
Volume 7, Issue 2, pp 42-49, February-2023
https://doi.org/10.47001/IRJIET/2023.702006

III. RELATED WORKS

Several studies dealt with the issue of software risks and their prediction, as follows:

In *2019 Erdoğan* investigated five different strategies for (DT) with dataset of 356 sample driven credit approach to risk assessment and discovered that the RF model used to have the top rating accuracy. Larger data sets with more organizations should be used to verify our work in order to increase the prediction system's effectiveness with maximum accuracy of 99.16%[43].

In *2020 Sousa et. al*., This overview gives a general understanding of the various things relating to risk and RM in software systems, such as the conventional methodologies used to identify and manage hazards in software companies as well as the ML techniques and methods applied to provide better estimates of relative risks and risk factors that can be experienced during growth [44].

In *2020 Akumba et. al*., introduce the NB Classifier to foretell dangers that will arise in some software projects' requirement analysis phase of the (SDLC). The most crucial factors in forecasting individual risk across a task are priority and likelihood, it should be noted. If both the risk likelihood and the priority are known in advance, they can prevent the failure of the entire project. However, if it is ignored, it will be terrible as projects won't be completed on time and will have less usefulness [45].

In *2021 Iftikhar et. al*., research was done on predicting the risks associated with time, money, and resources that distributed teams confront in a global software development environment. Using neural network techniques like Liebenberg-Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient to forecast mitigation plans relating to project duration, cost, and required resources in international software development the study's findings showed that Bayesian regularization outperformed the Logistic Regression Method and Backpropagation Gradient in terms of the MSE (Validation) criterion. It is reasoned that Levenberg-Marquardt gives the best presentation since it gives a genuinely precise expectation of the basic temperature for superconductors which the creators performed by plotting application-inferred SOM (self-sorting out maps) on the dataset through fluffy brain organizations. [46].

In *2021 Filippetto et. al*., provides a computational approach for lowering project failure likelihood through risk forecasting. The study's goal is to present a methodology to help teamsidentify and keep track of risks at various stages of a project's life cycle. The study argues that using context histories to infer risk recommendations for new initiatives is a scientific advancement [7].

In *2021 Gouthaman et. al.*, investigates the main software paradigms employed in the sector through dialogue with professionals in the field and a literature review. In this study, the researchers provided a new framework in which machine learning classifiers were applied to the data set obtained through the survey and risk estimates were made for each of the chosen software models. Software product managers can choose the suitable programming model based on the software needs and predictive percentage using this score [47].

In *2022 Hassan et. al.*, Developed and evaluated risk forecasting model using ML classification techniques such as LR, DT, RF, SVM, KNN, and NB. Worked on determining the most significant software design variables influencing the overall success or failure of a software project. The best outcomes, according to experiments, come from RF and LR models [48].

In *2022 Khalid et. al*., Through the use of methodologies and the creation of ML algorithms, financial ratios are utilized to effectively identify risks and automating company's risk management. The Python application was used to collect, analyses, and test annual data for non-financial enterprises in Pakistan from 2006 to 2020. The findings demonstrate that artificial intelligence methods can anticipate hazards with the least amount of error possible, and among all the methodologies employed, the RF method surpasses the others [49].

IV. CONCLUSION

One of the flourishing phases of the (SDLC) is necessity designing, and hazard examination is one more fundamental part of the SDLC. There are different dangers that should be considered while creating programming. As well as recognizing gambles, risk investigation's fundamental objective is to really try to oversee them. Moreover, it could incorporate exact insights about dangers and give thoughts for diminishing them. Precisely recognizing dangers is the principal objective of chance examination. Basic components including issue definition, plan, and information gathering ought to be remembered for risk investigation. Indeed, even one gamble variable can prompt framework disappointment on the off chance that the gamble examination isn't done accurately. SRs keep on introducing issues for clients, and specialists have proposed various systems for expecting and staying away from SRs. The motivation behind this study is to lead a precise writing survey (SLR) and assess the nature of the exploration writing that has proactively been distributed on SR expectation models (SRPM).ML techniques offer a reliadept detection capacity for SR forecasting. Performance measurements show that the majority of ML-based research performs remarkably well. Overall, we find that the SRPM literature lacks high-quality work and that the methodology employed to conduct forecast SR detection investigations is inconsistent.

ACKNOWLEDGEMENT

Authors would like to thank the University of Mosul in Iraq for providing Moral support.

REFERENCES

[1] A.Salehnia and H. Pournaghshband, “Ethical Issues in Software Engineering Revisited,” Ethical Issues Inf. Syst., pp. 31–52, 2011, doi: 10.4018/9781931777155.ch010.

[2] M. H. Mahmud, M. T. H. Nayan, D. M. N. A. Ashir, and M. A. Kabir, “Software Risk Prediction: Systematic Literature Review on Machine Learning Techniques,” Appl. Sci., vol. 12, no. 22, 2022, doi: 10.3390/app122211694.

[3] S. Saadah and G. S. Wulandari, “Information Science and Applications,” Lect. Notes Electr. Eng., vol. 339, pp. 777–784, 2015, doi: 10.1007/978-3-662-46578-3.

[4] Z. S. Shaukat, R. Naseem, and M. Zubair, “A dataset for software requirements risk prediction,” Proc. - 21st IEEE Int. Conf. Comput. Sci. Eng. CSE 2018, pp. 112–118, 2018, doi: 10.1109/CSE.2018.00022.

[5] M. A. Teklemariam and E. Mnkandla, “Software project risk management practice in Ethiopia,” Electron. J. Inf. Syst. Dev. Ctries., vol. 79, no. 1, pp. 1–14, 2017, doi: 10.1002/j.1681- 4835.2017.tb00583.x.

[6] S. N. Bhukya and S. Pabboju, “Software engineering: risk features in requirement engineering,” Cluster Comput., vol. 22, pp. 14789–14801, 2019, doi: 10.1007/s10586-018-2417-3.

[7] A.S. Filippetto, R. Lima, and J. L. V. Barbosa, “A risk prediction model for software project management based on similarity analysis of context histories,” Inf. Softw. Technol., vol. 131, no. March 2020, 2021, doi: 10.1016/j.infsof.2020.106497.

[8] R. Naseem et al., “Empirical assessment of machine learning techniques for software requirements risk prediction,” Electron., vol. 10, no. 2, pp. 1–19, 2021, doi: 10.3390/electronics10020168.

[9] J. McManus, “Risk management in software development projects,” Risk Manag. Softw. Dev. Proj., vol. 8, no. 2, pp. 1–172, 2012, doi: 10.4324/9780080498089.

[10] E. E. Odzaly, D. Greer, and D. Stewart, “Agile risk management using software agents,” J. Ambient Intell. Humaniz. Comput., vol. 9, no. 3, pp. 823–841, 2018, doi: 10.1007/s12652-017-0488-2.

[11] I.M. Alharbi, A. A. Alyoubi, M. Altuwairiqi, and M. A. Ellatif, “Enhance Risks Management of Software Development Projects in Concurrent Multi-Projects Environment to Optimize Resources Allocation Decisions,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, pp. 236–246, 2021, doi: 10.14569/IJACSA.2021.0120626.

[12] A.Elzamly and B. Hussin, “Classification and Identification of Risk Management Techniques for Mitigating Risks with Factor Analysis Technique in Software Risk Management,” Rev. Comput. Eng. Res., vol. 2, no. 1, pp. 22–38, 2015, doi: 10.18488/journal.76/2015.2.1/76.1.22.38.

[13] N. Chepa, B. A. Jn, R. N. Haizan, and M. A. Az, “A Review on Risk Mitigation of IT Governance,” Inf. Technol. J., vol. 14, no. 1, pp. 1–9, 2015, doi: 10.3923/itj.2015.1.9.

[14] C. Kumar and D. K. Yadav, “A Probabilistic Software Risk Assessment and Estimation Model for Software Projects,” Procedia - Procedia Comput. Sci., vol. 54, pp. 353–361, 2015, doi: 10.1016/j.procs.2015.06.041.

[15] A.S. Khatavakhotan and S. H. Ow, “DEVELOPMENT OF A SOFTWARE RISK MANAGEMENT MODEL USING UNIQUE FEATURES,” Malaysian J. Comput. Sci., vol. 28, no. 2, pp. 110–131, 2015.

[16] B. G. Tavares and C. Eduardo, “Practices to Improve Risk Management in Agile Projects,” Int. J. Softw. Eng. Knowl. Eng., vol. 29, no. 3, pp. 381–399, 2019, doi: 10.1142/S0218194019500165.

[17] M. Asif and J. Ahmed, “A Novel Case Base Reasoning and Frequent Pattern Based Decision Support System for Mitigating Software Risk Factors,” IEEE Access, vol. 8, 2020.

[18] S. Li and S. Duo, “Safety analysis of software requirements : model and process,” Procedia Eng., vol. 80, pp. 153–164, 2014, doi: 10.1016/j.proeng.2014.09.071.

[19] R. K. Bhujang and V. Suma, “A comprehensive solution for risk management in software development projects,” Int. J. Intell. Syst. Technol. Appl., vol. 17, 2018.

[20] J. Takamatsu, “2015 16th IEEE / ACIS International Conference on Software Engineering , Artificial Intelligence , Networking and Parallel / Distributed Computing,” no. June, 2015.

[21] A.M. Alsalemi and E. Yeoh, “A Systematic Literature Review of Requirements Volatility Prediction,” 2017 Int. Conf. Curr. Trends Comput. Electr. Electron. Commun., pp. 55–64, 2017.

[22] S. P. Pillai, “Consolidating Evidence Based Studies in Software Cost / Effort Estimation – A Tertiary Study,” IEEE Reg. 10 Conf., pp. 833–838, 2017.

[23] M. N. Mahdi et al., “Software Project Management Using Machine Learning Technique—A Review Mohammed,” Appl. Sci. Tech., 2021.

[24] M. Sajad and M. Sadiq, “Software Project Management : Tools assessment , Comparison and suggestions Software Project Management : Tools assessment , Comparison and suggestions for future development .,” Int. J. Comput. Sci. Netw. Secur., no. February, 2016.

[25] M. Sajad, M. Sadiq, K. Naveed, and M. S. Iqbal, “Software Project Management : Tools assessment, Comparison and suggestions for future development.,” Int. J. Comput. Sci. Netw. Secur., vol. 16, no. 1, pp. 31–42, 2016.

[26] M. Sajad and M. Sadiq, “Software Project Management : Tools assessment, Comparison and suggestions Software Project Management : Tools assessment , Comparison and suggestions for future development .,” Int. J. Comput. Sci. Netw. Secur., no. February, 2016.

[27] K. Vichova, P. Taraba, and T. Belantova, “Risk Management of the Project and the Use of Software in SME,” WSEAS Trans. Bus. Econ., vol. 17, pp. 551–559, 2020, doi: 10.37394/23207.2020.17.54.

[28] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, “estimation using different Artificial Neural Network architectures and Taguchi Orthogonal Arrays,” IEEE Access, vol. 4, pp. 1–12, 2021, doi: 10.1109/ACCESS.2021.3057807.

[29] C. Muriana and G. Vizzini, “ScienceDirect Project risk management : A deterministic quantitative technique for assessment and mitigation,” Int. J. Proj. Manag., pp. 1–3, 2017.

[30] N. Ghatasheh, H. Faris, I. Aljarah, and R. M. H. Al-sayyed, “Optimizing Software Effort Estimation Models Using Firefly Algorithm,” J. Softw. Eng. Appl., no. March, pp. 133–142, 2015.

[31] C. Muriana and G. Vizzini, “ScienceDirect Project risk management : A deterministic quantitative technique for assessment and mitigation,” Int. J. Proj. Manag., pp. 1–3, 2017.

[32] T. Karimi, M. Reza, and Y. Yahyazade, “Developing a Risk Management Model for Banking Software Development Projects Based on Fuzzy Inference System,” J. Optim. Ind. Eng., vol. 13, no. 2, pp. 267–278, 2020, doi: 10.22094/JOIE.2020.1.

[33] M. H. Mahmud, M. T. H. Nayan, D. M. N. A. Ashir, and M. A. Kabir, “Software Risk Prediction: Systematic Literature Review on Machine Learning Techniques,” Appl. Sci., vol. 12, no. 22, 2022, doi: 10.3390/app122211694.

[34] M. R. Hosseini and N. Chileshe, “A Discourse on the Concept of Virtuality in Globally Dispersed Teams,” 2015.

[35] J. Pajares and A. L, “Computers & Industrial Engineering On the project risk baseline : Integrating aleatory uncertainty into project scheduling,” ELSEVIER, vol. 160, 2021, doi: 10.1016/j.cie.2021.107537.

[36] A.Nieto-morote and F. Ruz-vila, “A fuzzy approach to construction project risk assessment,” Int. J. Proj. Manag., vol. 29, no. 2, pp. 220–231, 2011, doi: 10.1016/j.ijproman.2010.02.002.

[37] B. Baharuddin, M. P. Unit, and M. M. Yusof, “Evaluation of Risk Management Practices in Information Systems Project in the Public Sector (,” ASEAN Cit. Index MyCite Indexes Eval., vol. 53, 2018.

[38] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi, “Risks and risk mitigation in global software development : A tertiary study,” Inf. Softw. Technol., vol. 56, no. 1, pp. 54–78, 2014, doi: 10.1016/j.infsof.2013.06.005.

[39] S. Y. Chadli, A. Idri, and L. Fern, “Identifying Risks of Software Project Management in Global Software Development : An Integrative Framework,” IEEE, 2016.

[40] A.Iftikhar, S. M. Ali, M. Alam, S. Musa, and M. Mohd, “Analysis of Risk Factors in Global Software Development : A Cross-Continental Study Using Modified Firefly Algorithm,” Comput. Intell. Neurosci., vol. 2022, 2022.

[41] N. Bhatia, “Fuzzy Cognitive Map based Approach for Software Quality Risk Analysis,” ACM SIGSOFT Softw. Eng. Notes, vol. 36, no. 6, 2011, doi: 10.1145/2047414.2047422.

[42] J. M. Verner, O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi, “Risks and risk mitigation in global software development : A tertiary study,” ELSEVIER, vol. 56, pp. 54–78, 2014.

[43] O. Erdoğan, “Machine Learning Algorithms Based Credit Risk Assessment Models,” Int. Conf. Accounting, Business, Econ. Polit., pp. 354–359, 2019, doi: 10.23918/icabep2019p40.

[44] A.Sousa, J. P. Faria, and J. Mendes-Moreira, “An analysis of the state of the art of machine learning for risk assessment in software projects,” Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE, vol. 2021- July, pp. 217–222, 2020, doi: 10.18293/SEKE2021-097.

[45] B. O. Akumba, S. U. Otor, I. Agaji, and B. T. Akumba, “A Predictive Risk Model for Software Projects’

Requirement Gathering Phase," Int. J. Innov. Sci. Res. Technol., vol. 5, no. 6, pp. 231– 236, 2020, doi: 10.38124/ijisrt20jun066.

[46] A.Iftikhar, M. Alam, R. Ahmed, S. Musa, and M. M. Su'Ud, "Risk Prediction by Using Artificial Neural Network in Global Software Development," Comput. Intell. Neurosci., vol. 2021, 2021, doi: 10.1155/2021/2922728.

[47] G. P and S. Sankaranarayanan, "Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers," Comput. Electr. Eng., vol. 94, no. October 2020, p. 107362, 2021, doi: 10.1016/j.compeleceng.2021.107362.

[48] H. Hassan, M. A. Abdel-Fattah, and A. Ghoneim, "Risk Prediction Applied to Global Software Development using Machine Learning Methods," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 9, pp. 111–120, 2022, doi: 10.14569/IJACSA.2022.0130913.

[49] S. Khalid et al., "Predicting Risk through Artificial Intelligence Based on Machine Learning Algorithms: A Case of Pakistani Nonfinancial Firms," Complexity, vol. 2022, no. November 2019, 2022, doi: 10.1155/2022/6858916.

AUTHORS BIOGRAPHY

Rabab T. Mahmood, Master Student in Department of Software Engineering, Computer Science and Mathematics Collage, University of Mosul, Mosul, Iraq

Ibrahim A. Saleh, Assistant Professor in Department of Software Engineering, Computer Science and Mathematics Collage, University of Mosul, Mosul, Iraq

Citation of this Article:

Rabab T. Mahmood, Ibrahim A. Saleh, "Software Risk Prediction Through the Use of Machine Learning: Review" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 2, pp 42-49, February 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.702006

****