A Comparative Analysis of Machine Learning Techniques for Software Risk Assessment

Esha Khanna
Department of Computer Engineering, *J. C. Bose University of Science and Technology, YMCA,* Faridabad, Haryana, India, eshakhanna30@gmail.com

Rashmi Popli
Department of Computer Engineering, *J. C. Bose University of Science, and Technology, YMCA,* Faridabad, Haryana, India, rashmimukhija@jcboseust.ac.in

Naresh Chauhan
Department of Computer Engineering, *J. C. Bose University of Science, and Technology, YMCA,* Faridabad, Haryana, India, nareshchauhan19@gmail.com

Abstract

Risks in software development affects the success of a project. Software risk management reduces the probability of software failure. Software risk assessment is a process of identifying an upcoming risk, analyzing its impact and prioritizing it. Risk management is a tedious task and can be automated using machine learning algorithms. Machine learning algorithms are able to imitate the intelligence of human brain in order to perform the tasks of classification and regression. This paper aims to study the role of machine learning algorithms to predict the software risks. The work reviews the literature and present a comparative analysis of machine learning algorithms for software risk prediction. The paper further presents the current challenges in the field of software risk management that needs to be addressed.

Keywords— Software Risk Prediction, Software Risk Management, Machine Learning, Distributed Agile software development

DOI Number: 10.14704/nq.2022.20.9.NQ44382 **Neuro Quantology 2022; 20(9):3309-3315**

I. Introduction

Risks are unwanted events that negatively impact the success of project by exceeding time and cost and reducing the overall quality [1]. During the development, Software projects are impacted by various software risks including technical risks, management risks, financial risks, personnel risks and resources risks [2]. Software risk management (SRM) is one of the umbrella activities that ensure success of software within scheduled time and budget. According to Bohem, SRM is a task of identifying and dealing software risks in early stages of SDLC so as to minimize its impact on project success[3]. Risk assessment and Risk Control are two pillars of SRM. Risk assessment deals with the process of identifying the risk , analyzing its impact and prioritizing them so that most critical risks are dealt well on time. Manual risk assessment is a lengthy process. However, this can be automated using machine learning algorithms. Machine learning algorithms mimics the human brain to solve the complex tasks of classification, clustering and regression. In the previous years, machine learning algorithms has paved its way to software engineering and are proven to improve the tasks of development and testing.

This paper examines the role of machine learning algorithms in software risk management. During the literature review it has been observed that various machine learning algorithms has been implemented to predict and assess software development risks. This paper presents a comparative analysis of machine learning algorithms for software risk prediction. The goal of the paper are as follows.

- To carry out a literature review and present the comparative analysis of different machine learning algorithms implemented for software risk prediction.

- To present the current challenges in Software risk prediction.

The work has been organized as follows:
Section II presents the comparative analysis of various machine learning algorithms used for software risk prediction. Section III presents the current challenges and section IV concludes.

II. LITERATURE REVIEW

A literature review has been carried out in order to find gaps in the current literature. Databases explored included IEEE Xplore, Springer, Wiley, Science Direct and Elsevier. Summary of some of the important studies has been presented in this section. Table 1 presents the comparative analysis of machine learning algorithm used for software risk prediction. The table summarizes the algorithm used, inputs, outputs, accuracy and limitations of the studies.

III. CURRENT CHALLENGES

Software risk prediction is an important task in software risk management that enables the team members to analyze the upcoming problems and to timely react to the unwanted circumstances to achieve the success. However, the lack of importance of risk management by project managers hinders the success of the project [17]. It has been observed that industrial risk management practices are not up to the mark of recommended risk management best practices.

Table 1: Comparative Analysis of Machine Learning Algorithms in Software Risk Prediction

Author	Year	Algorithm	Input	Accuracy	output	Limitations
Asim Iftikhar et. al.[4]	2021	Levenberg Marquart, Bayesian regularization andScaled Conjugate Gradient	273 responses of 33 questions. Questionnaire collected by purposive sampling.	Bayesian Regularization outperformed others. MSE (Validation) LM:16.80 BR: 10.254 SCG: 15.16 MSE(Testing) LM:14.40 BR: 19.55 SCG:25.04	Risk level 0-3	Convenience sampling has been used that prevents the researchers to generalize the results.

B. O. Akumba et. al.[5]	2020	Naïve Bayes Classifier (Supervised machine learning technique)	299 data instances with 12 variable attributes and one target output.	98% confidence interval 95% Kappa	Five risk levels i.e. Insignificant, Low, Moderate, high or catastrophic	Only considered risks in software requirement gathering phase of SDLC
M. H. Calp and M. A. Akcayol [6]	2020	Neural Network	45 risk factors extracted from 20 software projects	MSE2 =0.001 Training R1= 0.9978 Testing R=0.9935 Validation R=0.996	Deflection in project cost and duration, number of personnel and project success.	The work does not consider all software development life cycle risks
M. A. Ibraigheeth, S. A. Fadzli [7]	2019	Fuzzy LogicToolbox in MATLAB	10 success factors identified by Standish organization (Survey 50,000 projects)	Proposed work applied on 8 virtual projects	Low, Medium, High Risks	Correlation between 10 identified factors have not been considered. Further, the work has not been applied to actual software projects.
M. Zavvar et. al. [8]	2017	Support Vector Machine	Data set of 530 samples out of which 70% used for training and 30% for testing	Accuracy of 99.51%	Risks classified as low or high	The work does not expose software development life cycle methodology.
Changkyun Jeon[9]	2015	Markovchain; Software repository data was used to propose a risk prediction model	Data from software repository	Accuracy of 69%	Risk levels	The work is dependent on the experience and analysis of the tester. Further work is not applicable in the scenario of any changes in team.

C. Kumar and D.K. Yadav[10]	2015	Bayesian Classifier	27 risk factors extracted from 12 software projects	MMRE2=0.03842	Low, medium or high risk projects	Does not consider project management risks, third party risks. Does not expose software development lifecycle model.
T. Christiansen et. al. [11]	2015	Multiple Logistic Regression	Questionnaire used to extract data from 70 software projects	Accuracy of 90%	Projects classified as risky or non risky	The work has not specified the software development model.
W. M. Han [12]	2015	Neural Network Logistic Regression	22 attributes extracted from 40 projects from OMRON database.	Neural Network= 82.2% Logistic Regression= 87.5%	Software projects classified as risky or non risky	OMRON dataset does not expose project attributes such as software development methodology and size of project.
Y. Hu. et. al.[13]	2013	Bayesian networkwith causality constraints Decision Trees Naïve Bayes Bayesian Classifiers	Data extracted using questionnaire from 302 projects	Bayesian network with causality constraints= 75.15% Decision Trees= 70.86% Naïve Bayes=72.85% Bayesian Classifiers=74.17%	Risks classified as low or high	Small sample size that causes the network to detect subset of underlying causalities.
Ekananta Manalif [14]	2012	Fuzzy logic Combined with Expert COCOMO	The proposed work has been validated with 3	Higher correlation in software size and actual effort for all three data sets.	Risk levels	Does not considered all life cycle risks.

			industrial data sets			
Y. Hu et. al. [15]	2009	Risk prediction model using Machine Learning Algorithm (ANN and SVM)	64 inputs with 8 sigmoidal hidden units	SVM resulted accuracy 85% and outperformed ANN (accuracy 75%)	1 sigmoidal output unit	SVM and ANN both have deficiency of imperceptibility and therefore it is difficult to explicitly trace and imitate the prediction results.
Y. Hu, et. al. [16]	2007	Neural Network Genetic AlgorithmNN Support VectorMachine	120 real software projects data gathered using questioners	NN=70% Genetic Algorithm NN= 85% SVM= 80%	Projects classified as successful, Challenged or failed.	Purposive sampling used

Due to COVID 19 pandemic, Distributed Agile Software Development (DASD) is the most widely practiced software development life cycle today. As DASD combines the benefits of Agile software development and Distributed software development, it also brings along various risk factors that arises due to contradictory principles of ASD and DSD. ASD focuses on faster development, frequent collaborations and daily meetings, while DSD exploits the development cost by outsourcing the tasks to 24* 7 talented team members that works from different global locations. Several risk factors related to DASD are identified and classified in previous studies [18,19,20]. But it has been observed that no work has used machinelearning algorithms in order to prioritize the DASD risk factors. The studies in table 1 does not consider risk factors associated to DASD. Hence, there is a need to automate risk management in DASD so that most severe risks are managed well on time[21].

IV. CONCLUSION

Software risk assessment is a process of identifying the upcoming risks, analyzing its impact and prioritizing them. Manual assessment of software risks is a lengthy and tedious process. Therefore, machine learning algorithms are used to automate this process. Different machine learning algorithms like Naïve Bayes Classifier, Bayesian Classifiers, Fuzzy logic, Levenberg Marquart, Scaled Conjugate Gradient, ANN, Support Vector Machine has been implemented to predict the software risks. This paper presents a comparative analysis of these techniques on the basis of accuracy. The study reveals that Support vector machine has an accuracy of 99% [8].

However, no study has taken DASD risk factors into account. After covid 19 pandemic, IT industry is moving towards DASD model for software development, which further causes several DASD associated risk factors, that may hinder the success of software. Several studies in literature has identified DASD risk factors and have classified them so as to group related risk factors together. However, the works lacks in assessing and prioritizing the DASD risk factors such that most crucial risks are addressed well ontime.

References

[1] S. M. Arıkan and Ö. Yürekten, "Software Risk Management Process Improvement Experience in Enterprise Projects," Turkish National Software Engineering Symposium (UYMS), pp. 1-5, doi: 10.1109/UYMS50627.2020.9247026, 2020.

[2] Andre Sousa, et. al., "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, 2021.

[3] B. W. Boehm, "Software risk management: principles and practices," in IEEE Software, vol. 8, no. 1, pp. 32-41, doi: 10.1109/52.62930,1991.

[4] Asim Iftikhar, et. al. , "Risk Prediction by Using Artificial Neural Network in Global Software Development" Computational Intelligence and Neuroscience, vol. 2021, Article ID 2922728, 25 pages, 2021.

[5] B. O. Akumba, et. al. " A Predictive Risk Model for Software Projects' Requirement Gathering Phase". International Journal of Innovative Science and Research Technology. 5. 231-236. 10.38124/IJISRT20JUN066,2020.

[6] M. H. Calp et. al., "A novel model for risk estimation in software projects using artificial neural network," Artificial Intelligence and Applied Mathematics in Engineering Problems, D. J. Hemanth and U. Kose, Eds. Cham: Springer International Publishing, , pp. 295–319., 2020.

[7] M. A. Ibraigheeth et. al., "Fuzzy Logic Driven Expert System for the Assessment of Software Projects Risk" International Journal of Advanced Computer Science and Applications(IJACSA), 10(2), 2019.

[8] M. Zavvar, et. al. "Classification of risk in software development projects using support vector machine," Journal of Telecommunication, Electronic and Computer Engineering, vol. 9, pp. 1–5, 2017.

[9] Chang-Kyun Jeon, et al. "Probabilistic Approach to Predicting Risk in Software Projects Using Software Repository Data." Int. J. Softw. Eng. Knowl. Eng. 25 , 1017-1032, 2015.

[10] C. Kumar, et. al.. "A Probabilistic Software Risk Assessment and Estimation Model for Software Projects." Procedia Computer Science 54 pp- 353-361, 2015.

[11] T. Christiansen, et. al., "Prediction of risk factors of software development project by using multiple logistic regression," ARPN Journal of Engineering and Applied Sciences, vol. 10, pp. 1324–1331, 01 2015.

[12] W.-M. Han, "Discriminating risky software project using neural networks," Computer Standards & Interfaces, vol. 40, pp. 15 – 22, 2015.

[13] Y. Hu, et. al., "Software project risk analysis using Bayesian networks with causality constraints" Decision Support Systems,Volume 56, pp- 439-449, 2013.

[14] Manalif, Ekananta et al. "Fuzzy-ExCOM Software Project Risk Assessment." 11th International Conference on Machine Learning and Applications 2, 2012.

[15] Y. Hu, et. al. , "An Intelligent Model for Software Project Risk Prediction". International Conference on Information Management, Innovation Management and Industrial Engineering, pp. 629-632, doi: 10.1109/ICIII.2009.157, 2009.

[16] Y. Hu, et. al. , "Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches," Third International Conference on Natural Computation , pp. 358-362, doi: 10.1109/ICNC.2007.672, 2007.

[17] Wan S, et. al., "Risk Management framework for distributed software team- A case study of Telecommunication company". Procidia Computer science, Science Direct, 2019.

[18] E. Mohammad et. al., "A risk management framework for distributed scrum using PRINCE2 methodology," Bulletin of Electrical and Engineering Informatics, vol. 9, issue 3, pp. 1299-1310, 2020.

[19] V. Suprika and R. Urvashi, "Categorization of risk factors for distributed agile projects," Information and software technology, Elsevier, pp. 1-15, 2014.

[20] S. Wan, H. Wan and A. Arzi, "Risk Management framework for distributed software team- A case study of Telecommunication company," Procidia Computer science, Science Direct, 2019.

[21] E. Khanna, R. Popli and N. Chauhan, "Artificial Intelligence based Risk Management Framework for Distributed Agile Software Development," 8th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 657-660, doi: 10.1109/SPIN52536.2021.9566000, 2021.