Software Project Management Using Machine Learning Technique - A Review

Mohamed Zabil M.H[1], Mohamed Najah Mahdi[2], Muhammad Sufyian Mohd Azmi[1], Lim Kok Cheng[1], Azlan Yusof[1],Abdul Rahim Ahmad [1]
{hazli, najah.mahdi, Sufyian, kokcheng, azlany, abdrahim}@uniten.edu.my
[1]College of Computing and Informatics, Universiti Tenaga Nasional
[2]Insitute of Informatics and Computing in Energy, Universiti Tenaga Nasional

Abstract— **Project management planning assessment is of great significance in project performance activities. The creation of project management cannot be effectively handled without a practical and rational strategy. This paper offers a large-scale review analysis of articles based on machine learning and risk evaluation management for software projects. The reviews are presented and classified into two groups. The first group covers project management analysis and survey articles. The second group contains works on the steps and experimental criteria that are widely used in the management of machine learning projects. The paper provides a deeper insight and an important framework for future work in the project risk assessment, highlights the estimation of project risk using machine-learning is more efficient in reducing the project's fault and provides a further way to reduce the probability chances effectively and to increase the software development performance ratio.**

Keywords— Software Project Management, formatting, Machine Learning Technique, Assessment, Risk.

I. INTRODUCTION

The challenges that organizations face in dealing with software projects are the improvement of the project performance and to ensure project success. The risk of project failure is typically due to a lack of information, expertise, instruments, and techniques in the implementation of projects[1]. The information saved in the projects' historical datasets can be used to build predictive models using either a statistical approach such as linear regression and correlation analysis or machine learning (ML) methods such as Artificial Neural Networks (ANN) and Support Vector Machine (SMV). Predictive approaches is a method to predict the future of the project, based on current or past project data. Due to large number of ML algorithms, many ML algorithms have still not been analyzed. Rationale for choosing software project management (SPM), the problems of project management assessment, and the production ML methodology are discussed according to the literature's findings. The results of the analysis are then analyzed.

Although the definition of project success and failure has been defined in literature, there are long disagreements about how to assess project progress. The views on what constitutes project success and how project success is measured[2] are different. Hughes, et al. [3], and PMI [4] differentiate between the performance of the project and the performance factors of the assessment of performance or failure, while progress factors for projects are considered inputs contributing to performance in projects. A good project is historically characterized in terms of delivery, within an agreed timeline, of the desired results, and using the chosen resources[5]. PMI [4] recognizes projects that are effective in meeting the project objectives, criteria, and aspirations of the stakeholder. The results of the classical objective outcome measures, such as project cost (below and on or above budget), length of the project (early, on or late), and the quality of the outcome of the project (with less or better than the necessary properties and functions), are defined by researchers, such as Aladwani [6], Cates and Mollaghasemi [7], Parsons [8] and Rosenfeld [9].

The criteria for project evaluation leads to cost and time spending, unmet expectations, or even canceled projects being a normal and unavoidable project risk with adverse effects for software-project reliability[10]. One of the principal factors posing challenges to software projects are the criteria for modification of the specifications (in terms of several added, removed, and modified) during software project development[11].

This paper is further organized as follows: Section II characterizes the study research questions. This is to identify and classify the methods and predictors for requirements of project management planning assessment, the taxonomy of literature on software project risk assessment using machine learning technique focus in search for every article related in three databases articles and divided into two classes. Section III discussed and described motivation in this field of research. Subsequently, a new view of the software project risk assessment using machine learning technique is highlighted. Finally, a conclusion is presented in Section IV.

II. RESEARCH QUESTIONS

Recently, numerous studies identify possible solutions in project management planning assessment to its concomitant risks. Many studies have focused on the important of requirements project management planning assessment but inadequately covered its effects and solutions. Thus, the main objectives of this paper are to identify and to classify the methods and predictors for requirements project management planning assessment. In particular, studies on descovering the current attributes that were used as predictors of requirements project management planning assessment in the literature and the prediction techniques that were used to improve the accuracy of predicting requirements. Thus, we intended to understand how traditional project management planning copes with the problem of requirements project assessment methods they utilize. Two research questions were formulated to understand the status:

RQ1. What is the current studies and research on project management planning assessment using ML techniques?

RQ2. How effectively and efficiently can these techniques be applied?

A. Review and Survey Articles

The review and study papers summarize the current state of understanding of ML technology in a software project risk and what ML algorithms are used to predict software

978-1-7281-7310-8/20/$31.00 ©2020 IEEE

development requirements. The table below contents two approaches (see Table I):

1) *Studies Conducted on Machine Learning Methods*

This section reviews the ML methods and how it's used. These articles have been divided into different subjects and applications. Based on the ML methods of software development techniques and risk management, selected research has been categorized into broad categories.

2) *Other Methods*

This section reviews the other methods and how it's used. Such papers have been separated into various topics and implementations. Based on the approaches of software engineering and risk management, selected works are categorized into specific categories.

TABLE I. THE REVIEW AND STUDY PAPERS SUMMARIZE THE STUDIES CONDUCTED ON MACHINE LEARNING AND OTHER METHODS

No	References	Approach	Method used	Studies Conducted on	Contribution
1	Idri, et al. [12]	Machine Learning Methods	K-nearest Neighbor Algorithm (KNN)	management of missing values (MV) in data systems of software engineering (SE)	(1) A classification schedule that categorizes articles in the field of SE-database MV-techniques; (2) A systematic MV-technique mapping study in SE, the structuring of relevant research work over the last 10 years with the review of 35 papers selected; (3) Review of the demographic trends in MV-technology in SE-database;
2	Alsalemi and Yeoh [13]		Not mentioned	methods and predictors for volatility prediction and the needs to classify them	Aims to respond to four research questions: 1) how does the volatility requirements prediction apply to different methods of software development? 2) What is the reliability of software engineering specifications dependent on ML? 3) What characteristics (predictors) are used to quantify app engineering criteria volatility? 4) What are the efficiency measures for the evaluation of current models?
3	Pillai, et al. [14]		Not mentioned	proposes SLR support a structured process for repeatable results.	The research cannot agree on the exactness of the usage of a data set relative to other data sets within the organization. In conjunction with comparison approaches, ML strategies such as FL and GA obtain more precise estimates.
4	Sharma and Singh [15]		Not mentioned	present a debate on ML-inspired software effort estimates for research trends.	The findings of the systematic analysis revealed influential patterns of ML techniques, scale measures, comparison data sets, evaluation processes, etc.
5	Shivhare and Rath [16]		ANN, fuzzy logic, analogy estimation	estimation of software project effort	The author explored how the ML methods for software effort estimation were used. And he provided a summary of a variety of software effort, cost-size estimation strategies focused on system functions and the key result was that no process and model should be favored to other methods.
6	Stewart, et al. [17]	Other Methods	reviewed existing approaches for assessing network infrastructure (CI) financial returns on investment (ROI)	The ROI financial concept is challenging about a service that does not produce a 'sold amount' as it would be found in the purchase and sale of stocks in most academic environments.	The goal of this study is less to direct the assessment of the CI's usefulness for scientific analysis than it can be a starting point for more analysis. This reasoning paradigm contributes to a strong difference between what we do and how we do it.
7	Malhotra and Chug [18]		Not mentioned	performed a systematic analysis of current software-maintenance studies.	According to the findings of the study the usage of ML algorithms
8	García, et al. [19]		Logic Model	explores schools and technical resources for initiatives and open access applications for the application of artificial intelligence strategies in recent decades.	The contribution of this study is connected to the projected need to create new project management models and IT techniques combining ML-based strategies and the handling of information imprecision, vagueness or uncertainty, with main success metrics linked to fundamental areas of expertise.

B. *Experimental Studies*

This second section focused on how efficiently and successfully can these techniques be applied and classifies the related studies into those that apply these technologies and those that conduct experiments on the development of these technologies. Such papers have been separated into various topics and implementations. (see Table II)

1) *Studies Conducted on Machine Learning Methods*

The works identified in software development techniques and risk management are divided into specific categories according to ML procedures.

2) *Other Methods*

This section reviews the other methods and how it's used. Such papers have been separated into various topics and implementations. Based on the approaches of software engineering and risk management, selected works are categorized into specific categories.

TABLE II. THE REVIEW AND STUDY PAPERS EXPERIMENTAL THE STUDIES CONDUCTED ON MACHINE LEARNING AND OTHER METHODS

No	References	Approach	Method used	Studies Conducted on	contribution
1	Castro-Herrera and Cleland-Huang [20]	Machine Learning Methods	introduces and evaluates two ML methods.	Both techniques resulted in improvements for the subset of HIPAA regulations that were hard to trace using traditional trace-retrieval methods.	designed to improve the quality of traces generated between regulatory codes and product-level requirements
2	Zhang, et al. [21]		Not mentioned	implementing analysis technology in software analysis practice, a broad range of domain expertise and information expertise should be incorporated into software analytics	management, ML, broad-scale processing, computing, and visualization of information; Place is focused on positive technical migration experiences in Microsoft Study Asia information analytics.
3	Pospieszny [22]		Not mentioned	define a direction for further research on software estimation applications of data mining and ML.	To improve process prediction, to maximize resource use, and product quality, and to eventually increase the chances for effective project delivery by recommending specific use cases for prescriptive analytics.
4	ManikReddy and Iyer [23]		Not mentioned	reveals that the digitalization and ML approach to this omnipresent issue is special.	The team built a platform assisted with ML through value-stream modeling, which helped to get ideas into the market more quickly.
5	Choetkiertikul, et al. [24]		The method is explained by an appraisal for the initiative estimation of alternate ML algorithms.	In addition to Agile's manual Planning Poker is introduced to tackle this automatic calculation technique dubbed "Auto-Estimate."	Defined ML approaches have been shown to work in the latter stages of a project stronger than Poker preparation. This method of estimation is tested for precision, applicability, and validity, and the findings are interpreted in a real-life context.
6	Moharreri, et al. [25]		focused on four machine learning techniques that were analyzed.	Ensemble Effort Estimation to estimate the software effort by integrating more than one solo estimate technology with a combination law.	The findings of this study indicate that the proposed heterogeneous EEE provides a very promising output and there is no suggested combiner law.
7	Hosni, et al. [26]		Not mentioned	include a tool for extracting data from different sources, a prediction model for predicting each team member's suitability for a specific task, and extracting, transforming, and loading tools for peer review and summary analysis to provide a viable solution to draw on peer assessment features.	This research provides a CollabCrew framework for task management that is developed especially for software development teams that delegate assignments dynamically based on team members' expertise and previous experience.
8	Samath, et al. [27]		Run tests on 44 updates from 14 open-source initiatives and using the corresponding classification Naive Bayes and the SVM.	Results indicate that the data filter technique enhances dramatically the efficiency of cross-project defect prediction and substantially enhances the hierarchical pick method.	offer a detailed analysis of esteemed data filters with a new filter invented in this article.
9	Li, et al. [28]		Not mentioned	With empirical analysis on Travis Continuous Integration, it was shown that F-Measure increased 40% by ACONA while the cumulative error was decreased by 63.2% and the modified paradigm overtakes current strategies. Collect TravisTorrent data that synthesizes Github and Travis-ci details.	proposing ACONA's Active Adaptation Online Model approach which adapts the community of classifiers trained in various projects dynamically to a new project using only a small amount of new data that it has actively chosen.
10	Ni and Li [29]		use RF, Multilayer Perceptron (MLP) and SVM ML models to forecast the effort	The predictive performance of the RF, MLP, and SVM models was tested. Compared to MLP and SVM, the experimental findings from the RF model were higher.	Owing to the advantages of making forecasts at the first stages of product growth, the Usage case points sample size metric is used. Before training models on the data collection, the standardization preprocessing strategy was implemented.
11	Papatheocharous and Andreou [30]		The experiment results imply that the SVM outperforms other classifiers not only in accuracy, precision, recall, and F-measure, but also has good stability and lower cost.	introducing the cost-sensitive learning method into outsourced software project risk prediction.	It is a good individual classifier for outsourced software project risk prediction. selected five classifiers, and introduced a cost-sensitive learning method to build intelligent prediction models respectively.

12	Hongming, et al. [31]		Investigated the effect of noisy domains on the learning accuracy of eight ML and statistical pattern recognition algorithms	Experimental results show that the algorithm can improve prediction for software effort corrupted by noise with reasonable and much-improved accuracy.	The behavior of these methods is explored for different levels of noise, type of noise in attributes and class labels, and type of noise in sampling set.
13	Twala [32]		The methodology finds interaction trends that can provide domain experts with valuable knowledge and improve their faith in decision making.	To predict if a software project is flawed and identify crucial factors in the data that indicated they predict, implement the methodology in the NASA MDP library software fault data sets.	The findings of the tests often indicate the impact on the efficiency of various methods of decertification.
15	Lopez-Martin, et al. [33]		For comparing ML algorithms, use the correlation coefficient to evaluate the results. Based on this value, rank the algorithms according to each metric of interest.	Each project dataset is used to compare the algorithms, including linear regression, SMO, Multilayer Perceptron, M5P, Leastmedsqr, REPtree, and Gaussian process.	Experiment results show this kind of algorithm is effective compared to the ML algorithms.
16	Han, et al. [34]		by proposing an approach called hyperheuristic instead of the Decision-Tree algorithm itself.	In software ventures, the capacity to build a highly detailed understood prediction model is important provided that it allows the stakeholder to handle the team's resources effectively with improved trust in the product predictions.	The findings demonstrate that the hyperheuristic decision algorithm automatically generates state-of-the-art and evolutionary decision-making algorithms statistically outperforming, alongside traditional logistic regression.
17	Basgalupp, et al. [35]		The techniques listed here are well known, but they are an original contribution to the ensemble.	Compared with an individual imputation process, benchmarking results on ten industrial data sets indicate that the proposed set strategy may boost forecasting efficiency, particularly if multiple imputations are a component of an ensemble.	suggest a way to enhance the estimation of machine action by utilizing an algorithm of decision-making and by constructing the ensemble utilizing two methods of imputation as components.
18	Twala and Cartwright [36]		K-Nearest Neighbor Algorithm (KNN). To investigate to what extent parameter settings affect the performance of ML in software effort estimation, and what ML is more sensitive to their parameters.	Considering an online learning scenario where ML is updated with new projects as they become available, systematic experiments were performed using five ML under several different parameter settings on three data sets.	The average performance of k-NN across different projects was not so much affected by different parameter settings, but the parameter settings that obtained the best average performance across time steps were not so consistently the best throughout time steps as in the other approaches
20	Song, et al. [37]		The process Relevance Vector Machine (RGM) implemented can overcome these three challenges. Shows that RVM directly models exercise noise and may evaluate data generated using approximate exertion noise standard.	Shows that RVM directly models exercise noise and may evaluate data generated using approximate exertion noise standard. In keeping with the Bayesian method, RVM allows a probabilistic prediction for a new initiative, where every projected commitment is related to a likelihood.	demonstrate that a confidence level prediction period dependent on this probabilistic forecast can be obtained. Also, it shows that RVM is an excellent prediction to tackle these three challenges.
21	Minku and Yao [38]		The process Relevance Vector Machine (RGM) implemented investigating the usage of a statistical algorithm utilizing developer characteristics to distribute story points to study.	Evaluated sample output metrics including precision, total actual error, and uniform precision. Preliminary findings indicate that the model utilizing developer apps visually beats text-based versions, suggesting a positive testing strategy.	The model's output is contrasted to the models' results, based on features derived from question document.
22	Song, et al. [39]		The process Relevance Vector Machine (RGM) implemented reported methods of tuning rely on the linear types of change, rather than a nonlinear artificial neural network.	In contrast with the other non-linear modification strategies ANN and Intense Learning System (ELM), the findings are corroborated in three positive repository test packages.	In the comprehensive analysis of the correct process of calibration, Least Squares SVM (LS-SVM) emerges as a ray of hope and is a nonlinear analogy error correction tool.
23	Scott and Pfahl [40]		K-Means: explores how clustering approaches are appropriate to help identify successful divisions for Dycom for Cross-Company (CC).	focused on four SEE tables of CC subsets of varying sizes.	The study often involves a Comparison Inside the Client (WC) model. Clustering Dycom with K-means will help separate the CC projects and produce comparable or better predictive efficiency compared to Dycom.

			For the development of the CC subsets, Dycom is expanded to use clustering methods.		
24	Benala and Bandarupalli [41]	Other Methods	Logic Model	adopt the context of software quality optimization; i.e. adjusting a software project such that improve quality attributes such as the defects, the months, and the effort.	Proponents of parametric models argue that there exist domain-independent models that can be tuned to local details.
25	Minku and Hou [42]		To easily identify crucial shifts in the project.	suggested incorporating inference + visualization. (1) delete superfluous project data by reducing dimensionality, reducing column, and reducing functionality. (2) imagine the project data reduced volume.	Found out that there is no variation in efficiency between data inference and space inferences. Managers should have a concise description of project results, through which decisions that most affect project commitment can be identified conveniently and objectively.
26	Brady and Menzies [43]		Parametric Models examine the benefits of approaches that involve expense misclassification in designing software error prediction models utilizing project knowledge from a shared repository.	To consider best modeling practices for estimation of the device malfunction, the tests described offering valuable knowledge.	Find out then that cost-sensitive schooling does not deliver operating points that outperform cost-insensitive classifiers. However, a cost-sensitive modeling advantage is that the operational threshold for the cost differential is chosen explicitly.

III. Discussion

The state-of-the-art of software project risk assessment using ML technologies are discussed in this study. This study aims to highlight the trends in research in this field. This study varies from prior assessments as it is recent and reflects not on implementations, but on the literature itself. The related literature is proposed as a taxonomy. Developing a literature taxonomy in a research area may have many advantages, particularly for an evolving one. On one side, there are many publications on the taxonomy. The vast number of papers in this field, the lack of some sort of framework, and therefore an analysis of this region may be overshadowed by a new researcher who studies a software project risk evaluation. Introducing numerous papers on the topic, other papers discuss existing developments in the risk management of software projects. Some studies have developed current ML models and applications. A taxonomy of the literature helps to arrange these different works and events in a relevant, accessible, and coherent way. On the other side, the taxonomy framework provides researchers with some valuable insights into the topic. First, the possible field directions for research are outlined. For example, the taxonomy of the assessment of software projects in current work shows that researchers are inclined to suggest structures for application creation and activity to include a direction to pursue in this field.

The taxonomy proposed in this research is using common language, similar to taxonomies in other fields, for researchers to convey and discuss emerging works such as development papers, comparative studies, and software project risk assessment review using ML technology. The research show facets of the literature content: the reasons behind the creation of digital project risk management utilizing ML technology, the obstacles to using such technologies effectively, and the guidance about how such issues can be alleviated.

The benefits of utilizing the ML platform used for software project risk management are clear and compelling. This portion describes some of the advantages in the literature that are grouped into groups dependent on specific advantages. For further discussion, the corresponding references are cited (see Figure. 1).

A. Benefits Related to Prediction Cost Evaluation Model

The system for validating the metrics of the source code and the correct metrics to boost the efficiency of the model for fault prediction. The predictive fault models are validated using a cost assessment framework. The main findings include: (1) Most voting methods overwhelmed by other methods (2) chosen source code process metrics use proposed validation system source code metrics, as an input for improved performance in contrast with all other methods (3) a failure prediction approach is effective in software projects with a percentage of error groups below the threshold value suggested in Wagner [44]. In the fault prediction process focused on ensemble approaches, the implementation of predicted fault removal costs, monitoring costs, and the uniform fault removal costs are the only input.

B. Benefits Related to The Risk Management

The various tasks in the preparation process of software projects can be grouped into two key practices, namely commitment evaluation and risk reductionPressman [45]. The software effort estimation is based on several cost factors and risk management activities include identifying, addressing, and removing the risk of the software project before unwanted outcomes [46]. The estimated effort in software development calculates efforts.

Risk management may be defined as all required risk management behavior. Risk-assessment and risk-control are the two primary stages of risk reduction. Risk-assessment is a research method in which risk factors are identified, danger potential impacts evaluated or measured, and threats are a priority. Risk-Control is a process to develop risk resolution software plans, monitor the status of the risk, implement a risk management plan, and solve the problem by correcting possible variations of the plan. A crucial factor in assessing the progress of the software development project is a risk evaluation that is the core task in the project preparation phase [47]. But the performance of risk assessment practices concerning conventional risk reduction approaches[48] relies heavily on human judgment and expertise, and risk assessment is viewed as being overly expensive and costly for the software project.

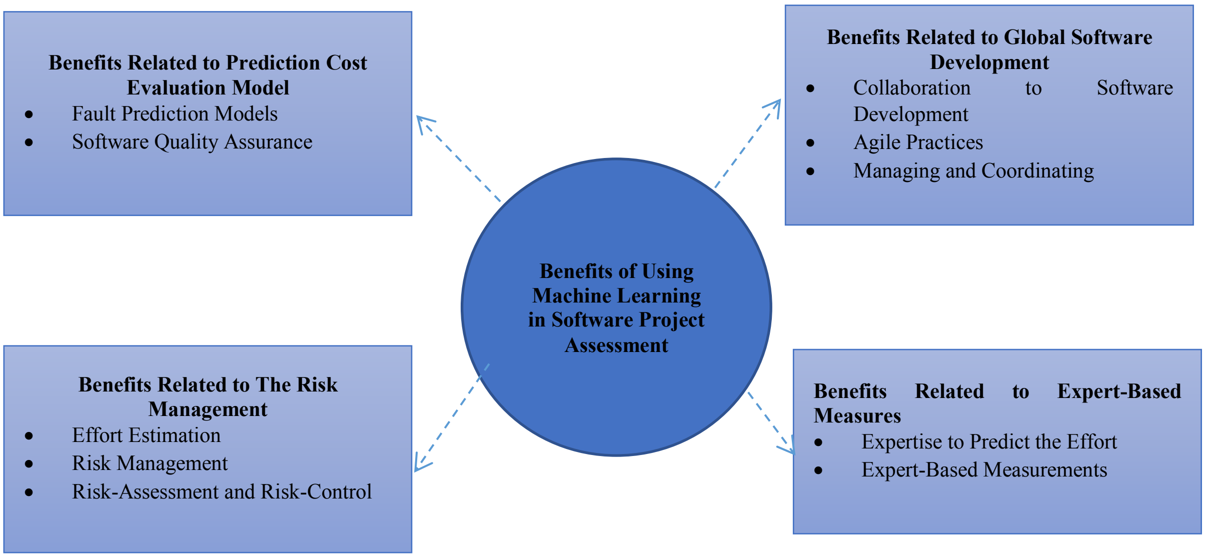
Benefits Related to Prediction Cost Evaluation Model
Fault Prediction Models
Software Quality Assurance
Benefits Related to Global Software Development
Collaboration to Software Development
Agile Practices
Managing and Coordinating
Benefits of Using Machine Learning in Software Project Assessment
Benefits Related to The Risk Management
Effort Estimation
Risk Management
Risk-Assessment and Risk-Control
Benefits Related to Expert-Based Measures
Expertise to Predict the Effort
Expert-Based Measurements

Fig. 1. Categories of Benefits of Using Machine Learning in Software Project Assessment

C. Benefits Related to Global Software Development (GSD)

The enhanced usage of GSD as a way of lowering the costs of producing applications and exposure to a wide package of skilled research is another significant development in the industry. GSD projects are often confronted with serious problems, although they are increasingly

These include communication issues between the project members in distribution, difficulties in establishing appropriate group relations, cultural issues, and difficulties in managing and coordinating the work in the projects in distribution. In short, in a distributed setting, intense cooperation in the creation of applications continues to be challenging[49]. Based on face-to-face coordination, which in certain GSD environments is challenging and hard to manage, certain agile methods appear complicated to integrate the two from the first try.

D. Benefits Related to Expert-Based Measures

The software project predictor uses its expertise to predict the work of such projects as experts in this group. The estimator's experience is focused on the question area as well as his knowledge of related and traditional ventures. The focused model would have a significant advantage if its reduced number of measures would allow expert measurement to be omitted. The role of expert actions must not be ignored. In order to avoid evaluations that require expert testing, the evaluation model developed without expert- based measurements performs considerably worse. The utilized quality model and quality assurance method help. For future work, study various constraints for the selection of the predictor measures. The amount of specific instruments used to quantify behaviour, the quality and the commitment needed to use a device or the precision of measurements may be potential constraints.

IV. CONCLUSIONS

The literature review concludes that considerable research has been conducted on ML approaches in the software project risk assessment. Work distribution is constant over the years. The principal ML methods used for automated effort calculation are ANN, Fuzzy Logics, Genetic and regression algorithms. The exact measurement of effort is one of the main software development practices. The period and complexity of the program programs was directly impacted. This paper aims to offer observations through the study and taxonomisation of similar works. Different work on the software effort estimation can draw specific patterns. These works are approximately divided into four categories, namely reviews and surveys, with review and survey papers related to risk assessments of the software project in the first category. The second group covers papers that focus on case studies, ML and project management methods. An in-depth review of these articles would help the software project risk review with the ML approaches to define and explain the threats, advantages and recommendations. However, because of the huge amount of ML algorithms, various machine study algorithms remain unanalyzed. Although the literature on project management explains the performance of projects and loss, there is a lengthy tradition of disagreements over whether project progress should be calculated. There are conflicts of opinion about what reflects the progress of a project and how it is calculated, these guidelines will resolve the problems facing a software project in ML methods and open up work possibilities in this sector.

Acknowledgment

This research is generously funded by Universiti Tenaga Nasional (UNITEN) and UNITEN R&D Sdn Bhd, under TNB Seed Grant projects U-TC-RD-19-10 titled “Smart Software Project Management Assessment System”.

References

[1] T. A. Oun, T. D. Blackburn, B. A. Olson, and P. Blessner, "An enterprise-wide knowledge management approach to project management," Engineering Management Journal, vol. 28, pp. 179-192, 2016.

[2] H. G. Gemünden, "Success factors of global new product development programs, the definition of project success, knowledge sharing, and special issues of project management journal®," Project Management Journal, vol. 46, pp. 2-11, 2015.

[3] S. W. Hughes, D. D. Tippett, and W. K. Thomas, "Measuring project success in the construction industry," Engineering Management Journal, vol. 16, pp. 31-37, 2004.

[4] A. PMI, "guide to the project management body of knowledge (PMBOK guide)," in Project Management Institute, 2013.

[5] L. J. Kirsch, "Software project management: An integrated perspective for an emerging paradigm," Framing the Domains of IT Management: Projecting the Future... Through the Past, pp. 285-304, 2000.

[6] A. M. Aladwani, "IT project uncertainty, planning and success," Information Technology & People, 2002.

[7] G. R. Cates and M. Mollaghasemi, "The project assessment by simulation technique," Engineering Management Journal, vol. 19, pp. 3-10, 2007.

[8] V. S. Parsons, "Project performance: How to assess the early stages," Engineering Management Journal, vol. 18, pp. 11-15, 2006.

[9] Y. Rosenfeld, "Root-cause analysis of construction-cost overruns," Journal of construction engineering and management, vol. 140, p. 04013039, 2014.

[10] J. Wang, J. Li, Q. Wang, H. Zhang, and H. Wang, "A simulation approach for impact analysis of requirement volatility considering dependency change," in International Working Conference on Requirements Engineering: Foundation for Software Quality, 2012, pp. 59-76.

[11] S. Ferreira, J. Collofello, D. Shunk, and G. Mackulak, "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation," *Journal of Systems and Software,* vol. 82, pp. 1568-1577, 2009.

[12] A. Idri, I. Abnane, and A. Abran, "Systematic mapping study of missing values techniques in software engineering data," in 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015, pp. 1-8.

[13] A. M. Alsalemi and E.-T. Yeoh, "A Systematic Literature Review of Requirements Volatility Prediction," in 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), 2017, pp. 55-64.

[14] S. P. Pillai, S. Madhukumar, and T. Radharamanan, "Consolidating evidence based studies in software cost/effort estimation—a tertiary study," in TENCON 2017-2017 IEEE Region 10 Conference, 2017, pp. 833-838.

[15] P. Sharma and J. Singh, "Machine Learning Based Effort Estimation Using Standardization," in 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 716-720.

[16] J. Shivhare and S. K. Rath, "Software effort estimation using machine learning techniques," in Proceedings of the 7th India Software Engineering Conference, 2014, pp. 1-6.

[17] C. A. Stewart, D. Y. Hancock, J. Wernert, T. Furlani, D. Lifka, A. Sill, et al., "Assessment of financial returns on investments in cyberinfrastructure facilities: A survey of current methods," in Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), ed, 2019, pp. 1-8.

[18] R. Malhotra and A. Chug, "Software maintainability: Systematic literature review and current trends," International Journal of Software Engineering and Knowledge Engineering, vol. 26, pp. 1221-1253, 2016.

[19] J. A. L. García, A. B. Peña, P. Y. P. Pérez, and R. B. Pérez, "Project control and computational intelligence: Trends and challenges," International Journal of Computational Intelligence Systems, vol. 10, pp. 320-335, 2017.

[20] C. Castro-Herrera and J. Cleland-Huang, "A machine learning approach for identifying expert stakeholders," in 2009 Second International Workshop on Managing Requirements Knowledge, 2009, pp. 45-49.

[21] D. Zhang, Y. Dang, J.-G. Lou, S. Han, H. Zhang, and T. Xie, "Software analytics as a learning case in practice: Approaches and experiences," in Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering, 2011, pp. 55-58.

[22] P. Pospieszny, "Software estimation: towards prescriptive analytics," in Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, 2017, pp. 221-226.

[23] P. ManikReddy and J. Iyer, "Effective collaboration across the globe through digital dash boards and machine learning," in 2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE), 2018, pp. 30-34.

[24] M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, "Predicting delays in software projects using networked classification (t)," in 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015, pp. 353-364.

[25] K. Moharreri, A. V. Sapre, J. Ramanathan, and R. Ramnath, "Cost-effective supervised learning models for software effort estimation in agile environments," in 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), 2016, pp. 135-140.

[26] M. Hosni, A. Idri, A. B. Nassif, and A. Abran, "Heterogeneous ensembles for software development effort estimation," in 2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI), 2016, pp. 174-178.

[27] S. Samath, D. Udalagama, H. Kurukulasooriya, D. Premarathne, and S. Thelijjagoda, "Collabcrew—an intelligent tool for dynamic task allocation within a software development team," in 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2017, pp. 1-9.

[28] Y. Li, Z. Huang, Y. Wang, and B. Fang, "Evaluating data filter on cross-project defect prediction: Comparison and improvements," IEEE Access, vol. 5, pp. 25646-25656, 2017.

[29] A. Ni and M. Li, "Poster: ACONA: Active Online Model Adaptation for Predicting Continuous Integration Build Failures," in 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion), 2018, pp. 366-367.

[30] E. Papatheocharous and A. S. Andreou, "A hybrid software cost estimation approach utilizing decision trees and fuzzy logic," International Journal of Software Engineering and Knowledge Engineering, vol. 22, pp. 435-465, 2012.

[31] Z. Hongming, F. Bin, M. Xizhu, S. Lijun, X. Z. Xiangzhou, and H. Yong, "A Cost-sensitive Intelligent Prediction Model for Outsourced Software Project Risk," 2013.

[32] B. Twala, "Reasoning with Noisy Software Effort Data," Applied Artificial Intelligence, vol. 28, pp. 533-554, 2014.

[33] C. Lopez-Martin, A. Chavoya, and M. E. Meda-Campaña, "A machine learning technique for predicting the productivity of practitioners from individually developed software projects," in 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2014, pp. 1-6.

[34] W. Han, H. Jiang, X. Zhang, and W. Li, "A Neural Network Based Algorithms for Project Duration Prediction," in 2014 7th International Conference on Control and Automation, 2014, pp. 60-63.

[35] M. P. Basgalupp, R. C. Barros, T. S. da Silva, and A. C. de Carvalho, "Software effort prediction: a hyper-heuristic decision-tree based approach," in Proceedings of the 28th Annual ACM Symposium on Applied Computing, 2013, pp. 1109-1116.

[36] B. Twala and M. Cartwright, "Ensemble missing data techniques for software effort prediction," Intelligent Data Analysis, vol. 14, pp. 299-331, 2010.

[37] L. Song, L. L. Minku, and X. Yao, "The impact of parameter tuning on software effort estimation using learning machines," in Proceedings of the 9th international conference on predictive models in software engineering, 2013, pp. 1-10.

[38] L. L. Minku and X. Yao, "How to make best use of cross-company data in software effort estimation?," in Proceedings of the 36th

International Conference on Software Engineering, 2014, pp. 446-456.

[39] L. Song, L. L. Minku, and X. Yao, "The potential benefit of relevance vector machine to software effort estimation," in Proceedings of the 10th International Conference on Predictive Models in Software Engineering, 2014, pp. 52-61.

[40] E. Scott and D. Pfahl, "Using developers' features to estimate story points," in Proceedings of the 2018 International Conference on Software and System Process, 2018, pp. 106-110.

[41] T. R. Benala and R. Bandarupalli, "Least square support vector machine in analogy-based software development effort estimation," in 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2016, pp. 1-6.

[42] L. L. Minku and S. Hou, "Clustering Dycom: an online cross-company software effort estimation study," in Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering, 2017, pp. 12-21.

[43] A. Brady and T. Menzies, "Case-based reasoning vs parametric models for software quality optimization," in Proceedings of the 6th International Conference on Predictive Models in Software Engineering, 2010, pp. 1-10.

[44] S. Wagner, "A literature survey of the quality economics of defect-detection techniques," in Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering, 2006, pp. 194-203.

[45] R. S. Pressman, Software engineering: a practitioner's approach: Palgrave macmillan, 2005.

[46] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," Journal of Systems and Software, vol. 86, pp. 144-160, 2013.

[47] K. Heldman and W. Heldman, CompTIA Project+ Study Guide Authorized Courseware: Exam PK0-003: John Wiley & Sons, 2010.

[48] E. M. Hall, Managing risk: Methods for software systems development: Pearson Education, 1998.

[49] T. Menzies, O. Mizuno, Y. Takagi, and T. Kikuno, "Explanation vs performance in data mining: A case study with predicting runaway projects," Journal of Software Engineering and Applications, vol. 2, p. 221, 2009.