Informatics Bulletin, Faculty of Computers and Artificial Intelligence, Helwan University
Published Online *Vol 5 Issue 1, January 2023*
(https://fcihib.journals.ekb.eg)

Risk Prediction using Machine Learning Techniques in the Domain of Global Software Development: A Review

Hossam Hassan[1,*], Manal Abdel Kader[1], Amr Ghoneim[2]
[1]*Information System , Computers and Artificial Intelligence , Helwan , Cairo ,Egypt*
[2]*Computer Science , Computers and Artificial Intelligence , Helwan , Cairo ,Egypt*

hossamhsb96@gmail.com, *manal_8@hotmail.com*, *amr.ghoneim@fci.helwan.edu.eg*

***Abstract*— The field of software engineering is currently trending toward the high demand for global software development. The idea of employing a software engineering specialist from anywhere in the world with a variety of skills and expertise to meet needs and at an affordable price is the main driver behind the fastest-growing global software development approach. On the other hand, it can be difficult to integrate distributed teams with a company's resources and tools. Therefore, a precise assessment of the risks associated with the software project is required, along with early risk prediction. This comprehensive literature review provides an overview of various risk prediction models used in global software development. This literature review discusses 12 studies that use Many models and techniques such as machine learning, neural networks, mathematics, algorithms, similarity analysis, and frameworks that try to predict software failures and risks. In addition, this research goes into depth and provides suggestions for improving machine learning models and frameworks for future studies.**

***Index Terms*—global software development, risk prediction, software prediction risk model, Risk factors, machine learning.**

I. Introduction

The entire software development approach has completely changed in the last two decades to support distributed environments with distributed teams rather than the traditional approach, which uses teams within the same environment and cultures [1]. The concept of employing external resources was derived from the business strategies of companies over the past two decades [2]. Nowadays, the concept of Global Software Development (GSD) is widely adopted because of the need for specialized resources and tools at affordable prices throughout the globe [3]. In addition, GSD has seen a significant increase in recent years, including contracts and companies that adopted GSD. It can be considered that GSD is the new age of development projects using different teams with different geographical locations and time zones [4]. The idea of distributed teams with different geolocations and time zones can be illustrated in Figure 1.
Companies prefer using the GSD because of its advantages, which include expertise sharing across the globe, using the latest tools and techniques, availability of resources, economic benefits, affordable costs, and efficient and effective overall project success [2],[7].
However, when organizations tried to adopt this approach, they faced issues related to communication between teams within different environments. But, thanks to modern agile methodologies, the difficulty of communication has been solved [2].

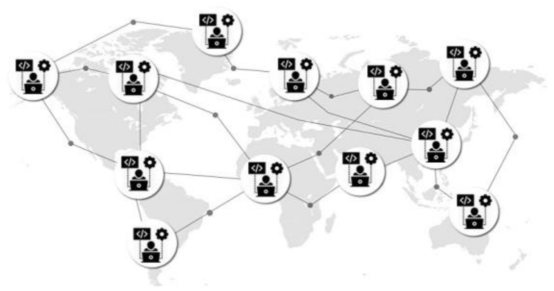


Fig. 1. Explaining the idea of global software development (GSD) [7].

While companies were trying to adopt GSD approach, they faced many difficulties and challenges, including communication between distributed teams, language barriers, cultural rules and limitations, time zone, leadership, team capacity, and project management [8]–[11]. In Figure 2, the authors classified the challenges that impact the GSD into three main categories (distance, communication, and coordination) that were affecting each other, which led to the complexity of adopting the GSD.
Therefore, companies were trying to minimize the risk of failure in adopting the GSD approach. Accordingly, Project risk can be defined as the collection of software factors, conditions, and rules that may be a potential threat to the overall project success. So, it's important to think about how often these risks happen and how to predict them [12]. A study by the Project Management Institute (PMI) shows that most approaches and techniques for risk management were ignored and abandoned, especially in the IT field, because they are too general or limited to a specific use case [13]. However, software projects that use tools to predict risk can detect around 70% of harmful risks and avoid 90% of them [14]. In addition, the Authors found that using old and traditional approaches to risk management tools,

could not address the modern and unique characteristics of GSD in deep detail [8]. And this explains why, in this literature review, machine learning techniques were used to predict the risk of software because of the nature of ML due to self-learning and healing. This Systematic Literature Review (SLR) highlights the effectiveness of machine learning, neural networks, algorithms, and other strategies for predicting software risk in GSD. It has a significant influence on both GSD and the overall performance of the project. This may also assist project managers to identify risks early in the project's development and provide them with the tools necessary to prevent them for improved software project development in the field of GSD.

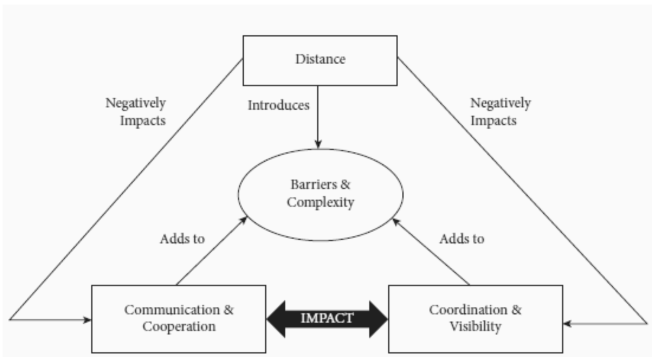


Fig. 2. Challenges and complexity of adopting the GSD Approach [8], [11].

The paper is further organized as follows: Section II describes the research methodology, while Section III provides context and motivation for risk prediction, global software development, and risk assessment using machine learning. Section IV describes the state-of-the-art studies and summarizes the understanding of global software development, challenges, and risks in this field using a machine learning technique. Section V discusses the results of this literature review. Finally, section VI discusses the conclusion of this Literature review and provides future work recommendations and enhancements.

II. RESEARCH METHODOLOGY

To achieve our goal, which is to minimize the risk of failure of adopting the GSD, SLR was used to analyze and research the software risk factors to investigate how machine learning and other techniques were used to predict software risks in global software development. We used SLR in our study because it is more accurate, powerful, and organized than the traditional literature review. It helps us figure out our goals, evaluate the results, and put them into useful groups [15], [16]. SLR consists of three main phases as guided by Kitchenhem [17]:

- Planning the review.
- Conducting the review.
- Reporting the review.

A. *Phase 1: Planning the Review:*

This phase includes the cornerstone of the SLR, which was initialized by defining prerequisite steps as follows.

- Specify the research questions.
- Select suitable research repositories.
- Defining research criteria.
- Defining inclusion and exclusion.
- Defining quality criteria.

1) Research Questions:

RQ1: Which machine learning techniques and other methods were used to predict the risks of a software project ?
RQ2: What data sets were used and what were their characteristics and validity ?
RQ3: What were the top factors that affect the overall success of the software project ?
RQ4: Can software project risk factors be categorized ?
RQ5: What level of efficiency can these techniques be applied?

2) Data Repositories:

In the SLR, online digital libraries were used, including:

- IEEE Xplore
- Google Scholar
- Science Direct
- Springer

3) Search String:

The search strings were chosen from the SLR keywords and their alternatives in GSD. These search strings were categorized into two categories as shown in Table1.

- Software project risk prediction.
- Global software development.

Table. 1. Search Strings for the Systematic Literature review.

Group	Search String
Software Risk prediction	(“Risk factors” OR “Risk Prediction” OR “Machine Learning Risk Prediction” OR “Failure Prediction” OR “Software Failure Prediction”)
	OR
Global Software Development	(“Global Software Development” OR “Distributed teams” OR “Distributed Environment” OR “GSD” OR “Off-shore”)

4) INCLUSION CRITERIA

Inclusion and exclusion criteria for SLR were guidelines by Kitchenhem [17]. Following inclusion criteria were listed:
IC1: The selected study should be a journal or conference only.
IC2: The studies should be focused on risk prediction.
IC3: The studies should be focused on global software development.
IC4: The studies should be focused on machine learning, algorithms, and a mathematical method for predicting risks.
IC5: Studies published between 2018 to 2022.

5) Exclusion Criteria:

EC1: Studies that were not answering the research questions.
EC2: Studies were not discussed software failure factors.
EC3: Studies were not related to the GSD.

EC4: Studies that were published before 2015.

6) Quality assessment of selected studies:
This phase is significant because the quality of the selected studies was checked and compared to our objectives, research questions, and goals.

B. Phase 2: Conducting the Review:

1) PRIMARY DATA SELECTION:
In this phase, filtration techniques, search criteria, inclusion, and exclusion were applied. And the process of selecting the primary studies has started. The Tollgate approach was used to make the selection process powerful and be more systematic and organized way [18]. In this phase, filtration techniques, search criteria, inclusion, and exclusion were applied, and the process of selecting the primary studies has started. The Tollgate approach was used to make the selection process powerful and be more systematic and organized way.

2) DATA EXTRACTION:
Studies were extracted based on some rules, including research method, publication year, kind of the study, restrictions, and limitations to the studies.

3) DATA SYNTHESIS:
The studies that were extracted were evaluated and compared against our research questions and the objectives of our study.

C. Phase 3: Reporting the Review:

In this phase, the chosen studies were double-checked against the quality questions. The result is a well-thought-out list of studies that were ready to be discussed and should be ready to be investigated.

III. BACKGROUND AND MOTIVATION

Many studies tried to design and develop techniques for software projects. Although, most of these techniques were high-level approaches or theoretical approaches [7]. therefore, the interest in this domain is still under development and needs more interest and focus. The challenges and difficulties of the distributed environment were different and more complex than those of the cooperative environments, and this difference was due to their different characteristics, i.e., geographical location, different time zones, competence level, and hidden costs [19].
Companies that are using the GSD methodology to minimize project costs are not able to maximize its utility because of the GSD nature and characteristics [20]. In the last few years, many studies have shown much interest and attention related to risk prediction of a software project using artificial intelligence techniques. However, some characteristics of GSD are still under investigation and development, i.e., relationships between risk factors and hazards may be due to more than one factor [21]. Figure 3's statistics showed that only 38% of offshoring projects were successful, and around 50% of the offshoring projects failed to meet expectations. Therefore, much attention should be provided to minimize the risk of software failure in the GSD.

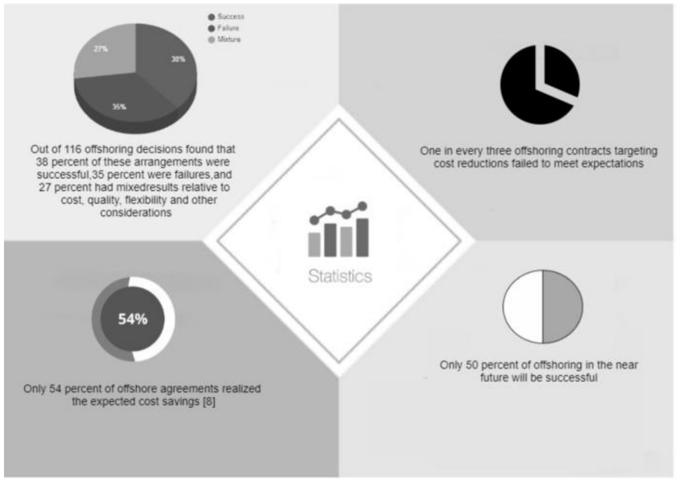


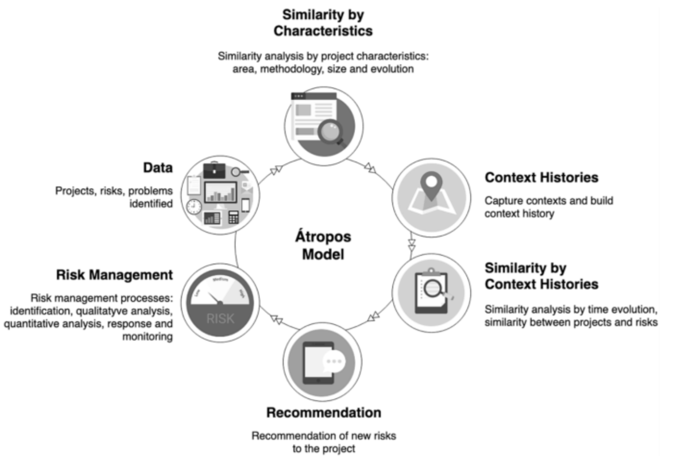
Fig. 3 Statistics on the success rate of offshoring software projects [7], [22].

IV. RELATED WORK

All the studies that look at how machine learning and other algorithms can be used to predict risks in GSD were discussed in this section.

In [14], a software risk prediction model was created based on risk analysis of the project by using its context history and project characteristics in the software development life cycle (SDLC) as shown in Fig.4. the model is called the Atropos model and consisted of six main phases listed below:
1) *Data Gathering through interface and bulk uploading.*
2) *Similarity by characteristics of the project.*
3) *Store context histories of the project*
4) *Similarities by context histories*
5) *Recommendation of any potential risks*
6) *Risk management and monitoring.*

The dataset was collected based on 153 software projects from a financial company. Evaluation metrics of the model showed an acceptance rate of 73% and an accuracy rate of 83%, and these results were assessed by 18 experts. Limitations and Future work for the model are improve the model's accuracy, improve the proposed model and methodology, additional use of prototype, the number of practitioners, and the duration of the case study (5 weeks only).


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
Fig. 4. The 6 phases of the proposed Atropos model [14].

In [8], artificial neural network (ANN) model was created to predict the risk factors in GSD. The model used algorithms such as Levenberg–Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient. The dataset was collected by sending 760 questionnaires to companies. 390 were received, and 116 were rejected, leaving 274 responses that were used as the primary data set. Evaluation metrics of the model were conducted by using least mean square error (MSE), and the results showed that Bayesian Regularization gave better results as compared with the other two approaches and matched the results from these studies [23], [24]. Limitations and Future work for the model are the sample dataset needs to include many companies and random data collection should be used to generalize the model, also the author recommended to used deep learning to get more insights and accurate results in the future.

The authors in [25], proposed a hybrid Fuzzy DEMATEL–FMCDM–TODIM approach for evaluating software risk factors to find out and rank the software risk factors during the SDLC related to project performance, and the results show that the fuzzy framework can be more helpful when making decisions. The dataset was collected by using 40 software projects from the real industry. Evaluation metrics of the model were conducted by using Fuzzy Multi-Criteria Decision Making (FMCDM). Limitations and future work for the model: software risk factors investigation, limitations related to statistical problems while dealing with uncertainty and vagueness problems, expanding the proposed Fuzzy and using ANN models in the future.

In [26], the authors provided a software reliability prediction algorithm. They used fuzzy logic and ANN in their model. The dataset was collected by using a dataset retrieved from John Musa of Bell Laboratories and received from the IEEE repository. Evaluation metrics of the model were conducted by using RMSE error and showed that the fuzzy-neural method was best compared to other algorithms. Limitations and future work for the model: the model is restricted to one factor (time to failure). In addition, many software risk factors should be used to evaluate this model better.

In [27], an empirical investigation was conducted to figure out the top requirements of engineering (RE) practices GSD. Among the 66 practices, the results showed that only six key factors play an important role in GSD, as listed below:

1. Identify and consult with system stakeholders.
2. Prioritize requirements.
3. Define system boundaries.
4. Define standard templates for describing requirements
5. Check requirements doc meet your standards.
6. Uniquely identify each requirement.

The dataset was collected by conducting an online survey questionnaire. For the evaluation of these factors, 56 experts from GSD were involved. Limitation and future work: the questionnaire relied only on closed questions and focused only on the company size, testing these factors, and trying to develop a framework to be used in the future.

The authors in [28], focused on scaling agile projects in the domain of GSD. They mapped 44 agile practices to the SAFe Framework. Instructions were given for how the SAFe practices can be used in agile global software development (AGSD) projects. The dataset was collected by reviewing 86 studies. Of these studies, only 24 papers discussed the scaling of agile, from which the authors selected 44 practices to be mapped on the SAFe Framework. Limitations and future work: (AGSD) practices need to be evaluated and should also be tested in the real industry. In addition, the mapped process of these practices needs to be evaluated.

In [29], the authors tried to prioritize the success factors that affect Requirement Change Management (RCM) in the GSD. Fuzzy logic analytical hierarchy progress (FAHB) was used to conduct the prioritization. The result of this study was to find out the RCM success factors and categorize them into four groups: team, technology, process, and organizational management, as shown in Fig.5. The dataset was collected by conducting a questionnaire survey and retrieved around 81 responses. Evaluation metrics for the prioritization were conducted by using experts' responses. Limitations and future work: sample size of the dataset needs to be widened, and organization size and types should be considered, in addition, success factors, barriers, and best practices need more investigation and analysis.

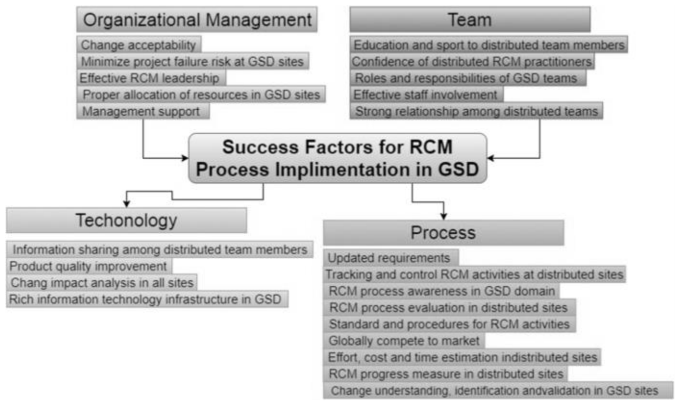

Organizational Management
Change acceptability
Minimize project failure risk at GSD sites
Effective RCM leadership
Proper allocation of resources in GSD sites
Management support
Team
Education and sport to distributed team members
Confidence of distributed RCM practitioners
Roles and responsibilities of GSD teams
Effective staff involvement
Strong relationship among distributed teams
Success Factors for RCM Process Implimentation in GSD
Techonology
Information sharing among distributed team members
Product quality improvement
Chang impact analysis in all sites
Rich information technology infrastructure in GSD
Process
Updated requirements
Tracking and control RCM activities at distributed sites
RCM process awareness in GSD domain
RCM process evaluation in distributed sites
Standard and procedures for RCM activities
Globally compete to market
Effort, cost and time estimation indistributed sites
RCM progress measure in distributed sites
Change understanding, identification andvalidation in GSD sites

Fig. 5. A theoretical model of the investigated success factors [29].

In [30]. Classification models were used to enhance the risk prediction of the agile software project. The authors used four models of classification: Support Vector Machine, k-nearest, ANN, and Random Forest. The dataset was collected by conducting a questionnaire and retrieved around 112 responses, then, it was divided into 80% for training and 20% for testing. The results showed that the Agile approach in the industry without a prediction system had a higher risk percentage than other approaches. Limitations and future work for the models: the dataset needs validation and evaluation by experts; evaluation metrics need to be enhanced and identify more methodologies in the agile approach. The complete process of building the ML models can be presented in Figure 6.

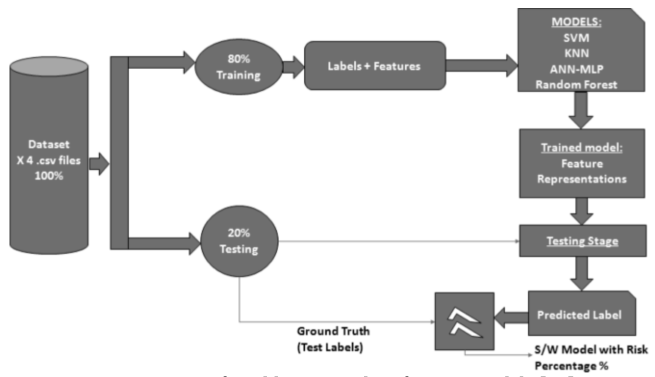


Fig. 6. Process of Building ML Classification models [25].

The authors in [31], developed a fuzzy logic hybridized framework for software risk prediction models during the decision-making process. Technique for Order of Preference by Similarity to Ideal Solution (IF-TOPSIS), Fuzzy DEMATEL, and crow search algorithm (CSA), optimized adaptive neuro-fuzzy inference system (ANFIS) were used for the software model prediction. The dataset consisted of 93 software projects, 70% used for training and the remaining used for testing and validating the model. The results showed that integrated fuzzy was accurate in software risk prediction. Limitations and future work: make a set of decisions and use many software factors and advanced machine learning techniques to improve and validate the results.

To reduce cost risks, the authors of [32] amplified the COCOMO-II in the GSD context. The dataset was collected by conducting a questionnaire and receiving around 175 responses. Evaluation metrics of the model were conducted by using Magnitude of Relative Estimates (MRE) measures and experts' judgment. Limitations and Future Work: the model is in an early stage and needs more validation, and mathematical or ML techniques may be used in the future.

In [33], The authors of this research conducted a literature review of machine learning models and techniques for risk prediction. The results showed that most machine learning (ML) techniques used for risk prediction were ANN, Fuzzy Logic, Genetic and Regression algorithms, and the results were promising in this area of study because it's still needed for more investigation and exploration. This research also provides a framework for recommendations that can be used for future work, as shown in Fig. 7.

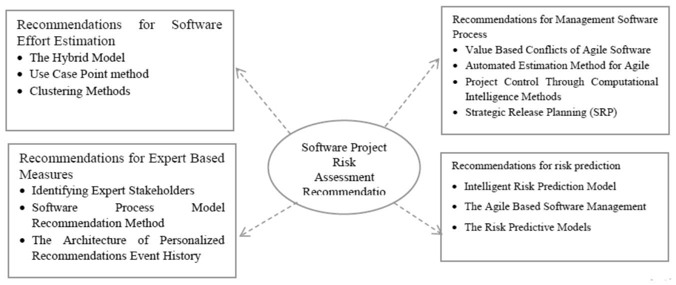

Recommendations for Software Effort Estimation
• The Hybrid Model
• Use Case Point method
• Clustering Methods
Recommendations for Management Software Process
• Value Based Conflicts of Agile Software
• Automated Estimation Method for Agile
• Project Control Through Computational Intelligence Methods
• Strategic Release Planning (SRP)
Software Project Risk Assessment Recommendatio
Recommendations for Expert Based Measures
• Identifying Expert Stakeholders
• Software Process Model Recommendation Method
• The Architecture of Personalized Recommendations Event History
Recommendations for risk prediction
• Intelligent Risk Prediction Model
• The Agile Based Software Management
• The Risk Predictive Models

Fig. 7 Software Project Risk Assessment Recommendation, to be used as a future work in ML Models [33].

In [34], the authors developed ML models for defect prediction in the domain of software reliability and performance. The models were built using ANN, RF, random tree (RT), decision table (DT), linear regression (LR), Gaussian processes (GP), SMOreg, and M5P. The dataset for these models was from the NASA promise repository. The results showed that the combination of different ML algorithms is effective in the prediction of software defects. Evaluation matrices used were (R^2), (MAE), (RMSE), (RAE), and (RRSE). Limitation and Future works: different datasets and ML algorithms can be used to evaluate the results. in addition, the investigation into more software factors to improve these results.

V. RESULTS

In this systematic literature review, 12 studies in software risk prediction using (ML) techniques and other algorithms were discussed in the domain of GSD, that studies were published from 2018 to 2022 as shown in Fig.8. In addition, table 2 represents a summary of this literature review. includes the following:

A. Dataset:

The dataset is the prerequisite step for training and testing ML models and algorithms. Some studies used private datasets, and others used a questionnaire or public dataset. Some papers, such as [8], [29], [30], [34], had some limitations in the dataset, and this may lead to inaccurate results.

B. Machine learning and other techniques:

Many ML algorithms and ANN were used in software risk prediction in the domain of GSD, while other studies used mathematical models to predict project costs and others used frameworks. But overall, most of these algorithms need more evaluation and improvement. They also need to be implemented in the real industry to validate their results.

C. Evaluation Model:

For the evaluation and estimation, studies depend on the expert's judgment and experience, such as: [14], [27], [29], [32]. Other studies used mean square error (MSE), magnitude of relative estimates (MRE), normalized root mean square error (NRMSE), comparative analysis.

D. Limitation and future work:

Most of the studies were needed to improve their models. widen their dataset and investigates more into the global software development factors and techniques.

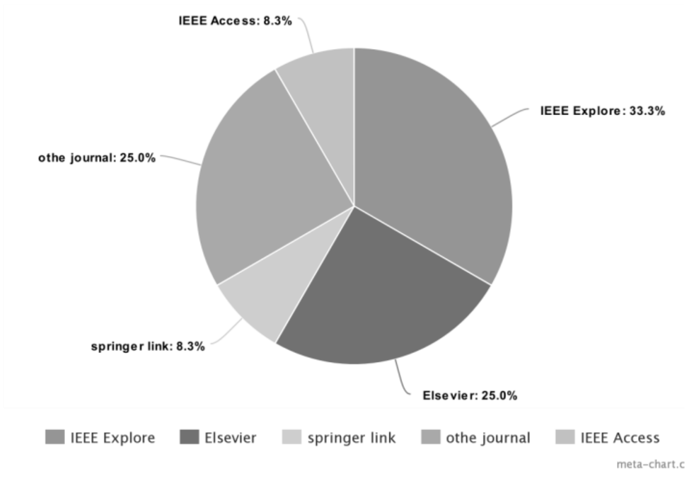


Fig. 8 Classification of studies discussed in this SLR.

The following are answers to questions listed in section II:

RQ1: many ML techniques and ANN were used in risk prediction, such as:

1. Levenberg–Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient.
2. Fuzzy Logic & Hybrid Fuzzy.
3. Analytical hierarchy process (AHP).
4. K-nearest, random forest and support vector machine.
5. Mathematical and context-historical algorithms.

RQ2: most of the studies used private datasets, and others used online datasets that are available online for improvement and enhancement. In addition, these studies were reviewed and discussed by experts.

RQ3: the top factors that affect software in the domain of global software development are still under investigation, but some studies listed these factors as the top factors affecting software:

1. Requirement change management.
2. Cost and profit.
3. Outsourcing team.
4. Organization type.
5. Technology.
6. Process of work.

RQ4: In [29], the authors categorized the software factors into four categories: team, organization, process, and technology, but this categorization needs to be validated and investigation needs to be conducted.

RQ5: ML techniques, ANN, and the fuzzy showed great efficiency and accuracy, but a common problem is that this model lacks industry testing with real data.

VI. CONCLUSION

GSD can be considered as a green field that still needs more investigation and exploration. It has a different characteristics than the normal software development process because of the nature of the GSD. Many studies were focused on this domain nowadays. In this study, we conducted a systematic literature review (SLR) that discussed12 studies in GSD and risk software prediction between 2018 and 2022. Summarizing the SLR as follows:

A. *Techniques & ML Models:*

Most studies used AI techniques, especially machine learning models and artificial neural network models, such as:

1. Levenberg–Marquard
2. Bayesian Regularization
3. Fuzzy Logic and hybrid Fuzzy
4. ML combined classifiers models ANN, SVM, K-Nearest, and random forest.
5. Amplified COCOMO-II Model
6. Scale Agile Framework (SAFe)
7. Analytical hierarchy process (AHP)

B. *Some limitations and issues related to proposed models:*

1. Availability and the sample size of the dataset.
2. The Low Accuracy of ML models.
3. Evaluation of the models and more investigation into software risk factors and practices.
4. Apply the models in the real industry.
5. Identifying factors in agile methodology that affect the overall GSD industry.

Table. 2. Summary of systematic Literature review algorithms and techniques used for software risk predictions

Reference	Dataset	ML Techniques and algorithms	Evaluation metrics	Limitation and Future work
(Filippetto et al, 2021) [14]	The dataset was collected based on 153 software projects from a financial company.	Risk analysis of the project by using its context history and project characteristics in the (SDLC).	Acceptance rate of 73% and an Accuracy rate of 83%, and these results were assessed by experts	1. Improve the proposed model methodology and accuracy. 2. Additional use of prototype. 3. Number of practitioners need to be increased and duration of the case study (5 weeks only)
(Iftikhar et al, 2021) [8]	The dataset was collected by sending 760 questionnaires to companies. 390 were received, and 116 were rejected, leaving 274 valid responses.	(ANN) model was created to predict the risk factors in GSD such as: Levenberg–Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient.	Least Mean Square Error (MSE)	1. the sample dataset needs to include many companies and random data collection should be used to generalize the model. 2. Deep learning should be used to get more accurate results.
(Sangaiah et al,2018) [25]	The dataset was collected by using 40 software projects from the real industry.	Hybrid Fuzzy DEMATEL–FMCDM–TODIM approach for evaluating software risk factors to find out and rank the software risk factors during the (SDLC).	Fuzzy Multi-Criteria Decision Making (FMCDM)	1. Include more factors 2. Expand this method 3. Use ANN and other techniques
(Sahu et al, 2018) [26]	The dataset was collected by using a dataset retrieved from John Musa of Bell Laboratories and received from the IEEE repository.	Fuzzy logic and ANN were used for building a software reliability prediction model.	Normalized Root Mean Square Error (NRMSE)	1. Model was restricted to one factor (time to failure). 2. Many software risk factors should be used to evaluate this model better.
(Khan et al,2021) [27]	The dataset was collected by conducting an online survey questionnaire.	Empirical investigation was conducted to figure out the top requirements of engineering (RE) practices GSD.	56 experts of GSD projects	1. Questionnaire was relied only on closed questions and focused on the company size. 2. Building framework to be used in the real industry.
(Marinho et al, 2021) [28]	The dataset was collected by reviewing 86 studies.	Scaling Agile projects in the domain of GSD. They mapped 44 agile practices to the SAFe Framework.	—	1. (AGSD) practices need to be evaluated and should also be tested in the real industry. 2. The mapped process of these practices needs to be evaluated.
(Akbar et al,2021) [29]	The dataset was collected by conducting a questionnaire survey and retrieved around 81 responses.	Fuzzy logic analytical hierarchy progress (FAHB) was used to prioritize the success factors that affect Requirement Change Management (RCM) in the GSD.	experts' judgment.	1. Sample size of the dataset needs to be widened. 2. Organization size and types should be considered. 3. Success factors, barriers, and best practices need more investigation and analysis.
(Gouthaman et al,2021) [30]	The dataset was collected by conducting a questionnaire and retrieved around 112 responses, then, it was divided into 80% for training and 20% for testing.	ML classifiers models ANN, SVM, K-Nearest, random forest were used to enhance the risk prediction of the agile software project.	—	1. The dataset needs validation and evaluation by experts. 2. Evaluation metrics need to be conducted and identify more methodologies in the agile approach.
(Suresh et al,2021) [31]	The dataset consisted of 93 software projects, 70% used for training and the remaining used for testing and validating the model.	Fuzzy logic hybridized framework for software risk prediction models during the decision-making process.	—	1. Make a group of decisions making and use sophisticated ML techniques 2. Use many software factors
(Khan et al,2021) [32]	The dataset was collected by conducting a questionnaire and receiving around 175 responses	Amplified COCOMO-II Model in the context of GSD.	Magnitude of Relative Estimates (MRE), experts' judgment	1. The model is in an early stage and needs more validation. 2. Mathematical or ML techniques may be used in the future.
(Mahdi et al,2020)[33]	—	Literature review of machine learning models and techniques for risk prediction.	—	Provides a framework for recommendations that can be used for future work.

(Assim et al,2020) [34]	The dataset for these models was from the NASA promise repository	ANN, RF, RT, decision table (DT), linear regression (LR), Gaussian processes (GP), SMOreg, and M5P were used for defect.	(R^2), (MAE), (RMSE), (RAE) and (RRSE)	1. different datasets and ML algorithms can be used to evaluate the results. 2. Investigation into more software factors to improve these results.

References

[1] Y. H. Shah *et al.*, "Communication Issues in GSD," 2012.

[2] S. Ali, H. Li, S. U. Khan, M. F. Abrar, and Y. Zhao, "Practitioner's view of barriers to software outsourcing partnership formation: An empirical exploration," *Journal of Software: Evolution and Process*, vol. 32, no. 5, May 2020, doi: 10.1002/smr.2233.

[3] IEEE Malaysia Section. Electron Devices Chapter, Universiti Kebangsaan Malaysia. Institute of Microengineering and Nanoelectronics, and Institute of Electrical and Electronics Engineers, "A Survey of Soft Computing Applications in Global Software Development."

[4] University of Management and Technology (Pakistan), Institute of Electrical and Electronics Engineers. Lahore Section., and Institute of Electrical and Electronics Engineers, *3rd International Conference on Innovative Computing (ICIC) : (IC)^2 2019 : 1st-2nd November 2019, Lahore, Pakistan*.

[5] M. A. Akbar, M. Shafiq, T. Kamal, and M. Hamza, "Towards the successful requirements change management in the domain of offshore software development outsourcing: Preliminary results," *International Journal of Computing and Digital Systems*, vol. 8, no. 3, pp. 205–215, May 2019, doi: 10.12785/ijcds/080301.

[6] M. Rizwan, J. Qureshi, A. Al-Zaidi, and R. Qureshi, "Global Software Development Geographical Distance Communication Challenges Article in International Arab Journal of Information Technology," 2017. [Online]. Available: https://www.researchgate.net/publication/308952993

[7] J. A. Khan, S. U. R. Khan, J. Iqbal, and I. U. Rehman, "Empirical Investigation about the Factors Affecting the Cost Estimation in Global Software Development Context," *IEEE Access*, vol. 9, pp. 22274–22294, 2021, doi: 10.1109/ACCESS.2021.3055858.

[8] A. Iftikhar, M. Alam, R. Ahmed, S. Musa, and M. M. Su'ud, "Risk Prediction by Using Artificial Neural Network in Global Software Development," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–25, Dec. 2021, doi: 10.1155/2021/2922728.

[9] M. Yaseen and Z. Ali, "Success Factors during Requirements Implementation in Global Software Development: A Systematic Literature Review," *International Journal of Computer Science and Software Engineering (IJCSSE)*, vol. 8, no. 3, 2019, [Online]. Available: www.IJCSSE.org

[10] B. J. Galli, "Addressing Risks in Global Software Development and Outsourcing," *International Journal of Risk and Contingency Management*, vol. 7, no. 3, pp. 1–41, May 2018, doi: 10.4018/ijrcm.2018070101.

[11] A. Iftikhar, M. Alam, S. Musa, M. Mohd, and S. ' Ud, "Trust Development in Virtual teams to Implement Global Software Development (GSD): A Structured Approach to Overcome Communication Barriers."

[12] B. G. Tavares, C. E. S. da Silva, and A. D. de Souza, "Risk management analysis in Scrum software projects," *International Transactions in Operational Research*, vol. 26, no. 5, pp. 1884–1905, Sep. 2019, doi: 10.1111/itor.12401.

[13] ACM Sigsoft. and Institute of Electrical and Electronics Engineers., *ESEM 2009 : 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM : October 15-15, 2009, Lake Buena Vista, Florida USA.* IEEE, 2009.

[14] 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," *Information and Software Technology*, vol. 131, Mar. 2021, doi: 10.1016/j.infsof.2020.106497.

[15] "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[16] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Information and Software Technology*, vol. 51, no. 1. pp. 7–15, Jan. 2009. doi: 10.1016/j.infsof.2008.09.009.

[17] B. Kitchenham, "Procedures for Performing Systematic Reviews," 2004.

[18] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Information and Software Technology*, vol. 51, no. 6. pp. 957–976, Jun. 2009. doi: 10.1016/j.infsof.2008.12.005.

[19] R. Jain and U. Suman, "A Project Management Framework for Global Software Development," *ACM SIGSOFT Software Engineering Notes*, vol. 43, no. 1, pp. 1–10, Mar. 2018, doi: 10.1145/3178315.3178329.

[20] R. Prikladnicki, J. L. N. Audy, and R. Evaristo, "A reference model for global software development," in *IFIP Advances in Information and Communication Technology*, 2004, vol. 149, pp. 369–377. doi: 10.1007/1-4020-8139-1_39.

[21] J. Menezes, C. Gusmão, and H. Moura, "Risk factors in software development projects: a systematic literature review," *Software Quality Journal*, vol. 27, no. 3. Springer New York LLC, pp. 1149–1174, Sep. 01, 2019. doi: 10.1007/s11219-018-9427-5.

[22] R. Britto, V. Freitas, E. Mendes, and M. Usman, “Effort estimation in global software development: A systematic literature review,” in *Proceedings - 2014 IEEE 9th International Conference on Global Software Engineering, ICGSE 2014*, Oct. 2014, pp. 135–144. doi: 10.1109/ICGSE.2014.11.

[23] A. N. Okon, S. E. Adewole, and E. M. Uguma, “Artificial neural network model for reservoir petrophysical properties: porosity, permeability and water saturation prediction,” *Modeling Earth Systems and Environment*, vol. 7, no. 4, pp. 2373–2390, Nov. 2021, doi: 10.1007/s40808-020-01012-4.

[24] S. K. Niranjan, V. N. M. Aradhya, Amity University, IEEE-USA, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section, and Institute of Electrical and Electronics Engineers, *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) : 14-17 December 2016, Noida, India*.

[25] A. K. Sangaiah, O. W. Samuel, X. Li, M. Abdel-Basset, and H. Wang, “Towards an efficient risk assessment in software projects–Fuzzy reinforcement paradigm,” *Computers and Electrical Engineering*, vol. 71, pp. 833–846, Oct. 2018, doi: 10.1016/j.compeleceng.2017.07.022.

[26] K. Sahu and R. K. Srivastava, “Soft computing approach for prediction of software reliability,” *ICIC Express Letters*, vol. 12, no. 12, pp. 1213–1222, Dec. 2018, doi: 10.24507/icicel.12.12.1213.

[27] H. U. Khan, M. Niazi, M. El-Attar, N. Ikram, S. U. Khan, and A. Q. Gill, “Empirical Investigation of Critical Requirements Engineering Practices for Global Software Development,” *IEEE Access*, vol. 9, pp. 93593–93613, 2021, doi: 10.1109/ACCESS.2021.3092679.

[28] M. Marinho, R. Camara, and S. Sampaio, “Toward unveiling how safe framework supports agile in global software development,” *IEEE Access*, vol. 9, pp. 109671–109692, 2021, doi: 10.1109/ACCESS.2021.3101963.

[29] M. A. Akbar, M. Shameem, A. A. Khan, M. Nadeem, A. Alsanad, and A. Gumaei, “A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development,” *Journal of Software: Evolution and Process*, vol. 33, no. 2, Feb. 2021, doi: 10.1002/smr.2292.

[30] G. P and S. Sankaranarayanan, “Prediction of Risk Percentage in Software Projects by Training Machine Learning Classifiers,” *Computers and Electrical Engineering*, vol. 94, Sep. 2021, doi: 10.1016/j.compeleceng.2021.107362.

[31] K. Suresh and R. Dillibabu, “An integrated approach using IF-TOPSIS, fuzzy DEMATEL, and enhanced CSA optimized ANFIS for software risk prediction,” *Knowledge and Information Systems*, vol. 63, no. 7, pp. 1909–1934, Jul. 2021, doi: 10.1007/s10115-021-01573-5.

[32] J. A. Khan, S. U. R. Khan, T. A. Khan, and I. U. R. Khan, “An Amplified COCOMO-II Based Cost Estimation Model in Global Software Development Context,” *IEEE Access*, vol. 9, pp. 88602–88620, 2021, doi: 10.1109/ACCESS.2021.3089870.

[33] M. N. Mahdi, M. Z. Mohamed, A. Yusof, L. K. Cheng, M. S. Mohd Azmi, and A. R. Ahmad, “Design and Development of Machine Learning Technique for Software Project Risk Assessment - A Review,” in *2020 8th International Conference on Information Technology and Multimedia, ICIMU 2020*, Aug. 2020, pp. 354–362. doi: 10.1109/ICIMU49871.2020.9243459.

[34] M. Assim, Q. Obeidat, and M. Hammad, “Software Defects Prediction using Machine Learning Algorithms,” Oct. 2020. doi: 10.1109/ICDABI51230.2020.9325677.