International Journal of Engineering & Technology, 7 (4.15) (2018) 304-310

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Application of Soft Computing Techniques in Global Software Development: state-of-the-art Review

Asim Iftikhar[1], Shahrulniza Musa[2*], Muhammad Alam[3]
Mazliham Mohd Su'ud[4], Syed Mubashir Ali[5]

[1,2]Malaysian Institute of Information Technology (MIIT), University of Kuala Lumpur (UniKL), Kuala Lumpur, Malaysia
[1,3,5]College of Computer Science & Information Systems, Institute of Business Management (IoBM), Karachi, Pakistan
[4]City Campus, University of Kuala Lumpur (UniKL), Kuala Lumpur, Malaysia
**Corresponding author E-mail: shahrulniza@unikl.edu.my*

Abstract

Developing Software through a globally distributed team is a modern trend, which is not only cost effective but also yields best project results mitigating risk and increasing return on investment. This is easily achieved by ensuring through put in production is maintained at all times irrespective of the clock time and geographical boundaries. This shift of phenomenon is happening across the board as more and more companies use this as a strategic tool. Modern day technology makes this all possible, without compromising quality, coding practices and project management techniques. In this paper we have researched several papers (2008 to 2018) and understood the data for soft computing to provide a strong basis for future directions

Keywords: *Software Development; Global Software Development; Soft Computing; Distributed Team.*

1. Introduction

In the past century, technological advancements have drastically altered the outlook of the professional world [1], [2]. The old days are now gone when people from one part of the world were not able to communicate or it was difficult for them to communicated with the people in the other part. Global village is not just restaurants or tall buildings but the sophisticated communication channels that makes it all possible!

Software industry is expanding at a very exponential growth rate. It now forms a major stake in the global economy. Some call it GSD some call it distributed computing. No matter what the name, the underlying principles are the same [1], [3]. Many organizations and software houses are adopting GSD environment throughout the world where there is huge demand of skilled people, which is unfortunately not readily available [4].

As time progresses and more regulators are formed, we will realize far more advantages of this technique, to name a few, time to ship a ready product, round the clock utilization of resources efficiently, better product quality, low cost of labor, access to a multitude of skill set across the globe [3], [5].

GSD is not something that simple. It does have a couple of inherent challenges under the umbrella, which should be realized earlier in the process of implementation. Distributed teams involve a good risk management practice in place, because you are dealing with individuals who are not only geographically dispersed with communication and collaboration issues [6]–[8], but also culturally and linguistically. One person or a central management, with one style of management or attitude may not work across borders. A style of management that works with people of culture A will or may turn tables upside down with people of culture B. This is a very inherent risk that comes under the belt!. It doesn't stop here, you could develop a high level of mistrust leading to miscommitment, inappropriate task timings and distribution since managing remote resources automatically eliminates the personal touch you get when teams are working on the same floor side by side. The sense to determine an individual's capability eliminates automatically, to a high extent. With time human resources are perceived like robotic resources, who can be popped tasks, and expect to deliver them - like a machine. This poses a big risk when working in GSD type of environment and must be dealt with early in the project management phases [9].

Soft computing (SC) techniques and algorithms deals with situations where there is uncertainty and ambiguity and helps in forecasting, optimization and decision making in real life situations [10]. A number of researchers have discussed about the importance of using SC techniques within GSD processes. SC is a set of techniques and algorithms including but not limited to Fuzzy Sets (FZ), Rough Set Theory (RSS), Artificial Neural Network (ANN), Genetic Algorithm (GA), Bayesian Networks (BN), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Firefly algorithm (FA) etc. SC techniques are useful in situations having imprecision, ambiguity, uncertainty and partial truth [11]. These techniques have numerous applications in a number of areas including software engineering. The main idea of using SC techniques is to get desired results in real life uncertain situations with minimum cost.

The paper has been distributed into sections which are as follows: Section 2 briefly discuss related work. Section 3 will discuss about Global software development (GSD) and its major processes. Section 4 will describe the result and discussions. Last section will conclude this paper along with discussing future research directions.

2. Related Work

In [9], A risk management process was introduced by researchers in virtual projects and performed qualitative and quantitative analysis. Researchers took expert views using questionnaire in quantitative analysis about the possibilities and occurrences of risk factors and their severe effects on the project goals. Decision Analysis Matrix technique was used to prioritize risk factors. In order to evaluate the risks associated with projects throughout project lifecycle the Fuzzy Linear Programming Model was incorporated in quantitative analysis. A method is introduced by researchers for developing appropriate strategies of reacting to each risk factor and implemented a model so as to select appropriate strategies and finally tested it on an example.

In [12] researchers analysed the possible cost of long term risk related to product quality and competitiveness of technology firms. The research has suggested that design and analysis phase with GSD should be carried out locally whereas unit testing and software implementation should be done globally across and around all the virtual teams and integration of different modules being developed must be perform centrally.

Authors in [13] discussed about how important and significant risk management is when it comes to GSD projects. To manage risks and gaining greater insight about the role of decision making in strategic, tactical and operational, some approaches had been discussed. With the perspective of strategic and tactical, software organizations could create a model named Offshore Distribution model which is used to perform risk analysis and risk assessment. Project risk management process is adopted for the decision taken at operational level

Researchers in [14] did a comparative study between fuzzy inference systems, adaptive neuro fuzzy inference system and ANN in order develop model for predicting cost associated with software maintenance. The research concluded that among the three techniques being compared, adaptive neuro fuzzy inference system outperforms other models and provides with the most accurate results.

Another research [15] proposed and discussed two techniques which include fuzzy expert system using genetic algorithm along with multilinear discriminant analysis in order to develop models for quality prediction. They used genetic algorithm to automatically generate fuzzy rules so as to improve the accuracy along with reducing cost. Their created model resulted in improved quality prediction.

In [16], the researchers developed and presented various ensemble models for predicting reliability of software with improved accuracy. They used a number of statistical techniques from multiple linear regressions to multivariate adaptive regression splines along with integrating soft computing techniques such as backpropagation neural network and dynamic evolving neuro fuzzy inference system into their ensemble models. They performed experiments on software reliability using the data from the available literature. The result showed that non-linear ensemble model proved to be a better choice compared to other statistical and AI based techniques.

Researchers in [17] discussed about software project risk analysis using Bayesian network. They utilised the data collected from 302 software projects and demonstrated that their suggested model performs better compared to other predictive techniques and algorithms. They also proposed and discussed another framework for risk causality analysis of software projects and project risk management.

In [18], authors described the importance of risk management when software product is being developed. It is important to focus and address risks involved in software development process. Otherwise they can create undesired results. Researchers also presented a software tool based upon Fuzzy Cognitive Map (FCM) to analysis risks. FCMs describe different concepts with different aspects of the behaviour of complex systems. The tool analyses risks using three cases. In case 1, Poor Management, Market Competition, Deadline Pressure, Wrong Documents, Development of wrong user interface and straining computer science capabilities factors are selected. In case 2, environmental failure, changes in customer requirements, lack of user support, incorrect matrices and lack of training factors are selected. Finally in case 3, shortfall in supplied components, adding unnecessary features, developing wrong S/W functions factors are selected. After this, chances of personnel shortfall, technical failure, poor software quality and project failure is presented as output.

Researchers in [19] introduced a risk assessment methodology in light of fuzzy set theory, an effective tool to deal with risks, and Analytic Hierarchy Process (AHP), used to structure huge amount of risks. In this methodology authors gathered knowledge and experience from many research gurus. To deal with huge amount of risks a hierarchical weighting method has been developed to assess risks weights. The weighting method includes an algorithm to cut the irregularity and to avoid ambiguous solutions. The approach provides a simple and effective mechanism for modelling risk assessment problems involving subjective evaluations of the members in the risk assessment group.

3. Global Software Development

According to literature, SC techniques have a lot of applications in GSD projects and therefore have been extensively used by a number of researchers which is shown in Table 1. In a GSD setup, the software houses hire developers and other team members from all over the world. The team not only works round the clock due to being at dispersed geographical locations but also provides round the clock customer service. As a result, projects risk management[9], maintenance prediction [14], software quality prediction[15], software reliability prediction [16], schedule estimation [20], [21] and decision making within GSD context can be resolved using SC techniques. By using less office space and other resources, a company greatly reduces its labor and other overheads thus saving them money. Also this model helps virtual teams to have the right attitude towards work and performs their assigned task more quickly.

Following are some of the major processes within global software development:

3.1. Risk Management

The likelihood of losing or gaining something having a value is often referred to as risk. Software risk management mainly deals with quantification of risk which includes defining risk, probability risk, loss as a result of risk and potential risk liability. A number of risks are associated with GSD which includes [22]:

- Distances both temporal and spatial among virtual teams
- Cultural differences including attitude and ways of requirement elicitation

- Use of different tools and software development models / techniques by dispersed teams
- Risk factors related to technology
- Software maintenance issues (customer and technical support, software documentation etc)
- Software issues related to its performance (while different modules from different teams are integrated together)

3.2. Software Maintenance

One of the major concerns within GSD environment is software maintenance [23]. In order to develop a product that is easy to maintain in longer run, consistent and complete software documentation is imperative [24] which is a big challenge to implement in GSD. There is a need to have a service level agreement with the customer so as to provide the required technical support [25].

3.3. Software Reliability

The ability of any software to perform all of its functions when the software is live without experiencing any downtime or failure is referred to as software reliability. It may also be defined as the probability of software to carry out its functions for a given period of time without any failure [16]. Software reliability is said to be growing when with time, there are less software failures while carrying out its full functions [26].

3.4. Software Quality

Software quality is defined as software having a reliable, accurate, timely and error-free operation within the budget as well as meeting the requirements (both functional and non-functional) and needs of the customers. In other words, software quality is nothing but a preferred blend of characteristics including but not limited to maintainability, reliability, portability, efficiency and reusability [27].

3.5. Software Fault

A fault in a software product is a bug, imperfection that emerges when the normal outcome don't coordinate with the genuine outcomes. An incorrect or wrong step, process, piece of code or a physical shortcoming that arises in software is referred to as software fault. Most software faults arise from errors made by developers. The main cause of any software failure is usually the software fault [28].

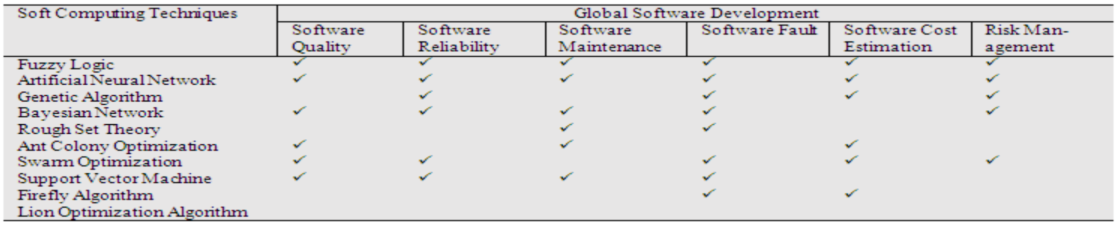
3.6. Software Cost Estimation

The effort to perform work as well as predicting the schedule to perform the work are an important aspect when it comes to software development and this whole process is referred to as software cost estimation. One of the most important and critical process when it comes to software products management [29]. Software development is a complex activity and a Business manager wants this more manageable as far as time and budget is concerned and has become an essential investment for many software firms. Software cost estimation models are required that predict, monitor, control and assess software development effectively.

4. Result and discussions

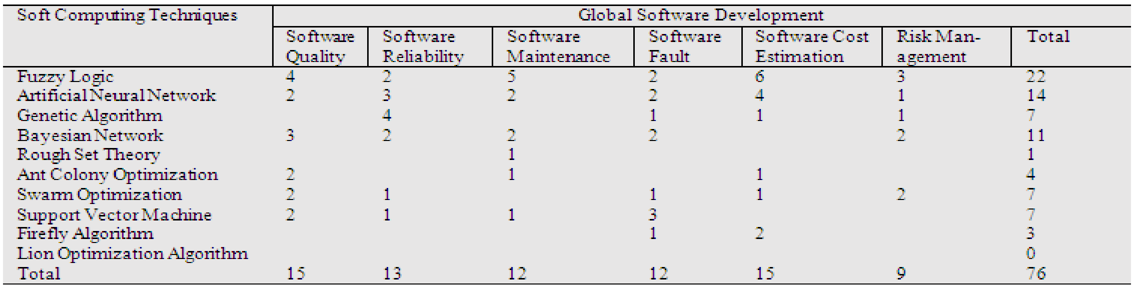
An extensive survey of peer-reviewed articles is being carried out. Table 1 and table 2 shows from different perspectives about various soft computing techniques and algorithms being used in a number of global software development processes.

Table 1: SC techniques used in GSD



Soft Computing Techniques	Global Software Development					
	Software Quality	Software Reliability	Software Maintenance	Software Fault	Software Cost Estimation	Risk Management
Fuzzy Logic	✓	✓	✓	✓	✓	✓
Artificial Neural Network	✓	✓	✓	✓	✓	✓
Genetic Algorithm		✓		✓	✓	✓
Bayesian Network	✓	✓	✓	✓		✓
Rough Set Theory			✓	✓		
Ant Colony Optimization	✓		✓		✓	
Swarm Optimization	✓	✓		✓	✓	✓
Support Vector Machine	✓	✓	✓	✓		
Firefly Algorithm				✓	✓	
Lion Optimization Algorithm						

Table 2: Frequency of usage of various soft computing techniques in GSD



Soft Computing Techniques	Global Software Development						
	Software Quality	Software Reliability	Software Maintenance	Software Fault	Software Cost Estimation	Risk Management	Total
Fuzzy Logic	4	2	5	2	6	3	22
Artificial Neural Network	2	3	2	2	4	1	14
Genetic Algorithm		4		1	1	1	7
Bayesian Network	3	2	2	2		2	11
Rough Set Theory			1				1
Ant Colony Optimization	2		1		1		4
Swarm Optimization	2	1		1	1	2	7
Support Vector Machine	2	1	1	3			7
Firefly Algorithm				1	2		3
Lion Optimization Algorithm							0
Total	15	13	12	12	15	9	76

After an extensive literature review, it is evident that ANN, FL, BN, PSO are the SC techniques being widely used within software development processes as shown in figure 1 and figure 2. FL technique is used by several authors in their research articles [14], [16], [18], [19], [27], [30]–[46]. ANN technique in the field of software development is used in research articles that were found [16], [47]–[58]. BN is also a widely used technique in several research papers reviewed during survey [17], [59]–[67]. Many authors used PSO technique in their research [45], [46], [53], [68]–[71]. Ant Colony optimization, Firefly Algorithm, Genetic Algorithm, Rough Set theory and Support Vector Machine are rarely used in the field of software development as shown in figure 3. In ACO technique only few research articles were found [72]–[75]. GA is a famous soft computing techniques but have been used sparingly in the area of software engineering and development and in this area again insufficient work has been carried out [26], [75]–[80]. FA and RSS are also a least discussed area in the field of software development and very few researches found [81]–[84].

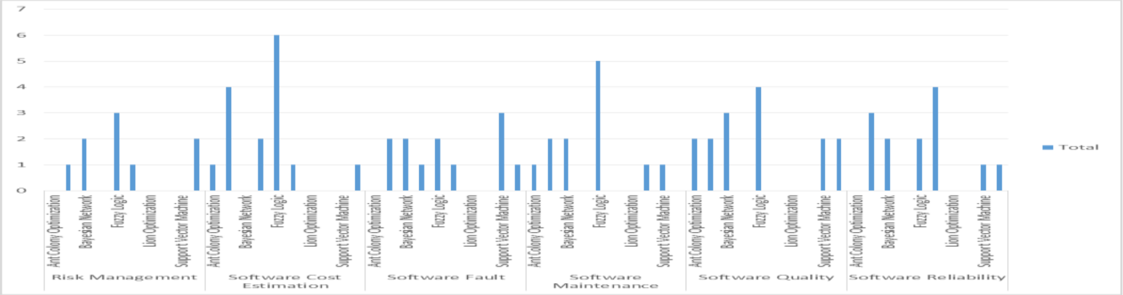


Fig. 1: No of papers using soft computing techniques in GSD

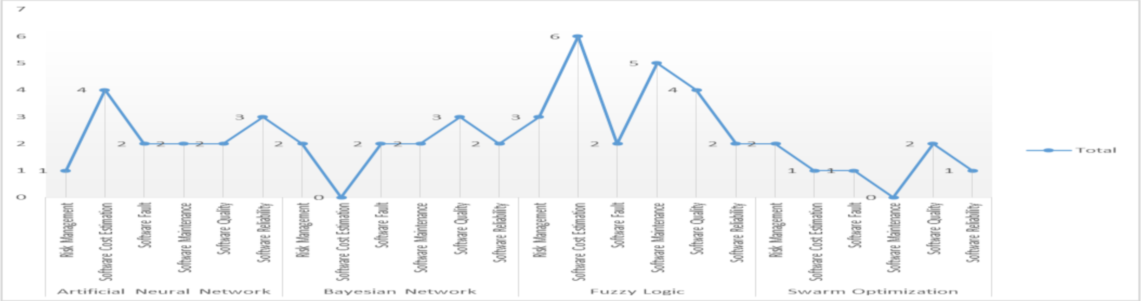


Fig. 2: Widely used soft computing techniques in GSD

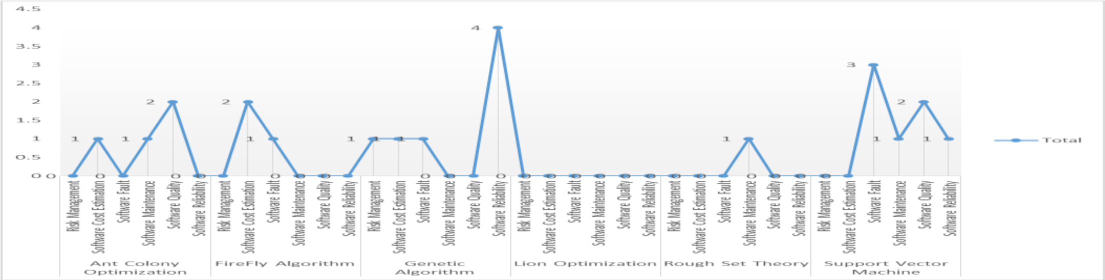


Fig. 3: Rarely used soft computing techniques in GSD

5. Conclusion

This study has done an extensive review of the available literature. A number of SC techniques have been applied into various GSD process including but not limited to management of software risk, software quality, software maintenance, software reliability and software project management. It has been found out that FL, ANN and GA are the most widely used SC techniques. And when it comes to GSD process, software development and risk management are the least addressed by the researchers.

Since very few researchers have use SVM and ACO which creates a research gap within this area. Genetic Algorithm, Rough Set theory, Firefly algorithm and Lion optimization algorithm are the techniques which have not been used in GSD and thus can be seen as a huge research gap and future research can be carried out in this area.

References

[1] Shah, Yasir Hassan, Mushtaq Raza, and Sani UlHaq. "Communication issues in GSD." *International Journal of Advanced Science and Technology* 40.2012 (2012): 69-76 , available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.640.1760&rep=rep1&type=pdf

[2] Ali, Syed Mubashir, et al. "University Students' Perception on the Impact of 3G Mobile Broadband in Pakistan-A Survey." *Research Inventy: International journal of Engineering and Science* 5.2 (2015), available online: http://www.academia.edu/download/36988940/A5200105.pdf

[3] Al-Zaidi, Areej, and Rizwan Qureshi. "Global software development geographical distance communication challenges." *Int. Arab J. Inf. Technol.* 14.2 (2017): 215-222, available online: http://www.ccis2k.org/iajit/PDF/Vol%2014,%20No.%202/8063.pdf

[4] Iftikhar, Asim, et al. "Trust Development in virtual teams to implement global software development (GSD): A structured approach to overcome communication barriers." *Engineering Technologies and Social Sciences (ICETSS), 2017 IEEE 3rd Interna-*

tional Conference on. IEEE, 2017. https://doi.org/10.1109/ICETSS.2017.8324169

[5] Anjum, Maria, Muhammad Islam Zafar, and Syed Atif Mehdi. "Establishing guidelines for management of virtual teams." *IADIS Virtual Multi Conference on Computer Science and Information Systems (Software Engineering and Applications)*. 2006.

[6] Casey, Valentine. "Imparting the importance of culture to global software development." *ACM inroads* 1.3 (2010): 51-57, available online: https://dl.acm.org/citation.cfm?id=1835443

[7] Wu, Shujian. "Overview of communication in global software development process." *Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference on*. IEEE, 2012. https://doi.org/10.1109/SOLI.2012.6273583

[8] Casey, Valentine, and Ita Richardson. "Implementation of Global Software Development: a structured approach." *Software Process: Improvement and Practice* 14.5 (2009): 247-262, available online: https://onlinelibrary.wiley.com/doi/full/10.1002/spip.422

[9] Ghaffari, Mona, Farrokh Sheikhahmadi, and Gholamreza Safakish. "Modeling and risk analysis of virtual project team through project life cycle with fuzzy approach." *Computers & Industrial Engineering* 72 (2014): 98-105, available online: https://www.sciencedirect.com/science/article/abs/pii/S0360835214000540

[10] S. A. Burney, S. M. Ali, and S. Burney, "A survey of soft computing applications for decision making in supply chain management," in *Engineering Technologies and Social Sciences (ICETSS), 2017 IEEE 3rd International Conference on*, 2017, pp. 1–6. https://doi.org/10.1109/ICETSS.2017.8324158

[11] A. Iftikhar, S. Musa, M. Alam, M. M. Su'ud and S. M. Ali, "A survey of soft computing applications in global software development," *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, Bangkok, Thailand, 2018, pp. 1-4. https://doi.org/10.1109/ICIRD.2018.8376330

[12] Yu, Liguo, and Alok Mishra. "Risk analysis of global software development and proposed solutions." *Automatika* 51.1 (2010): 89-98, available online: http://www.tandfonline.com/doi/pdf/10.1080/00051144.2010.11828358

[13] Prikladnicki, Rafael, and Marcelo Hideki Yamaguti. "Risk management in global software development: A position paper." *Third International Workshop on Global Software Development*. 2004. http://dx.doi.org/10.1049/ic:20040306

[14] Kaur, Arvinder, Kamaldeep Kaur, and Ruchika Malhotra. "Soft computing approaches for prediction of software maintenance effort." *International Journal of Computer Applications* 1.16 (2010), available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.3368&rep=rep1&type=pdf

[15] Baisch, Ekkehard, and Thomas Liedtke. "Comparison of conventional approaches and soft-computing approaches for software quality prediction." *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation. 1997 IEEE International Conference on*. Vol. 2. IEEE, 1997. https://doi.org/10.1109/ICSMC.1997.638086

[16] Kiran, N. Raj, and Vadlamani Ravi. "Software reliability prediction by soft computing techniques." *Journal of Systems and Software* 81.4 (2008): 576-583, available online: https://www.sciencedirect.com/science/article/pii/S0164121207001227

[17] Hu, Yong, et al. "Software project risk analysis using Bayesian networks with causality constraints." *Decision Support Systems* 56 (2013): 439-449, available online: https://www.sciencedirect.com/science/article/pii/S0167923612003338

[18] Bhatia, Nitin, and Namarta Kapoor. "Fuzzy cognitive map based approach for software quality risk analysis." *ACM SIGSOFT Software Engineering Notes* 36.6 (2011): 1-9, available online: https://dl.acm.org/citation.cfm?id=2047422

[19] A. Nieto-Morote and F. Ruz-Vila, "A fuzzy approach to construction project risk assessment," *International Journal of Project Management*, vol. 29, no. 2, pp. 220–231, 2011, available online: https://www.sciencedirect.com/science/article/pii/S0263786310000268

[20] S. Hartmann, "A competitive genetic algorithm for resource-constrained project scheduling," *Naval Research Logistics (NRL)*, vol. 45, no. 7, pp. 733–750, 1998, available online: https://www.sciencedirect.com/science/article/pii/S0263786310000268

[21] Sheta, Alaa, David Rine, and Aladdin Ayesh. "Development of software effort and schedule estimation models using soft computing techniques." *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE, 2008. https://doi.org/10.1109/CEC.2008.4630961

[22] Chadli, Saad Yasser, et al. "Identifying risks of software project management in Global Software Development: An integrative framework." *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*. IEEE, 2016. https://doi.org/10.1109/AICCSA.2016.7945664

[23] Pilatti, Leonardo, and Jorge Luis Nicolas Audy. "Global software development offshore insourcing organizations characteristics: Lessons learned from a case study." *Global Software Engineering, 2006. ICGSE'06. International Conference on*. IEEE, 2006. https://doi.org/10.1109/ICGSE.2006.261244

[24] Prikladnicki, Rafael, Jorge Luis Nicolas Audy, and Roberto Evaristo. "A reference model for global software development: findings from a case study." *Global Software Engineering, 2006. ICGSE'06. International Conference on*. IEEE, 2006. https://doi.org/10.1109/ICGSE.2006.261212

[25] g Yan, Zhen. "Efficient Maintenance Support in Offshore Software Development: a Case Study on a Global E-Commerce Project." *The 3rd International Workshop on Global Software Development* (2004). http://gsd2004.cs.uvic.ca/docs/proceedings.pdf#page=15

[26] Aljahdali, Sultan H., and Mohammed E. El-Telbany. "Software reliability prediction using multi-objective genetic algorithm." *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*. IEEE, 2009. https://doi.org/10.1109/AICCSA.2009.5069339

[27] Mittal, Harish, and Pradeep Bhatia. "Software maintainability assessment based on fuzzy logic technique." *ACM SIGSOFT Software Engineering Notes* 34.3 (2009): 1-5, available online: https://dl.acm.org/citation.cfm?id=1527210

[28] Erturk, Ezgi, and Ebru Akcapinar Sezer. "A comparison of some soft computing methods for software fault prediction." *Expert Systems with Applications* 42.4 (2015): 1872-1879, available online: https://www.sciencedirect.com/science/article/pii/S0957417414006496

[29] Idri, Ali, Alain Abran, and T. M. Khoshgoftaar. "Fuzzy analogy: A new approach for software cost estimation." *International Workshop on Software Measurement*. 2001.

[30] K. Khatatneh and T. Mustafa, "Software reliability modeling using soft computing technique," *European Journal of Scientific Research*, vol. 26, no. 1, pp. 154–160, 2009, available online: https://www.researchgate.net/profile/Khatatneh_Khalaf/publication/241821620_Software_Reliability_Modeling_Using_Soft_Computing_Technique/links/58a769b1aca27206d9ac3e71/Software-Reliability-Modeling-Using-Soft-Computing-Technique.pdf

[31] Yadav, Harikesh Bahadur, and Dilip Kumar Yadav. "A fuzzy logic based approach for phase-wise software defects prediction using software metrics." *Information and Software Technology* 63 (2015): 44-57, available online: https://www.sciencedirect.com/science/article/pii/S095058491500052X

[32] H. Yang, "Measuring software product quality with ISO standards base on fuzzy logic technique," in *Affective Computing and Intelligent Interaction*, Springer, 2012, pp. 59–67, available online:https://link.springer.com/chapter/10.1007/978-3-642-27866-2_8

[33] Challa, Jagat Sesh, et al. "Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach." JIPS 7.3 (2011): 473-518. available Online, http://arindampaul.yolasite.com/resources/JIPS_v07_no3_paper07.pdf

[34] Singh, Yogesh, Pradeep Kumar Bhatia, and Omprakash Sangwan. "Predicting software maintenance using fuzzy model." *ACM SIGSOFT Software Engineering Notes* 34.4 (2009): 1-6, available online https://dl.acm.org/citation.cfm?id=1543425

[35] Mittal, Harish, and Pradeep Bhatia. "Software maintainability assessment based on fuzzy logic technique." *ACM SIGSOFT Software Engineering Notes* 34.3 (2009): 1-5. https://dl.acm.org/citation.cfm?id=1527210

[36] Lu, Kun-Yung, and Chun-Chin Sy. "A real-time decision-making of maintenance using fuzzy agent." *Expert Systems with Applications* 36.2 (2009): 2691-2698, available online: https://www.sciencedirect.com/science/article/pii/S0957417408000365

[37] Mittal, J. P., Pradeep Bhatia, and Harish Mittal. "Software maintenance productivity assessment using fuzzy logic." *ACM SIGSOFT Software Engineering Notes* 34.5 (2009): 1-4. https://dl.acm.org/citation.cfm?id=1598739

[38] Pandey, Ajeet Kumar, and N. K. Goyal. "A fuzzy model for early software fault prediction using process maturity and software metrics." *International Journal of Electronics Engineering* 1.2 (2009): 239-245, available online: https://www.researchgate.net/profile/Neeraj_Goyal3/publication/230739115_A_Fuzzy_Model_for_Early_Software_Fault_Prediction_Using_Process_Maturity_and_Software_Metrics/links/0fcfd503c6b13f1347000000/A-Fuzzy-Model-for-Early-Software-Fault-Prediction-Using-Process-Maturity-and-Software-Metrics.pdf

[39] Pandey, Ajeet Kumar, and Neeraj Kumar Goyal. "Predicting fault-prone software module using data mining technique and fuzzy logic." *International Journal of Computer and Communication Technology* 2.2 (2010): 56-63, available online: https://www.researchgate.net/profile/Neeraj_Goyal3/publication/228744278_Predicting_Fault-prone_Software_Module_Using_Data_Mining_Technique_and_Fuzzy_Logic/links/0fcfd50921eaa65718000000/Predicting-Fault-prone-Software-Module-Using-Data-Mining-Technique-and-Fuzzy-Logic.pdf

[40] Mittal, Anish, Kamal Parkash, and Harish Mittal. "Software cost estimation using fuzzy logic." *ACM SIGSOFT Software Engineering Notes* 35.1 (2010): 1-7. https://dl.acm.org/citation.cfm?id=1668866

[41] Attarzadeh, Iman, and Siew Hock Ow. "Improving the accuracy of software cost estimation model based on a new fuzzy logic model." *World applied sciences Journal* 8.2 (2010): 177-184, available online: https://www.researchgate.net/profile/Jack_Son6/post/Need_latest_articles_of_Software_Cost_Estimation_of_FPA_using_Fuzzy_Logic/attachment/5a3bf1ecb53d2f0bba4605d4/AS:574060074422272@1513877995874/download/Improving_the_Accuracy_of_Software_Cost.pdf

[42] Andreou, Andreas S., and Efi Papatheocharous. "Software cost estimation using fuzzy decision trees." *Automated Software Engineering, 2008. ASE 2008. 23rd IEEE/ACM International Conference on*. IEEE, 2008. http://ieeexplore.ieee.org/abstract/document/4639344/

[43] Kazemifard, Mohammad, et al. "Fuzzy emotional COCOMO II software cost estimation (FECSCE) using multi-agent systems." *Applied Soft Computing* 11.2 (2011): 2260-2270, available online: https://www.sciencedirect.com/science/article/pii/S1568494610002085

[44] Du, Wei Lin, Danny Ho, and Luiz Fernando Capretz. "Improving software effort estimation using neuro-fuzzy model with SEER-SEM." *arXiv preprint arXiv:1507.06917* (2015), available online: https://arxiv.org/abs/1507.06917

[45] PVGD, Prasad Reddy. "Particle swarm optimization in the fine-tuning of fuzzy software cost estimation models." *International Journal of Software Engineering (IJSE)* 1.2 (2010): 12-23, available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.7693&rep=rep1&type=pdf

[46] Chang Lee, Kun, Namho Lee, and Honglei Li. "A particle swarm optimization-driven cognitive map approach to analyzing information systems project risk." *Journal of the Association for Information Science and Technology* 60.6 (2009): 1208-1221, available online: http://onlinelibrary.wiley.com/doi/10.1002/asi.21019/full

[47] Zheng, Jun. "Predicting software reliability with neural network ensembles." *Expert systems with applications* 36.2 (2009): 2116-2122, available online: https://www.sciencedirect.com/science/article/pii/S0957417407006628

[48] Ma, Changjie, Guochang Gu, and Jing Zhao. "A Novel Software Reliability Assessment Approach based on Neural Network in Network Environment." *IJACT: International Journal of Advancements in Computing Technology* 4.1 (2012): 136-144, available online: https://pdfs.semanticscholar.org/04a1/7c4f6c0a0d192c4c9c1118c8221f0639323e.pdf

[49] Zheng, Jun. "Cost-sensitive boosting neural networks for software defect prediction." *Expert Systems with Applications*37.6 (2010): 4537-4543, available online: https://www.sciencedirect.com/science/article/pii/S0957417409011026

[50] Shukla, Ruchi, and Arun Kumar Misra. "Estimating software maintenance effort: a neural network approach." *Proceedings of the 1st India software engineering conference*. ACM, 2008. https://dl.acm.org/citation.cfm?id=1342232

[51] Dash, Yajnaseni, Sanjay Kumar Dubey, and Ajay Rana. "Maintainability prediction of object oriented software system by using artificial neural network approach." *International Journal of Soft Computing and Engineering (IJSCE)* 2.2 (2012): 420-423, available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.452.7737&rep=rep1&type=pdf

[52] Al-Jamimi, Hamdi A., and Lahouari Ghouti. "Efficient prediction of software fault proneness modules using support vector machines and probabilistic neural networks." *Software Engineering (MySEC), 2011 5th Malaysian Conference in*. IEEE, 2011. http://ieeexplore.ieee.org/abstract/document/6140679/

[53] Jin, Cong, and Shu-Wei Jin. "Prediction approach of software fault-proneness based on hybrid artificial neural network and quantum particle swarm optimization." *Applied Soft Computing*35 (2015): 717-725, available online: https://www.sciencedirect.com/science/article/pii/S1568494615004366

[54] Kumar, K. Vinay, et al. "Software development cost estimation using wavelet neural networks." *Journal of Systems and Software* 81.11 (2008): 1853-1867, available online: https://www.sciencedirect.com/science/article/pii/S0164121208000071

[55] Gharehchopogh, Farhad Soleimanian. "Neural networks application in software cost estimation: a case study." *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*. IEEE, 2011. http://ieeexplore.ieee.org/abstract/document/5946160/

[56] Rao, B. Tirimula, et al. "A novel neural network approach for software cost estimation using Functional Link Artificial Neural Network (FLANN)." *International Journal of Computer Science and Network Security* 9.6 (2009): 126-131, available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.494.1694&rep=rep1&type=pdf

[57] Attarzadeh, Iman, and Siew Hock Ow. "Proposing a new software cost estimation model based on artificial neural networks." *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*. Vol. 3. IEEE, 2010. http://ieeexplore.ieee.org/abstract/document/5485840/

[58] Yong, Hu, et al. "A neural networks approach for software risk analysis." *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE, 2006. http://ieeexplore.ieee.org/abstract/document/4063720/

[59] Fenton, N., Martin Neil, and D. Marquez. "Using Bayesian networks to predict software defects and reliability." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 222.4 (2008): 701-712. http://journals.sagepub.com/doi/abs/10.1243/1748006XJRR161

[60] Doguc, Ozge, and Jose Emmanuel Ramirez-Marquez. "A generic method for estimating system reliability using Bayesian networks." *Reliability Engineering & System Safety*94.2 (2009): 542-550, available online: https://www.sciencedirect.com/science/article/pii/S0951832008001804

[61] Okutan, Ahmet, and Olcay Taner Yıldız. "Software defect prediction using Bayesian networks." *Empirical Software Engineering* 19.1 (2014): 154-181, available online: https://link.springer.com/article/10.1007/s10664-012-9218-8

[62] Khomh, Foutse, et al. "A bayesian approach for the detection of code and design smells." *Quality Software, 2009. QSIC'09. 9th International Conference on*. IEEE, 2009. http://ieeexplore.ieee.org/abstract/document/5381430/

[63] de Melo, Ana CV, and Adilson J. Sanchez. "Software maintenance project delays prediction using Bayesian Networks." *Expert Systems with Applications* 34.2 (2008): 908-91, available online: https://www.sciencedirect.com/science/article/pii/S0957417406003526

[64] Weber, Philippe, et al. "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas." *Engineering Applications of Artificial Intelligence* 25.4 (2012): 671-682, available online:

https://www.sciencedirect.com/science/article/pii/S095219761000117X

[65] Catal, Cagatay, and Banu Diri. "Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem." *Information Sciences* 179.8 (2009): 1040-1058, available online: https://www.sciencedirect.com/science/article/pii/S0020025508005173

[66] Dejaeger, Karel, Thomas Verbraken, and Bart Baesens. "Toward comprehensible software fault prediction models using bayesian network classifiers." *IEEE Transactions on Software Engineering* 39.2 (2013): 237-257. http://ieeexplore.ieee.org/abstract/document/6175912/

[67] Khodakarami, Vahid, and Abdollah Abdi. "Project cost risk analysis: A Bayesian networks approach for modeling dependencies between cost items." *International Journal of Project Management* 32.7 (2014): 1233-1245, available online: https://www.sciencedirect.com/science/article/pii/S0263786314000027

[68] Lins, Isis Didier, et al. "A particle swarm-optimized support vector machine for reliability prediction." *Quality and Reliability Engineering International* 28.2 (2012): 141-158. http://onlinelibrary.wiley.com/doi/10.1002/qre.1221/full

[69] Can, He, et al. "A new model for software defect prediction using particle swarm optimization and support vector machine." *Control and Decision Conference (CCDC), 2013 25th Chinese*. IEEE, 2013. http://ieeexplore.ieee.org/abstract/document/6561670/

[70] Wahono, Romi Satria, and Nanna Suryana. "Combining particle swarm optimization based feature selection and bagging technique for software defect prediction." *International Journal of Software Engineering and Its Applications* 7.5 (2013): 153-166, available online: http://romisatriawahono.net/lecture/rm/paper/Wahono%20-%20pso%20bagging%20for%20sdp%20-%20%202013.pdf

[71] Cao, Ping, and FuJi Chen. "A risk control optimization model for software project." *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*. IEEE, 2009. http://ieeexplore.ieee.org/abstract/document/5362886/

[72] Azar, Danielle, and Joseph Vybihal. "An ant colony optimization algorithm to improve software quality prediction models: Case of class stability." *Information and Software Technology* 53.4 (2011): 388-393, available online: https://www.sciencedirect.com/science/article/pii/S0950584910002144

[73] Xiao, Jing, Xian-Ting Ao, and Yong Tang. "Solving software project scheduling problems with ant colony optimization." *Computers & Operations Research* 40.1 (2013): 33-46, available online: https://www.sciencedirect.com/science/article/pii/S0305054812001104

[74] Sun, Peng, and Xiaoping Wang. "Application of ant colony optimization in preventive software maintenance policy." *Information Science and Technology (ICIST), 2012 International Conference on*. IEEE, 2012. http://ieeexplore.ieee.org/abstract/document/6221624/

[75] Maleki, Isa, Ali Ghaffari, and Mohammad Masdari. "A new approach for software cost estimation with hybrid genetic algorithm and ant colony optimization." *International Journal of Innovation and Applied Studies* 5.1 (2014): 72, available online: http://search.proquest.com/openview/7b475b6eca4598c05982badd77802a59/1?pq-origsite=gscholar&cbl=2031961

[76] Martens, Anne, et al. "Automatically improve software architecture models for performance, reliability, and cost using evolutionary algorithms." *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*. ACM, 2010. https://dl.acm.org/citation.cfm?id=1712624

[77] Tavakkoli-Moghaddam, Reza, Jalal Safari, and Farrokh Sassani. "Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm." *Reliability Engineering & System Safety* 93.4 (2008): 550-556, available online: https://www.sciencedirect.com/science/article/pii/S0951832007000579

[78] Aljahdali, Sultan H., and Mohammed E. El-Telbany. "Genetic algorithms for optimizing ensemble of models in software reliability prediction." *International Journal on Artificial Intelligence and Machine Learning (AIML) ICGST* 8.1 (2008): 5-13, available online: http://www.academia.edu/download/39928454/AIML-Volume8-issue1-P1121546431.pdf#page=9

[79] Di Martino, Sergio, et al. "A genetic algorithm to configure support vector machines for predicting fault-prone components." *International Conference on Product Focused Software Process Improvement*. Springer, Berlin, Heidelberg, 2011, available online: https://link.springer.com/10.1007%2F978-3-642-21843-9_20

[80] Reyes, Francisco, et al. "The optimization of success probability for software projects using genetic algorithms." *Journal of Systems and Software* 84.5 (2011): 775-785, available online: https://www.sciencedirect.com/science/article/pii/S0164121210003456

[81] Maleki, Isa, Laya Ebrahimi, and Farhad Soleimanian Gharehchopogh. "A hybrid approach of firefly and genetic algorithms in software cost estimation." *Magnt Research Report* 2.6 (2014): 372-388, available online: http://brisjast.com/wp-content/uploads/2015/06/Nov-50-2014.pdf

[82] Ghatasheh, Nazeeh, et al. "Optimizing software effort estimation models using firefly algorithm." *Journal of Software Engineering and Applications* 8.03 (2015): 133, available online: https://www.researchgate.net/profile/Nazeeh_Ghatasheh/publication/273136303_Optimizing_Software_Effort_Estimation_Models_Using_Firefly_Algorithm/links/55095cd30cf26ff55f855453.pdf

[83] Arora, Ishani, and Anju Saha. "Software fault prediction using firefly algorithm." *International Journal of Intelligent Engineering Informatics* 6.3-4 (2018): 356-377, available online https://www.inderscienceonline.com/doi/abs/10.1504/IJIEI.2018.091870

[84] Magro, Micaela Caserza, and Paolo Pinceti. "A confirmation technique for predictive maintenance using the Rough Set Theory." *Computers & Industrial Engineering* 56.4 (2009): 1319-1327, available online https://www.sciencedirect.com/science/article/pii/S0360835208001678