

Review Article

Soft Computing and Decision Support System for Software Process Improvement: A Systematic Literature Review

He Xiaolong,¹ Zhao Huiqi ,² Zhong Lunchao ,³ Shah Nazir ,⁴ Deng Jun,⁵
and Adnan Shahid Khan⁶

¹Robotics Engineering Laboratory for Sichuan Equipment Manufacturing Industry, Sichuan Engineering Technical College, Deyang 618000, China

²College of Intelligent Equipment, Shandong University of Science and Technology, Qingdao, China

³Shenzhen International Graduate School, Tsinghua University, Shenzhen 518000, China

⁴Department of Computer Science, University of Swabi, Swabi, Pakistan

⁵Sichuan Staff University of Science and Technology, Chengdu 610101, China

⁶Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia

Correspondence should be addressed to Zhao Huiqi; zhqskd@163.com and Zhong Lunchao; qiangshou302@mail.tsinghua.edu.cn

Received 14 April 2021; Revised 9 May 2021; Accepted 27 May 2021; Published 8 June 2021

Academic Editor: Zhongguo Yang

Copyright © 2021 He Xiaolong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software project development is very crucial, and measuring the exact cost and effort of development is becoming tedious and challenging. Organizations are trying to wind up their project of software development within the agreed budget and schedule successfully. Traditional practices are inadequate to achieve the current needs of the software industry. Underestimation and overestimation of software development effort lead to financial implications in the form of resources, cost of staffing, and budget of developing the software project. Soft computing (SC) approaches and tools deliver an addition of techniques for anticipating resistance to the deception, defect, incomplete truth for traceability and ambiguity, low arrangement cost, and strength. A large amount of SC approaches is prevailing in the literature to accomplish way-out to difficulties precisely, practically, and speedily. The approaches of SC can give better prediction, high performance, and dynamic behavior. SC deals with computational intelligence which integrates the concept of agent paradigm and SC. The proposed study presents a systematic literature review (SLR) of the approaches, tools, and techniques of SC used in the literature. The study presented a comprehensive review by searching the defined keywords in the popular libraries, filtered the paper, and obtained most relevant papers. After the selection of the papers, the quality assessment process of the included papers has been done in order to determine the relevancy of the papers. The study will help researchers in the area of research to devise novel ideas and solutions to overcome the existing issue on the basis of this study as evidence of the literature.

1. Introduction

Soft computing approaches provide an accumulation of procedures for intending resistance to the deception, defect, incomplete truth for traceability and ambiguity, low arrangement cost, and strength. Various approaches exist for the fault prediction of the software process. The applications of SC such as artificial neural network (ANN), Mamdani and Takagi–Sugeno style fuzzy inference system (FIS), and adaptive neuro-FIS are used to build the fault prediction

model. A framework was presented for making process and product for predicting agile software development faulty modules. Using the Ant dataset project version of the PROMISE data repository, various experiments were performed for obtaining the goal of the research. Furthermore, for assessing the outcomes of the suggested predictive model, the receiver operating characteristic with area under the curve was applied. The proposed research can help the developer in the design phase as well in testing and maintenance phases. The approach can also lessen the time effort

in the review process and code [1]. The research elaborated the issues residing in the development of integrated intelligent information systems, wherein information is observed as a function of idea and which has open-ended query controlling features. SC with specific reference to rough sets as the probable candidate has been analyzed. Large-scale computing shift to large-scope computing in the multilingual databases and multimedia context has been highlighted [2].

The developmental phase of the software project is very crucial, and measuring the exact cost and effort of development is becoming tedious and challenging. Organizations are trying to wind up their project of software development with the agreed budget and schedule successfully. Traditional techniques are inadequate to accomplish the current needs. Underestimation and overestimation of the effort of software development lead to financial implications in the form of resources, cost of staffing, and budget of developing the software project. The needs of an appropriate and well-defined SC model can be integrated for estimating precisely and predicting the cost or effort in time. The proposed study has considered approaches of SC such as support vector regression, linear regression, ensemble learning, multilayer perceptron, and random forest which are applied and compared for assessing the precise effort estimation, and several error assessment metrics are calculated and documented. Chinese datasets have been considered which consist of 499 standard project instances. During the training and testing of the dataset, it was achieved that mean absolute error, medium of MRE, magnitude of the error related to the estimate, PRED (25), and magnitude of the relative error for the decision tree and random forest showed improved results than the available techniques [3]. The phases of software engineering such as project planning, monitoring and controlling, and scheduling are based on the precise estimation of effort and cost. At the early stages of the software development life cycle, it is uncertain to accurately and precisely measure the effort of software which can eventually lead software to failure. The study has presented experimental comparisons of cost estimation models of software based on the approaches applied in the model through statistical criteria. The results were good compared to the existing software cost estimation approaches to the defined parameters in whole and also dependent on the project type, apaches, and data used in implementation [4].

At present, no detailed study has been reported in the area of research. The contribution of the proposed study is to present an SLR of the approaches, tools, and techniques of SC. The study presented a comprehensive review by searching the defined keywords in the popular libraries, filtered the paper, and obtained most relevant papers. After the selection of the papers, the quality assessment process of the included papers has been done in order to determine the relevancy of the papers.

The organization of the paper is as follows: Section 2 is the representation of the research method and protocol followed for conducting the proposed study. This section consists of other sections which further elaborate the research method in detail. Section 3 shows the results and

discussion section of the paper. The paper is concluded in Section 4.

2. Research Method

2.1. Process of the Research Plan. The protocol of the SLR is a way of facilitating the process of keywords' search, identifying associated materials related to the area of research, bringing together studies in a well-organized manner, and originating significant information and derivations from the selected studies and materials. Figure 1 signifies the process followed for the planned study, where initially, the protocol of the review process is defined, then the search strategies are followed by documentation of search strategies, then appropriate materials are encompassed while the rest of materials are excluded, the process of quality evaluation is done for the included papers, and lastly, the data analysis and extraction from the included papers are performed.

Figure 2 graphically represents the SLR process for conducting the proposed study. The figure consists of three research questions, four selected libraries, search string of keywords, initial number of obtained research publications, filtering process by title, abstract, and then contents, the quality assessment of the included papers, and data extraction and synthesis from the studies.

The total process of papers' filtering is shown in Figure 3.

2.2. Research Questions. The following research questions were defined for the proposed study:

- (i) What are the existing approaches and techniques of SC generally used in software development?
- (ii) What techniques and approaches are used in which situation of software development?
- (iii) How to analyze the existing research in the area of SC in software development?

2.3. Keywords and Libraries. Popular libraries were considered for the search process. Other libraries such as Google Scholar were skipped due to the reason that they are publishing all the materials, while the selected libraries are only publishing the materials which are peer reviewed. Following is the list of libraries considered for the proposed study:

- (i) ACM
- (ii) ScienceDirect
- (iii) IEEE
- (iv) Springer

The following keywords were searched in the given libraries. These keywords were decided as the most relevant to the proposed study. ("Soft computing") AND ("decision support system") OR ("DSS") AND ("software process") AND ("improvement").

2.4. Assessing the Quality of the Included Papers. The quality assessment of the included papers was considered as a significant part of the proposed review for checking the

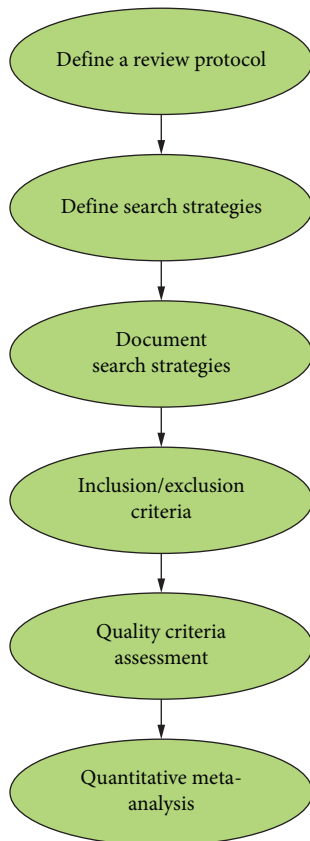


FIGURE 1: Generic process of the SLR.

quality and relevancy of the included studies. Various measures were used for ranking the relevancy of the selected papers related to the proposed study. Score 1 was given for the paper which is more relevant to the defined research question, score 0.5 was given to the paper which is somewhat relevant to the proposed study, and score 0 was given to the paper which is not relevant to the defined research question.

3. Results and Discussion

The following sections describe the results and discussion of the proposed study.

3.1. What Are the Existing Approaches and Techniques of SC Generally Used in Software Development? Soft computing idea came into existence with the replacement of traditional computing, dealing with approximation and giving solutions for real-life problems. Distinct from hard computing, soft computing accepts the issues of uncertainty, approximation, imprecision, and partial truth. The approaches of soft computing mostly consist of genetic algorithms, fuzzy logic, machine learning, artificial neural networks, and expert systems. Various applications of soft computing exist in different domains of real life such as automatic control engineering and many others. The approaches of soft computing are currently being used in many commercial, industrial, and domestic applications. Figure 4 represents

the conventional and soft computing-based problem-solving approach [5].

Figure 5 represents the included papers in the study along with the details of the methods used, the type of publication, and its year of publishing [1, 4, 6–45].

3.2. What Techniques and Approaches of SC Are Used in Which Situation of Software Development? Various approaches of soft computing are used in different situations of software development. Sharma and Chandra [47] presented comparisons of different approaches of SC concerning the fault prediction of the process of software. Apart from this, software fault prediction process pros and cons for approaches of SC are presented. Results of the study have shown that approaches of SC have the tendency for identifying faults in the software development process. The applications of neural networks (NNs) and approaches of SC for the estimation of effort through fuzzy logic, random forest, adaptive neuro-FIS, and support vector machine are presented. Models of SC are established through the NN, fuzzy logic, and ANFIS through the Desharnais and NASA93 dataset. These models are compared according to various criteria of assessment. Among these models, the ANFIS achieved improved effort estimates [39]. The study has proposed a default algorithmic model as a substitute when the efficient model exists. The approach has the feature of inherently independent of the algorithmic model choice. With the combination of neural networks, algorithmic models, and fuzzy logic, the approach has integration ability, learning capability, robustness, and good interpretability to uncertain and vague inputs. The results have shown that the proposed approach is effective using projects of the industry for validation [48]. The study has concentrated on the evaluation of normally used approaches of SC such as NN, genetic algorithm, SVM, particle swarm optimization, fuzzy logic, ant bee colony, and ant colony optimization. The study focus based on working of these scientific computing approaches and evaluation of scientific computing approaches for predicting reliability. Various parameters used to estimate and predict reliability are elaborated. Various fields such as software engineering, computer engineering, and mechanical engineering and medical system are considered for estimating and predicting the reliability of different instruments. These ideas can be useful for both hardware and software in terms of predicting reliability through CBSE [28]. The study provides optimization approach solutions for different and complex issues in various software engineering fields. The optimization algorithms consist of mathematical procedures intending for best ideal results to the fault, defect, strength, low arrangement cost, and failure to achieve tractability. An inclusive overview of software metrics and testing based on optimization approaches and SC is introduced. The issues such as software fault prediction and defect prediction and their solutions through optimization algorithms and SC are explained. An overview of the applications of SC approaches and optimization algorithms is presented [19].

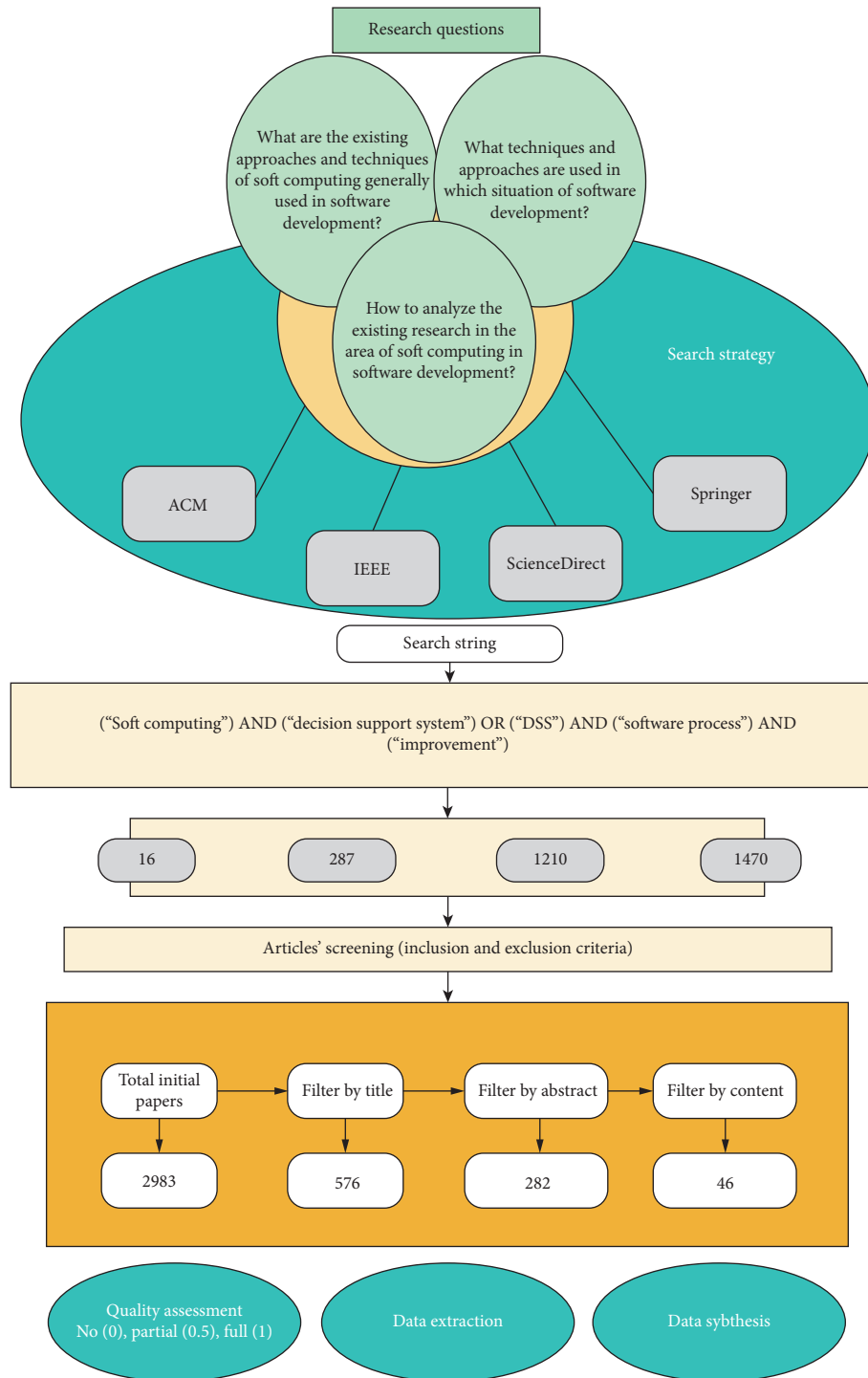


FIGURE 2: Process of conducting the systematic literature review.

A large amount of SC techniques exists in the literature to accomplish way-out to problems accurately, reasonably, and speedily. The approaches of SC can give improved prediction, high performance, and dynamic behavior. The study has presented a comprehensive survey on the available delicate processing approaches, and after that, it analyzed the work in the field of software reliability [18]. Estimation of software effort can play a significant role in the software

project success. Inconsistent, unreliable, and inaccurate estimation of software can eventually lead software to failure. The estimation of software effort in developing software is becoming a crucial task. The effort of software must be measured efficiently for avoiding unexpected results. SC is a link with the approaches involving fuzzy logic, evolutionary computation, and ANN. Such approaches provide and facilitate flexible information-processing ability

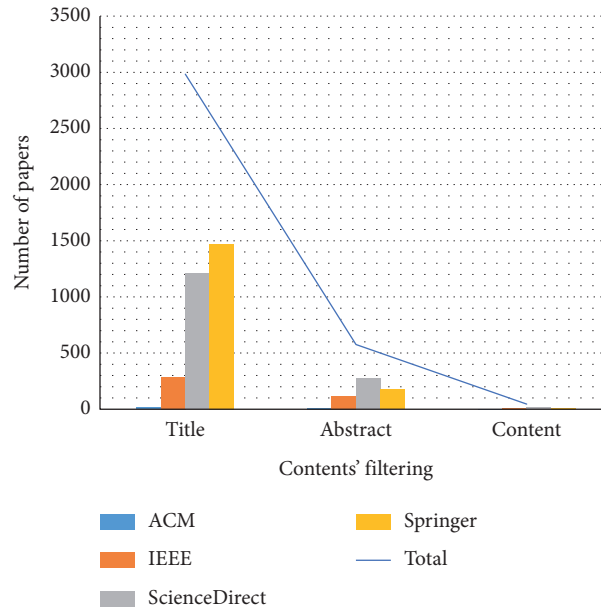


FIGURE 3: Number of papers with the filtering process.

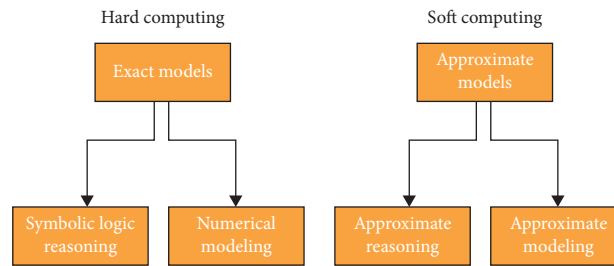


FIGURE 4: Problem-solving approach.

to handle real-life circumstances. The study delivers a detailed report on the effort estimation of software from the early stages including expert judgement-based SEE to the advanced approaches of SC [37]. Sharma and Chandra [23] presented a comprehensive overview on the approaches of SC for testing of software. Various issues of software engineering and specifically software testing and their solutions based on SC were elaborated. The review includes various approaches of SC such as fuzzy logic, NN, particle swarm optimization, ant colony optimization, and genetic algorithm in software testing. Soft agent computing is a branch of computational intelligence which integrates the concept of agent paradigm and SC. This idea delivers technology inheriting the assistances and provides solutions which have some of the features of human intuitive and cognition, abstraction of agent technology, and system level. Research presented an investigation of these limitations from two diverse aspects, architectural and agent characteristics. The research presented diverse layers of abstraction for soft agent computing and then described and proposed the particle soft agent computing system which is based on the model of beliefs-desires-intentions. Lastly, different perspectives were considered during building the system of soft agent computing which are elaborated, and the study presented various research fields for further research [49].

Fuzzy aggregation and fuzzy query were introduced for predicting and ranking the university admission and credit scoring as an alternative, utilizing subjective and imprecise process. Furthermore, the study presented the BISC decision support system. The vital feature of the BISC decision system for the applications of internet is (a) the usage of a huge amount of data intelligently and optimizing the way in organizations as the decision support system and (b) to share securely and intelligently company data within and with the customer and partner of business. Five parts of the model are application template, fuzzy search engine, database, user interface, and evolutionary computing [50]. Chawla et al. [13] proposed a hybrid genetic algorithm and particle swarm optimization-based heuristic for test suite automatic generation. The development and implementation strategy of the proposed system and assessment of the model through experiments with the Java standard library contain ten classes. The algorithm is analyzed statistically with the criterion of test adequacy as branch coverage. The algorithm is compared with the heuristic based on particle swarm optimization, genetic algorithm, and existing hybrid strategies. The efficiency of the research is shown with results of the study. SC approaches can be applied for automating, designing, maximizing, and maintaining the value of the decision process. The advantages are presented in two

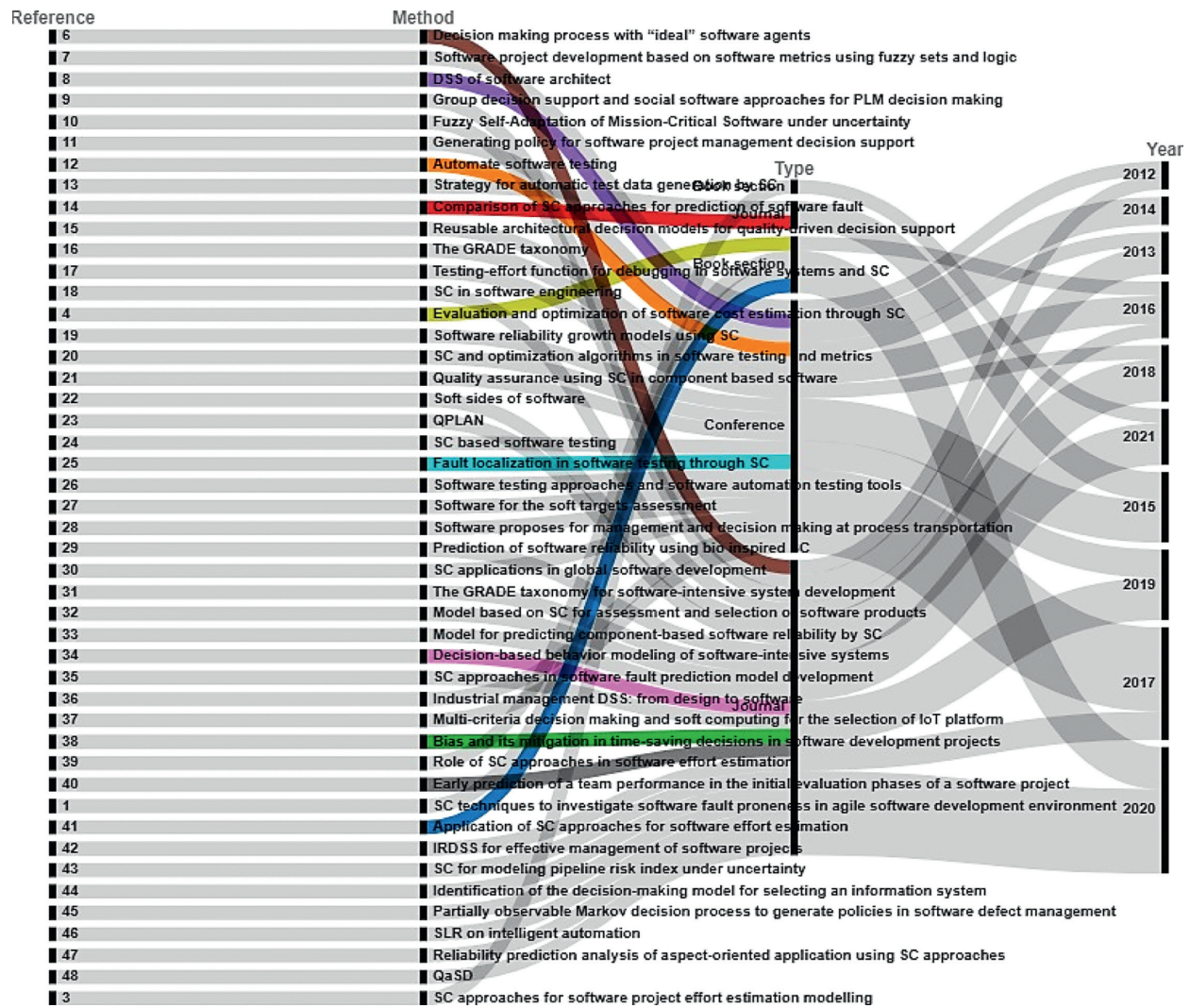


FIGURE 5: Details of the studies and the methods used.

applications; the first one used fuzzy rule-based and case-based reasoning for creating the decision system capable to control risk for assurance applications. The second one is fuzzy information fusion to select replacement parts needed for repairing mechanism of medical imaging [51]. The study has presented a web-based risk avoidance group DSS which facilitates to provide a platform to the expert group for making risk decisions to bidding of the software project. The study has examined the process, principle, function, and frame of the system. The study has presented the prototype system based on the variable precision rough set model [52]. Table 1 summarizes the details of the techniques and methods used in the literature.

3.3. How to Analyze the Existing Research in the Area of SC in Software Development? The purpose of the proposed study was to in-depth analyze the existing literature and to identify the existing materials published in the area. For this purpose,

various popular libraries including ACM, IEEE, ScienceDirect, and Springer were searched. This study has identified the number of publications in the given year, type of publication, venue of publication, and others in the given libraries. Analysis of the publications has been performed from different perspectives which are given in this section. In the first step of the process, the library of ACM was searched based on the defined keywords. The articles' type and the total publications are shown in Figure 6.

In the same library, the sponsors of publications are presented in Figure 7.

The conference events are also identified and are shown in Figure 8.

After completing the search process of the above library, the library of IEEE was searched, and the materials obtained are shown in Figures 9 and 10.

In the same library, the article types and the total of publications are given in Figure 11.

TABLE 1: Techniques and methods used in the literature.

Reference	Method	Type	Year
[13]	Automatic test data generation by SC	Journal	2015
[18]	Software reliability growth models using SC	Conference	2016
[19]	SC and optimization algorithms in software testing and metrics	Conference	2016
[23]	SC-based software testing	Conference	2017
[28]	Prediction of software reliability using bio-inspired SC	Journal	2018
[47]	SC approaches in software fault prediction model development	Journal	2019
[37]	Role of SC approaches in software effort estimation	Book section	2020
[42]	Identification of the decision-making model for selecting an information system	Journal	2020
[48]	Framework for software effort estimation	Journal	2006
[51]	DSS and soft computing	Book section	2003
[52]	Web-based risk avoidance group DSS in software project bidding	Conference	2006

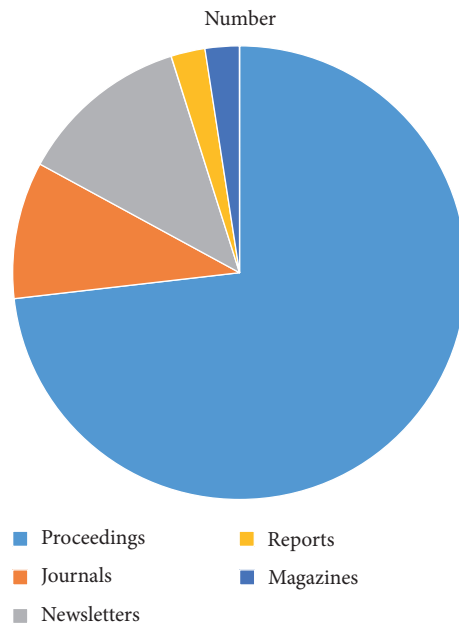


FIGURE 6: Article type.

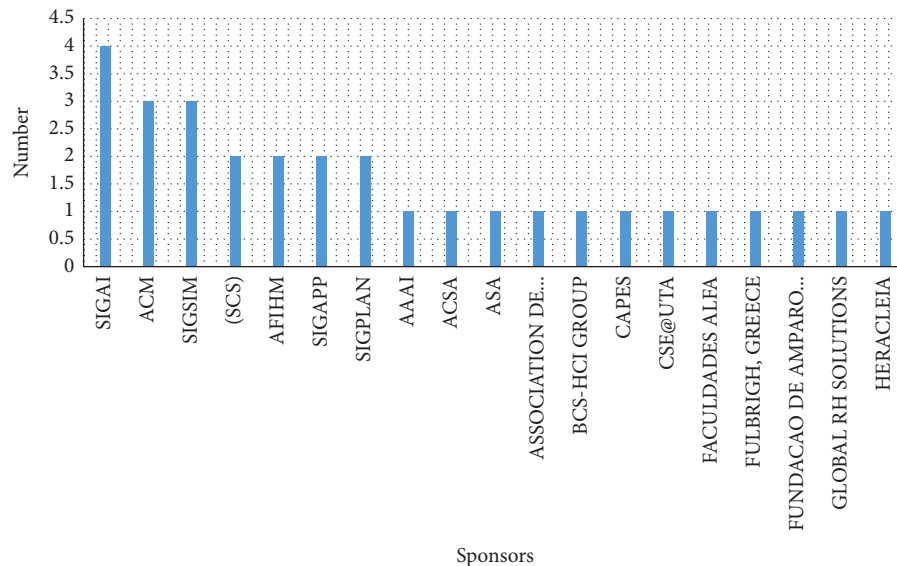


FIGURE 7: Sponsors of publications.



FIGURE 8: Conference events.

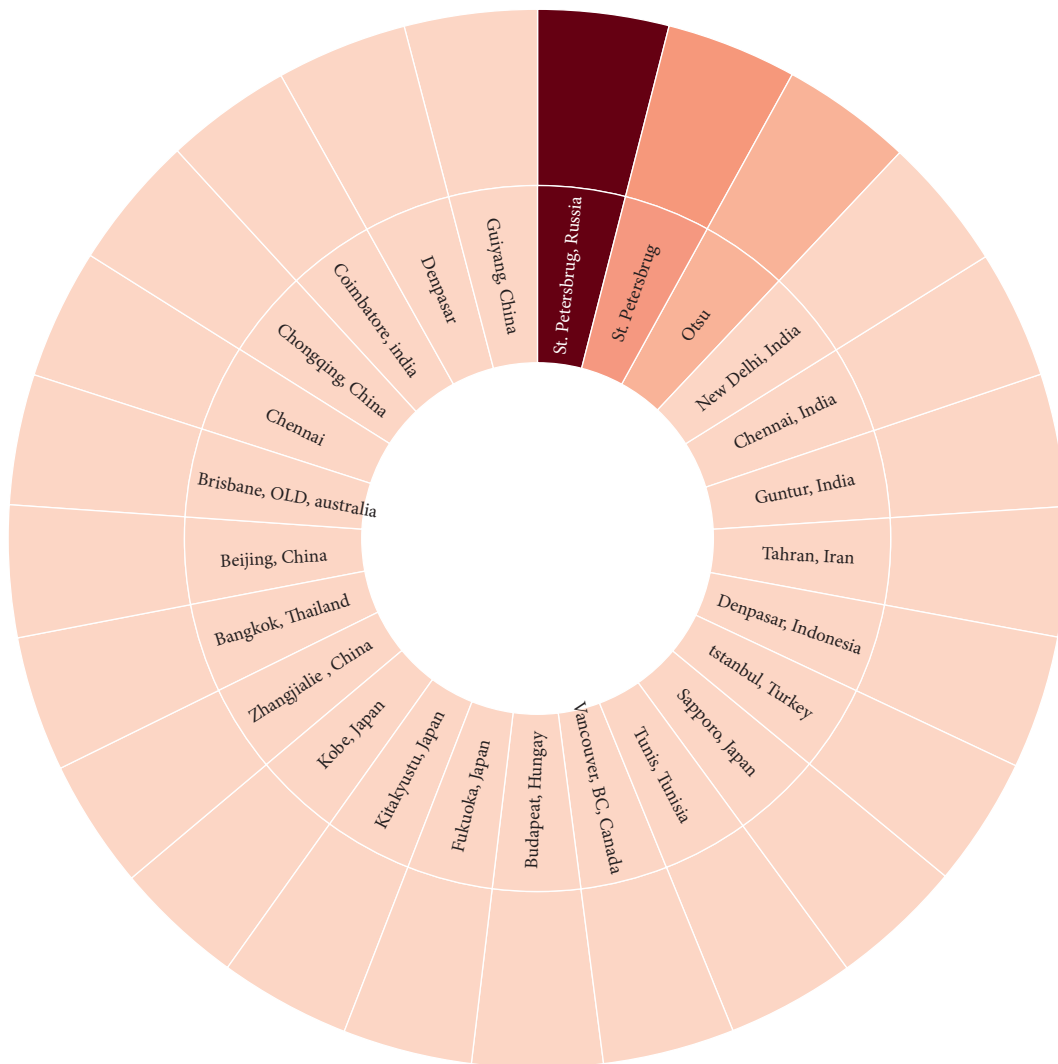


FIGURE 9: Location of conferences.



FIGURE 10: Publication topics.

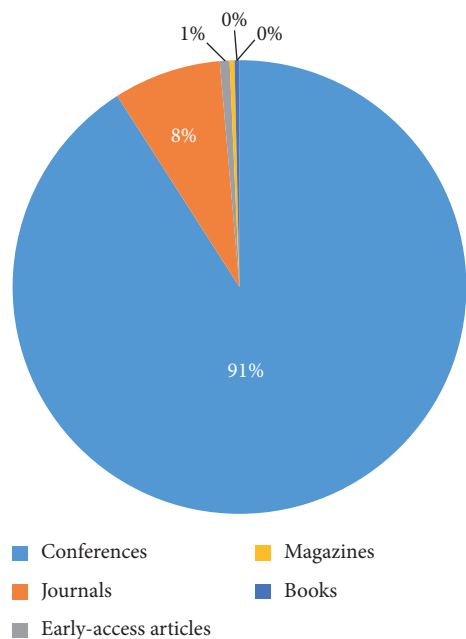


FIGURE 11: Article type.

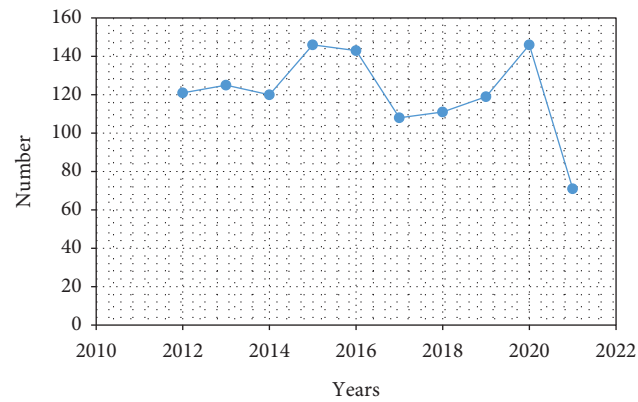


FIGURE 12: Years with the number of publications.

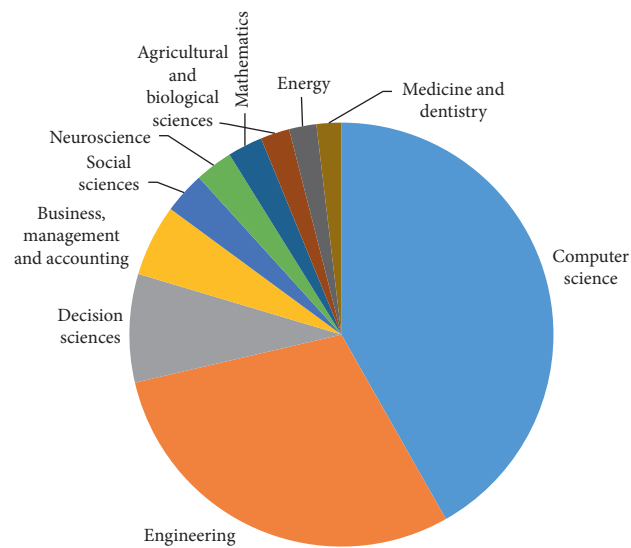


FIGURE 13: Subject and number of papers.

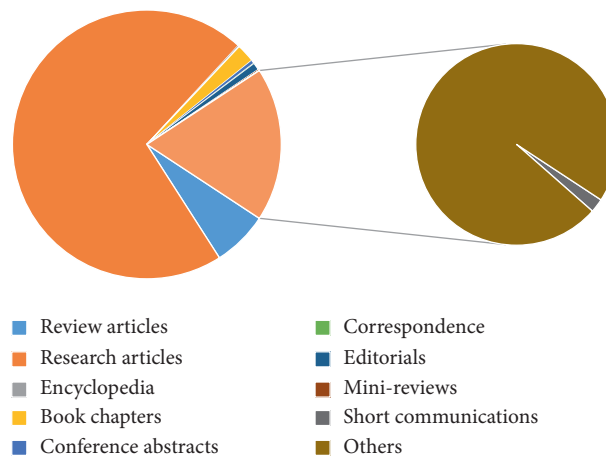


FIGURE 14: Article types.

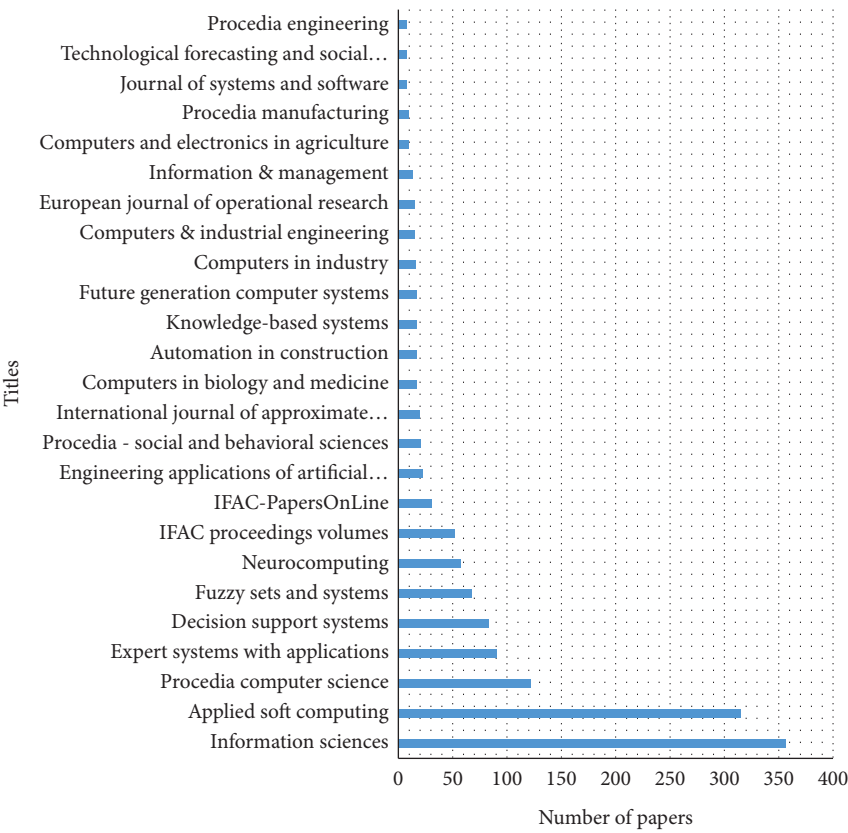


FIGURE 15: Publication titles and papers.

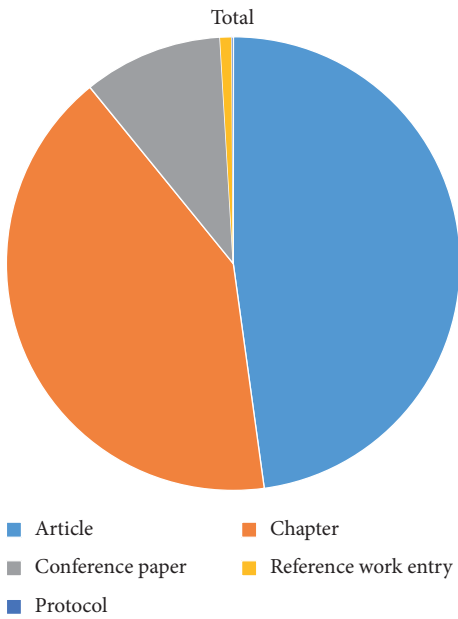


FIGURE 16: Article types.

The library of ScienceDirect was searched for obtaining the associated materials in the area of the study. Figure 12 graphically shows the years with the total of publications. The subject along with the number of publications is given in Figure 13.

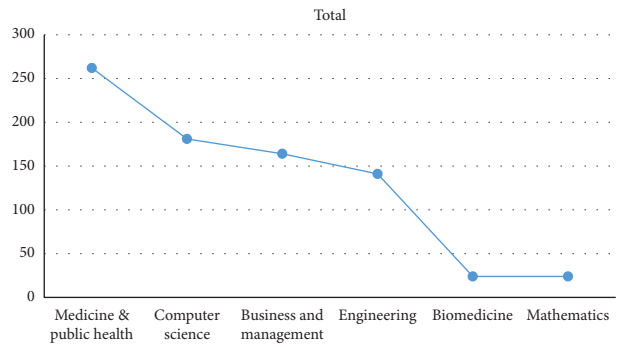


FIGURE 17: Disciplines and papers.

The type of articles is shown in Figure 14. In this library, furthermore, the publication titles and the number of publications are identified and shown in Figure 15. At last, the library of Springer was searched according to the defined search query. In this library, the article types are represented in Figure 16. Figure 17 graphically depicts the disciplines and the number of papers. Subdisciplines were studied in the given library which are shown in Figure 18. The current study has identified some areas of research which need improvement for better software development. These areas of research include software process

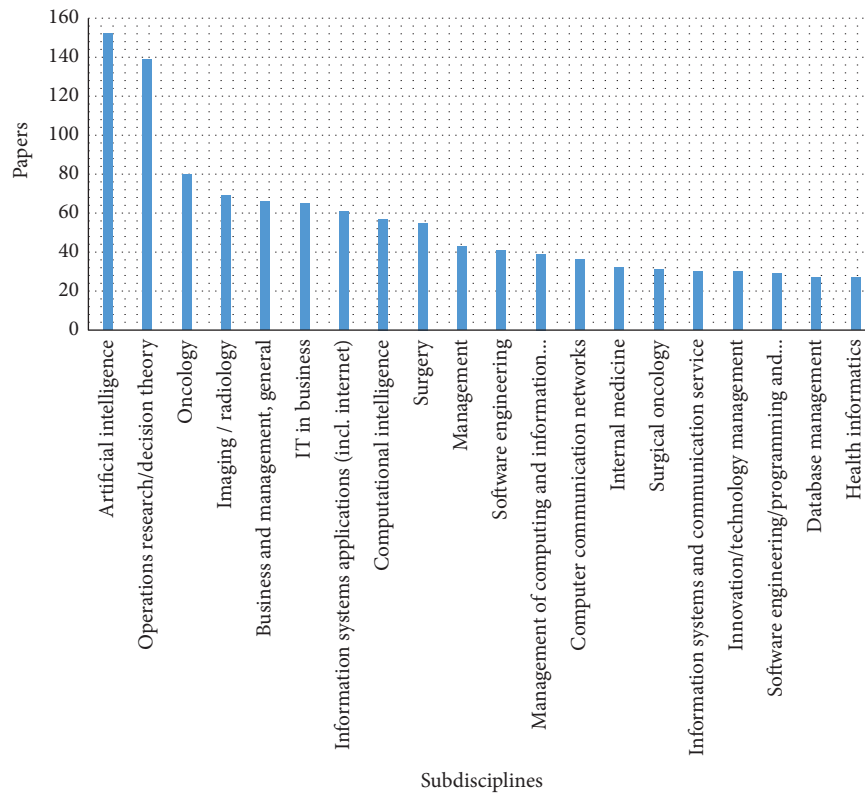


FIGURE 18: Subdisciplines and papers.

improvement, software quality, software engineering, program testing, software maintenance, software metrics, performance evaluation of software, software architecture, and public domain software.

4. Conclusion

Software development is a very fundamental phase of software. Determining the exact cost and effort of development is becoming a tedious and challenging task. Traditional practices are insufficient to attain the present needs of the software industry. Underestimation and overestimation of software development effort lead to financial implications in the form of resources, cost of staffing, and budget of developing the software project. Soft computing approaches deliver an accumulation of measures for intending resistance to the deception, defect, incomplete truth for traceability and ambiguity, low arrangement cost, and strength. A large amount of soft computing approaches is prevailing in the literature for accomplishing way-out to problems precisely, practically, and speedily. The proposed study has presented an SLR of the approaches, tools, and techniques of soft computing used in the literature. The study presented a comprehensive review by searching the defined keywords in the popular libraries, filtered the paper, and obtained most relevant papers. After the selection of the papers, the quality assessment process of the included papers has been done in order to conclude the relevancy of the papers. The study shows that there is rise up in the literature in the proposed research area. This study will provide

support for researchers in the field of research to devise novel ideas and solutions, and this study will be considered as evidence of the literature.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] P. Sharma and A. L. Sangal, "Soft computing approaches to investigate software fault proneness in agile software development environment," in *Applications of Machine Learning*, pp. 217–233, Springer, Berlin, Germany, 2020.
- [2] A. B. Patki, G. V. Raghunathan, S. Ghosh, S. Sivasubramanian, and A. Khurshid, "Soft computing for evolutionary information systems—potentials of rough sets," in *Soft Computing in Industrial Applications*, pp. 481–494, Springer, Berlin, Germany, 2000.
- [3] S. Sharma and S. Vijayvargiya, "Applying soft computing techniques for software project effort estimation modelling," in *Nanoelectronics, Circuits and Communication Systems*, pp. 211–227, Springer, Singapore, 2021.
- [4] G. Kumar and P. K. Bhatia, "Empirical assessment and optimization of software cost estimation using soft computing techniques," in *Advanced Computing and Communication Technologies*, pp. 117–130, Springer, Singapore, 2016.

- [5] D. Ibrahim, "An overview of soft computing," *Procedia Computer Science*, vol. 102, pp. 34–38, 2016.
- [6] Y. Duan, V. K. Ong, M. Xu, and B. Mathews, "Supporting decision making process with "ideal" software agents—what do business executives want?" *Expert Systems with Applications*, vol. 39, no. 5, pp. 5534–5547, 2012.
- [7] S. Mirseidova and L. Atymtayeva, "Definition of software metrics for software project development by using fuzzy sets and logic," in *Proceedings of the 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligence Systems*, pp. 272–276, Kobe, Japan, November 2012.
- [8] A. Harchenko, I. Bodnarchuk, and I. Halay, "Decision support system of software architect," in *Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, pp. 265–269, Berlin, Germany, September 2013.
- [9] R. Pasley and B. MacCarthy, "Group decision support and social software techniques for PLM decision making," *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 1756–1761, 2013.
- [10] Q.-L. Yang, J. Lv, X.-P. Tao, X.-X. Ma, J.-C. Xing, and W. Song, "Fuzzy self-adaptation of mission-critical software under uncertainty," *Journal of Computer Science and Technology*, vol. 28, no. 1, pp. 165–187, 2013.
- [11] A. T. Sadabadi and N. Kama, "Generating policy for software project management decision support paradigm," in *Proceedings of the 2014 IEEE 5th International Conference on Software Engineering and Service Science*, pp. 58–63, Beijing, China, June 2014.
- [12] Z. Sahaf, V. Garousi, D. Pfahl, R. Irving, and Y. Amannejad, "When to automate software testing? decision support based on system dynamics: an industrial case study," in *Proceedings of the 2014 International Conference on Software and System Process*, Nanjing, China, May 2014.
- [13] P. Chawla, I. Chana, and A. Rana, "A novel strategy for automatic test data generation using soft computing technique," *Frontiers of Computer Science*, vol. 9, no. 3, pp. 346–363, 2015.
- [14] E. Erturk and E. A. Sezer, "A comparison of some soft computing methods for software fault prediction," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1872–1879, 2015.
- [15] E. Papatheocharous, K. Petersen, A. Cicchetti, S. Sentilles, S. M. A. Shah, and T. Gorschek, "Decision support for choosing architectural assets in the development of software-intensive systems: the GRADE taxonomy," in *Proceedings of the 2015 European Conference on Software Architecture Workshops, Dubrovnik, Cavtat, Croatia*, 2015.
- [16] M. J. Pasha, S. Ranjitha, and H. N. Suresh, "Testing-effort function for debugging in software systems and soft computing model," in *Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 913–919, Greater Noida, India, October 2015.
- [17] C. Catal, C. Bayrak, A. B. Nassif, K. Polat, and A. Akbulut, "Special issue: soft computing in software engineering," *Applied Soft Computing*, vol. 49, pp. 953–955, 2016.
- [18] S. Lohmor and B. B. Sagar, "A comprehensive review on software reliability growth models utilizing soft computing approaches," in *Proceedings of the International Symposium on Intelligent Systems Technologies and Applications*, pp. 509–523, Springer, Berlin, Germany, September 2016.
- [19] D. Sharma and P. Chandra, "Applicability of soft computing and optimization algorithms in software testing and metrics—a brief review," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition*, pp. 535–546, Springer, Berlin, Germany, 2016.
- [20] O. Bhardwaj and S. Kumar Jha, "Quality assurance through soft computing techniques in component based software," in *Proceedings of the 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, pp. 277–282, Bengaluru, India, August 2017.
- [21] L. F. Capretz, F. Ahmed, and F. Q. B. da Silva, "Soft sides of software," *Information and Software Technology*, vol. 92, pp. 92–94, 2017.
- [22] M. A. A. Féris, O. Zwikael, and S. Gregor, "QPLAN: decision support for evaluating planning quality in software development projects," *Decision Support Systems*, vol. 96, pp. 92–102, 2017.
- [23] D. Sharma and P. Chandra, "Soft computing based software testing—a concise travelogue," in *Proceedings of the Sixth International Conference on Soft Computing for Problem Solving*, pp. 220–228, Springer, Singapore, April 2017.
- [24] P. K. Singh, S. Garg, M. Kaur, M. S. Bajwa, and Y. Kumar, "Fault localization in software testing using soft computing approaches," in *Proceedings of the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*, pp. 627–631, Solan, India, September 2017.
- [25] K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," in *Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 77–81, Chennai, India, August 2017.
- [26] L. Duricova, M. Hromada, and J. Mrazek, "The proposal of the software for the soft targets assessment," in *Proceedings of the 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIIT)*, pp. 90–95, Denpasar, Indonesia, September 2017.
- [27] J. Mrazek, L. Duricova, and M. Hromada, "The software proposes for management and decision making at process transportation," in *Proceedings of the 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIIT)*, pp. 120–123, Denpasar, Indonesia, September 2017.
- [28] C. Diwaker, P. Tomar, R. C. Poonia, and V. Singh, "Prediction of software reliability using bio inspired soft computing techniques," *Journal of Medical Systems*, vol. 42, no. 5, p. 93, 2018.
- [29] A. Iftikhar, S. Musa, M. Alam, M. M. Su'ud, and S. M. Ali, "A survey of soft computing applications in global software development," in *Proceedings of the 2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pp. 1–4, Bangkok, Thailand, May 2018.
- [30] E. Papatheocharous, K. Wnuk, K. Petersen et al., "The GRADE taxonomy for supporting decision-making of asset selection in software-intensive system development," *Information and Software Technology*, vol. 100, pp. 1–17, 2018.
- [31] Y. F. Perez, C. Cruz Corona, and J. L. Verdegay Galdeano, "A new model based on soft computing for evaluation and selection of software products," *IEEE Latin America Transactions*, vol. 16, no. 4, pp. 1186–1192, 2018.
- [32] C. Diwaker, P. Tomar, A. Solanki et al., "A new model for predicting component-based software reliability using soft computing," *IEEE Access*, vol. 7, pp. 147191–147203, 2019.
- [33] J. Kirby, "Decision-based behavior modeling of software-intensive systems," *Procedia Computer Science*, vol. 153, pp. 193–201, 2019.
- [34] A. G. Shmeleva and A. I. Ladynin, "Industrial management decision support system: from design to software," in

- Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIcon-Rus)*, pp. 1474–1477, Saint Petersburg and Moscow, Russia, January 2019.
- [35] Y. Kondratenko, G. Kondratenko, and I. Sidenko, “Multi-criteria decision making and soft computing for the selection of specialized IoT platform,” in *Recent Developments in Data Science and Intelligent Analysis of Information*, O. Chertov, T. Mylovanov, Y. Kondratenko, J. Kacprzyk, V. Kreinovich, and V. Stefanuk, Eds., Springer, Cham, Berlin, Germany, pp. 71–80, 2019.
 - [36] L. Fink and B. Pinchovski, “It is about time: bias and its mitigation in time-saving decisions in software development projects,” *International Journal of Project Management*, vol. 38, no. 2, pp. 99–111, 2020.
 - [37] P. S. Kumar and H. S. Behera, “Role of soft computing techniques in software effort estimation: an analytical study,” in *Computational Intelligence in Pattern Recognition*, pp. 807–831, Springer, Berlin, Germany, 2020.
 - [38] M. Naseer, W. Zhang, and W. Zhu, “Early prediction of a team performance in the initial assessment phases of a software project for sustainable software engineering education,” *Sustainability*, vol. 12, no. 11, p. 4663, 2020.
 - [39] S. R. Sree and C. P. Rao, “A study on application of soft computing techniques for software effort estimation,” in *A Journey Towards Bio-Inspired Techniques in Software Engineering*, pp. 141–165, Springer, Berlin, Germany, 2020.
 - [40] M. Hamid, F. Zeshan, A. Ahmad et al., “An intelligent recommender and decision support system (IRDSS) for effective management of software projects,” *IEEE Access*, vol. 8, pp. 140752–140766, 2020.
 - [41] T. Dawood, E. Elwakil, H. M. Novoa, and J. F. G. Delgado, “Soft computing for modeling pipeline risk index under uncertainty,” *Engineering Failure Analysis*, vol. 117, Article ID 104949, 2020.
 - [42] B. Paradowski and Z. Drajek, “Identification of the decision-making model for selecting an information system,” *Procedia Computer Science*, vol. 176, pp. 3802–3809, 2020.
 - [43] S. Akbarinasaji, C. Kavaklioglu, A. Başar, and A. Neal, “Partially observable Markov decision process to generate policies in software defect management,” *Journal of Systems and Software*, vol. 163, Article ID 110518, 2020.
 - [44] K. K. H. Ng, C.-H. Chen, C. K. M. Lee, J. Jiao, and Z.-X. Yang, “A systematic literature review on intelligent automation: aligning concepts from theory, practice, and future perspectives,” *Advanced Engineering Informatics*, vol. 47, Article ID 101246, 2021.
 - [45] P. Kumar, S. K. Singh, and S. Deo Choudhary, “Reliability prediction analysis of aspect-oriented application using soft computing techniques,” *Materials Today: Proceedings*, 2021.
 - [46] L. López, M. Manzano, C. Gómez et al., “QaSD: a quality-aware strategic dashboard for supporting decision makers in agile software development,” *Science of Computer Programming*, vol. 202, Article ID 102568, 2021.
 - [47] D. Sharma and P. Chandra, “A comparative analysis of soft computing techniques in software fault prediction model development,” *International Journal of Information Technology*, vol. 11, no. 1, pp. 37–46, 2019.
 - [48] X. Huang, D. Ho, J. Ren, and L. F. Capretz, “A soft computing framework for software effort estimation,” *Soft Computing*, vol. 10, no. 2, pp. 170–177, 2006.
 - [49] A. Mamdani, A. G. Sichanie, and J. Pitt, “Soft agent computing: towards enhancing agent technology with soft computing,” in *Intelligent Systems and Soft Computing*, pp. 110–135, Springer, Berlin, Germany, 2000.
 - [50] M. Nikraves and S. Bensafi, “Soft computing for perception-based decision processing and analysis: web-based BISC-DSS,” in *Soft Computing for Information Processing and Analysis*, pp. 93–188, Springer, Berlin, Germany, 2005.
 - [51] W. Cheetham, K. Goebel, and P. Bonissone, “Benefits of decision support using soft computing,” in *Applied Decision Support with Soft Computing*, pp. 61–85, Springer, Berlin, Germany, 2003.
 - [52] G. Xie, J.-l. Zhang, and K. K. Lai, “Web-based risk avoidance group decision support system in software project bidding,” in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Hong Kong, China, December 2006.