International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Survey Paper on Soft Computing Techniques for Software Reliability

Parmanand kushwah[*1] *, Mr Ramnaresh Sharma*[*2]

[*1, 2] Maharana Pratap College of Technology, CSE,Gwalior, MP, India

ABSTRACT

Software Reliability is the prospect of failure-free operation of software for a quantified period in a specified environment. The software turns out to be an essential part of many commercial, industrial, and military operations. With the application of software in many safety-critical systems, led to the fact that software is now an important research area. Software reliability measure is one tool used to evaluate software engineering technologies. Researchers had developed many software reliability parameter models in the past 40 years to assess software reliability, such as those based on stochastic differential equation, non-homogeneous Poisson process, and Bayes process. Software reliability models evaluate the reliability by anticipating issues for the software. Reliability is a true marvel with many related continuous issues. To acquire answers for issues rapidly, precisely and satisfactory, countless computing techniques have been produced, however it is extremely hard to discover which one is the most appropriate and can be utilized all around. There is a need to enhance quality software model according to the type of software as well as the efforts needed to complete risk-less Software Development (SD) in a predictive way.

Keywords: Soft computing, Technique for software reliability, Neural Network, Fuzzy logic

1 Introduction

The system specialists, inventors, and administrators introduced software failures during different segments of software development life cycle. To find out and remove these errors they tested the software systems. The quality of software system in terms of reliability is calculated by the deduction of these errors. It plays a significant role in many critical and dayto-day life applications that has led to the tremendous work being carried out in modeling process. These models successfully have been used for approximation and prediction of errors outstanding in the software.

In the Software engineering, Software reliability is an active research area over the past forty five years and still several challenges and transparent questions still exist. It's a core part in software quality along with functional, usability, maintainability, serviceability performance and documentation. Software reliability is defined as the probability of failure-free operation of the software over a specified period of time in a specified environment.Software Reliability is also an important factor which affects the system reliability. Software reliability is different from hardware reliability because it defines the design perfection, than manufacturing perfection. The high complexity of software is a major contributing factor of Software Reliability problems.To obtained solutions to problems quickly, accurately and acceptably, a large number of soft computing techniques have been developed, but it is very difficult to find out which one is the most suitable and can be used globally. In this paper, we have provided an overview of existing soft computing techniques, and then critically analyzed the work done by the various researchers in the field of software reliability. The probability of failure-free operation of a software system for a specified time in a specifies environment. Further to this, we have also compared soft computing techniques in terms of software reliability modeling capabilities.Software Engineering are playing very important role in software life and there is always a need of high quality software. Software reliability is the most measurable aspect of software quality. Software reliability can be defined as the probability of failure-free software operation for a specified period of time in a specified environment.The software failures are introduced by the system analysts, designers, programmers and managers during different phases of software development life cycle. The probability of failure-free operation of a software system for a specified time in a specified environment. To detect and remove these errors, the software system is tested. The quality of software system in terms of reliability is measured by the removal of these errors. Software reliability modeling plays a significant role in many critical and daily life applications, which has led to the tremendous work being carried out in software reliability modeling. These models successfully have been used for estimation and prediction of the number of errors remaining in the software.Soft computing techniques are the accumulation of various ideas and techniques that plan to conquer the troubles experienced in certifiable issues. It manages the issues that appear to be loose, indeterminate and hard to sort. One may see soft computing as an endeavor to impersonate normal animals: plants, creatures, people, that are soft, adaptable, versatile and smart. In this sense, soft computing is the name of a group of critical thinking techniques that have a similarity with organic thinking and critical thinking.

2 Significance of the Study:

Reliable software must comprise often redundant code to perform the essential checking for exceptional conditions. That diminishes program execution speed and surges the quantity of stock mandatory by the program. Unreliable software is liable to be discarded by users: If a company achieves a reputation for unreliability as of single unreliable product, it is likely to disturb future sales of all of that company's products. Unreliable systems may cause info (data) loss: Data is very difficult to gather and to maintain, sometimes it may be of value more than the computer system on which it is processed. The probability of failure-free operation of a software system for a specified time in a specifies environment. Further to this, we have also compared soft computing techniques in terms of software reliability modeling capabilities.

3 Related works:

Soft computing consists of both natural and artificial ideas. Some soft computing techniques play an essential role in many fields such as in computer science machine learning, artificial intelligence applied in engineering such as the mobile robot, cooling, heating, communication network, inverters, converters, electric power system, power electronics, motion control and aircraft, etc. They are preferred because they have the best mapping mechanism for real time situations. Some of the relevant and recent papers are mentioned here with their critical findings.

Anusha Merugu et al. (2019) we have examined the work done by different researchers, with the undertaking made to incorporate however many references as could reasonably. We explore some soft computing techniques, for example, Neural systems (NN), Fuzzy Logic (FL), Genetic Algorithm (GA), Genetic Programming (GP), Artificial Bee Colony (ABC) and Ant Colony and so forth. We underlined on the job of existing soft computing techniques in software reliability modeling, with the dependence that it would fill in as a source of perspective to both old and new, approaching researchers in this field, to help their understanding of flow patterns and help their future research prospects and bearings.

Malhotra, et al. (2015) proposed Software Reliability is vital piece of software quality and is one among the most inescapable perspective for assessing quality of a software product. Software industry bears different difficulties in growing very solid software. Demonstration of machine learning techniques for software reliability prediction has shown particular and significant outputs. In these ML techniques are proposed for prediction and calculate the system reliability based on performance criteria. Founded on the experiments conducted, it was detected that ANFIS yields improved results in all the cases and thus can be used for expecting software reliability. In this study, they give analysis between the two failures data cumulative failure data and inter failure time's data and find that cumulative failure data give the good results than the inter failure time's data.

Succi et al. (2018) Assessment of software reliability is inevitable in modern software manufacture process. Several works designed at better methods for measurement and calculation of reliability of software products. Tens of methods have been developed and calculated so far. Therefore, very few works concentrate on approaches to compare existing systems with respect to reliability. Despite a tremendous importance to practice, a complete and sound comparison methodology does not exist. A methodology for software reliability comparison extensively applies the GQM approach and software reliability growth methods.

Sahar et al. (2017) recommended an effective and appropriate method to measure the Software reliability growth model (SRGMs) efficiency by using an approach which was based on Genetic Algorithm (GA). This approach was useful for assessing the SGRMs parameters. In GA based method three operators were utilized which were viz. mutation, selection and crossover. To evaluate the performance of suggested technique, experiments were performed in the four real data sets and four traditional models; and also compared the efficiency and concert of this technique to further optimization methods. The experimental outcomes show that GA based approach could determine the best solution more efficiently and frequently than other techniques.

Cai et al. (2015) discussed Fuzzy Logic is derived from fuzzy set theory dealing with reasoning that is appropriate rather than precisely deduced from classical predicate logic. A fuzzy model is a mapping between linguistic terms, attached to variables. Hence the input to and output from a fuzzy model can be either numerical or linguistic. The development of fuzzy software reliability models is better in place of probabilistic software reliability models (PSRMs). It was based on the proof that software reliability is fuzzy in nature. A demonstration of how to develop a fuzzy model to characterize software reliability was also presented.

Gondra et al. (2008) developed a machine learning technique based on artificial neural network (ANN) model. In this approach values of software metrics and number of errors are measured for training the neural network. The neural network is trained using the critical values of each software metrics and a sensitivity analysis is performed for performance measurement. In this method of software defect prediction, machine learning techniques involved researchers due to its performance for imbalanced and uncertainty datasets.

Karunanithi et al. (1991) predicted cumulative number of failures by designing first neural network-based software reliability model. They used feed-forward neural network, recurrent neural network and Elman neural network in their study and used execution time as the input of the network. They found that their models are better prediction models than some other statistical models. They used connectionist models for software reliability prediction. Design the architecture of neural network by Falman's cascade correlation algorithm.

4 Summary of Review Paper findings:

The critical observations after reading several papers on software reliability prediction are:

1. Missing or incomplete data.
2. Large number of variables or unused extra variables.
3. Strong co-linearity (cohesion) between/among the variables.
4. Complex linear/non-linear relationships between model variables.

5. Outliers and small sizes of the data sets used for the model evaluation.
6. There is actually no common agreement on procedures to be followed for software reliability prediction by researchers.
7. Some researchers worked more on quantification of datasets but fell short on giving accurate reliability measures.
8. Having a good testing strategy is crucial for any industry with high software development costs.
9. Requirements and design phases need to be in focus while trying to maintain software reliability.
10. As the datasets are very vague and imprecise at early stages fuzzy techniques may be the only methods which may be able to capture the metric values of these stages of SDLC.

5 Methodology: -

5.1 Neural Network:

Neural networks are beginner's model of the biologic neuron system. It is particularly parallel distributed processing system prepared up of highly interconnected neural computing elements. Neural network has been applied for parameters assessment of the formal model and self-learning process in order to predict the future outcomes. Back-error propagation is one of the most widely used neural network paradigms and has been applied successfully in a broad range of areas.

5.2 Fuzzy Computing:

Fuzzy Logic is derived from fuzzy set theory dealing with reasoning that is appropriate rather than precisely deduced from classical predicate logic. A fuzzy model is a mapping between linguistic terms, attached to variables. Therefore the input to and output from a fuzzy model can be either numerical or linguistic.

5.3 Neuro-fuzzy System:

In neuro-fuzzy system we combine fuzzy logic and neural networks. It can be used for software reliability modeling investigation. Neuro-fuzzy describes a methodology for controlling neural networks by fuzzy logic. Neural network exhibit the ability to learn pattern of static or dynamical system. In the following fuzzy approach, the learning of NN are exploited in two stages: first to learn static response curves of a given system and second, to learn the real time dynamical changes in a software system to serve as a reference model. The neuro-fuzzy control architecture uses two neural network to modify the parameters of an adaptive FLC. The adaptive capability of the fuzzy controller is manifested in a rule generation mechanism and automatic adjustment of scaling factors or shapes of membership function.

5.4 Swarm Intelligence:

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of “intelligent” global behavior, unknown to the individual agents. The swarm intelligence are most usable technology in software reliability.

5.5 Bayesian Network:

Bayesian networks are a type of **Probabilistic Graphical Model** that can be used to build models from data and/or expert opinion. They can be used for a wide range of tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty. These capabilities in terms of the four major analytics disciplines, **Descriptive analytics, Diagnostic analytics, Predictive analytics** and **Prescriptive analytics.**

6 Conclusion:

The study shows that the software reliability prediction models proposed by various researchers had certain shortcomings and failed to fulfill all testing scenarios. Software reliability measurement is not a precise science. Though exasperating, the pursuit of quantifying software reliability has never stopped. Accurate and early prediction of software reliability with a single SRGM is difficult. Many researchers have used neural networks but they are suitable only for large datasets. Software failure prediction is often impossible in real world. However we could observe this through character of work in process. The study shows that the software reliability prediction models proposed by various researchers had certain shortcomings and failed to fulfill all testing scenarios. Software reliability measurement is not a precise science. soft computing techniques, such as: Neural networks (NN), Fuzzy Logic (FL), Genetic Algorithm (GA), Genetic Programming (GP), Artificial Bee Colony (ABC) and Ant Colony etc. We emphasized on the role of existing soft computing techniques in software reliability modeling, with the reliance that it would serve as a reference to both old and new, incoming researchers in this field, to support their understanding of current trends and assist their future research prospects and directions.

REFERENCES

1. B W. Boehm, Software engineering economics, Englewood Cliffs, NJ: Prentice-Hall, 1981, pp. 50-100.
2. Malhotra, R., Jaiswal, A. Software reliability prediction using machine learning techniques. International Journal of System Assurance Engineering and Management, 9(1), 2018, pp. 230-244.
3. Succi, G., Reznik. A., Ivanov. V. Comparing the reliability of software systems: A case study on mobile operating systems. Information Sciences,

423, 2018, pp. 398-411

4. Kapur PK, Garg RB (1990) Cost reliability optimum release policies for a software system with testing effort. Operations Research 27: 109-116.
5. Musa JD (1999) Software Reliability Engineering: More Reliable Software, Faster Development and Testing, McGraw-Hill.
6. Musa JD, Iannino A, Komodo K (1987) Software Reliability: Measurement, Prediction and Application. McGraw-Hill.
7. Vyas, O. P., Verma, S., Pal, N. R., Singh, P. Fuzzy rule-based approach for software fault prediction. IEEE Transactions on Systems, Man, and Cybernetics Systems, 47(5), 2017, pp. 826-837.
8. Sahar, O., Latif, M. A., & Imran, M. Machine learning techniques for the evaluation of efficiency of the software reliability growth models. Gomal University Journal of Research [GUJR], 33(1), 2017.
9. Cai, Q., Zhang, D., Zheng, W., Leung, S. A new fuzzy time series forecasting model combined with ant colony optimization and autoregression. Knowledge -Based Systems, 74, 2015, pp. 61-68.
10. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. A survey of deep neural network architectures and their applications,2017.