International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Impact Factor (2020): 6.733

Soft Computing Techniques for Software Reliability

Parmanand Kushwah[1], Ramnaresh Sharma[2]

[1, 2]Maharana Pratap College of Technology, CSE, Gwalior, MP, India
parmanandkushwah920[at]gmail.com

Abstract: ***Software is a major part of today's technical systems, which is developed in various fields. 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. Reliable software is the primary and significant criterion for all software industries, developers and end users. In this paper, we propose a solution to the problem by using large number of soft computing techniques for proving the software reliability. Further to this, we have likewise analyzed soft computing techniques regarding software reliability modeling abilities.***

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

1. Introduction

Software engineering discusses and cost-effective techniques to software development [1]. Software engineering is a discipline whose aim is the production of quality software, software that is delivered on time, within budget, and that satisfies its requirements [2]. Software Engineering are playing very important role in software life and there is always a need of high quality software. Software Engineering assumes a noteworthy job in software life on the grounds that there is dependably a requirement for the high caliber.

The reliability of a software product essentially denotes its trustworthiness. Software reliability is the most quantifiable part of its quality. 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 [3].

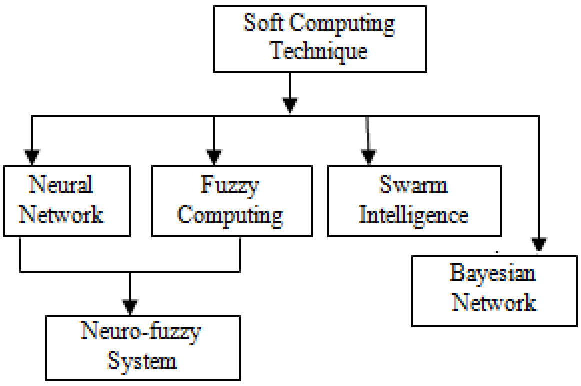
Software reliability modeling plays a significant role in many critical and daily life applications, which has led to the tremendous work being carried out in software reliability modeling. These models successfully have been used for estimation and prediction of the number of errors remaining in the software. User can access the current and future reliability through testing using these models, and can make decisions about the software such as whether the product can be released in its present state or we require further testing in order to improve the quality of software [3]. The software disappointments are presented by the framework investigators, creators, software engineers, and chiefs amid various periods of software improvement life cycle. To recognize and expel these blunders, the software framework is tried. The nature of a software framework as far as reliability is estimated by the expulsion of these mistakes. Software reliability modeling assumes a noteworthy job in numerous basic and day by day life applications that have prompted the colossal work being done in the modeling procedure. These models effectively have been utilized for the estimation and expectation of blunders staying in the software.

Soft computing techniques are the collection of different concepts and techniques which aim to overcome the difficulties encountered in real world problems. 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.

Soft-computing Techniques

In software engineering, soft computing is an association of computing method that includes at its basic member of, neural network, fuzzy logic, wisdom system, neuro fuzzy system, Swarm Intelligence & Bayesian Network. Soft computing is a technique associated with the idea of natural and artificial. Several theory and implementations exists in many areas by this computing. Soft Computing turned into a formal territory of concentrate in Computer Science in the mid-1990s. [2] Earlier computational methodologies could demonstrate and correctly investigate just moderately straightforward frameworks. Progressively mind-boggling frameworks emerging in science, drug, the humanities, the executives' sciences, and comparable fields regularly stayed obstinate to traditional scientific and investigative techniques. Nonetheless, it ought to be called attention to that the multifaceted nature of frameworks is relative and that numerous traditional numerical models have been exceptionally beneficial regardless of their unpredictability. Soft computing manages imprecision, vulnerability, fractional truth, and guess to accomplish process ability, heartiness and low arrangement cost. All things considered, it shapes the premise of a lot of machine learning techniques. Ongoing patterns will, in general, include transformative and swarm knowledge-based calculations and bio-roused computation.

There are a number of soft computing techniques exists and plays an important role in many areas such as, in computer science, machine learning, artificial intelligence applied in engineering areas such as mobile robot, cooling heating, communication network, inverters, converters, electric power system, power electronics, motion control and aircraft etc. . We discussed about the different classification schemes of existing soft computing techniques as shown in Figure-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 [4].

Karunanithi et al. [5] [6] foresee an aggregate number of disappointment by plan first neural system based software reliability demonstrate. They utilized the feed-forward neural system, intermittent neural system and Elman neural system in their examination and use execution time as the contribution of the system. They found that their models are preferable expectation models over some other measurable models [7]. Utilized connectionist models for software reliability expectation. Structure the design of a neural system by Falman's course relationship calculation. They found that for endpoint expectation connectionist approach. Used the neural network for predicting the number of faults and shown static reliability modelling. The trained two neural networks; one with the complete set of principal components and one with the position of components selected by multiple regression model selection. Comparison of these models showed a better accepting of neural network software quality models [8]. Proposed an on-line adaptive software reliability prediction model using evolutionary association approach based on multiple-delayed-input single-output architecture, which showed better performance with esteem to next-step predictability compared to existing NN model [9]. Proposed a model that uses the neural network approach to build a dynamic weighted combinational model. They evaluated the performances of the neural network models with some conventional SRGMs from three aspects: goodness of fit, forecast ability for short-term prediction and long-term prediction. Result shows that purposed model has more accurateness with both goodness of fit and the prediction ability compared to existing conventional models [10]. Proposed two models such as neural network based exponential training and neural network based logarithmic encoding for prediction of cumulative numeral of failures in software. He used execution time as the input and applied on four data sets. Result showed that its result is better that other statistical model.

Exhibited a neural system based strategy for software reliability forecasts, utilized the backpropagation calculation for preparing. Execution of this purposed methodology is assessed by utilizing an alternate number of information hubs and concealed hubs. The outcome demonstrated that its execution relies on the idea of the handled informational collections [11]. Examined an altered Elman intermittent neural system in modeling and foreseeing software disappointments and then played out a complete investigation of connectionist models and their relevance to software reliability forecast and observed them to be preferable and increasingly adaptable over the conventional models [12]. Proposed a method to assess the reliability of the software consisting of system by using different neural network architectures, then estimate the faults prediction behavior in the set of components over a cumulative execution time interval moreover this the prediction of faults is estimated for the complete software [13].

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. Fuzzy Logic approach has been successfully used to solve selections of problems in modeling and recognition of nonlinear systems developed by Prof. Lotfi A. Zadeh [14], Fuzzy Inference System is based on the fuzzy set theory, fuzzy if then rules, and fuzzy reasoning and is a popular computing framework. The basic idea is to model a method which can develop its own software reliability estimation model based on the history of earlier period failures of software. Thus such a model shall eliminate the need of any outside variables or any other dependencies to estimate software reliability [15]. Developed a software reliability prediction model that implemented using the fuzzy logic technique. This model focused on a particular dataset behavior in predicting reliability. Focusing on a particular dataset behavior is performed to develop an accurate model since the recent work focused on developing a model which can be more accurate [16]. Proposed an approach leading to a multi technique knowledge extraction and development of a comprehensive meta-model prediction system in the area of corrective maintenance of software. The system was based on evidence theory and a number of fuzzy-based models. Purposed model consists of a collection of linear sub-models, based on the Takagi-Sugeno technique and attached efficiently using fuzzy membership functions to represent the expected software faults as a function of historical measured faults. This purposed model gives a high performance modeling capabilities. Two fuzzy time series based software reliability models have been proposed. The first one predicts the time between failures of software and the second one predicts the number of errors present in software. The purposed models are flexible, assumption free and very simple in computation. It not required any de-fuzzication techniques separately, which results in a significant reduction of computation time [17].

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 [18].

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.

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.**

2. Comparison of Different Soft Computing Techniques in Terms of Software Reliability Models

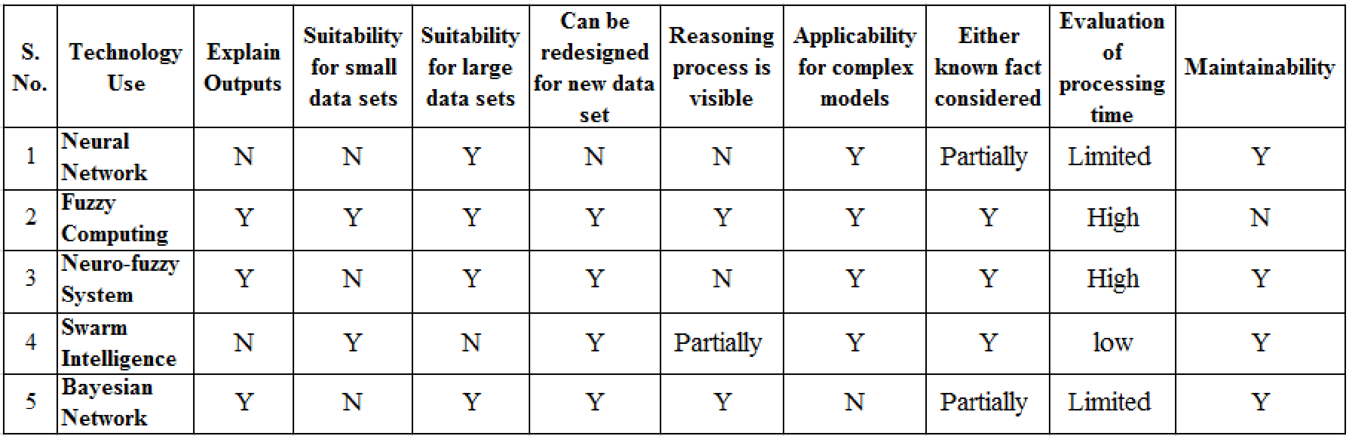
Correlations are extremely helpful if there should arise an occurrence of ideal determination, the client can see every conceivable decision on a solitary stage and select the most appropriate as his/her necessities. In Table 1, we have compared different soft computing techniques in terms of software **reliability modeling capabilities such as data sets, re-adjustments for new data set, process visibility, facts and outputs etc**. This comparison has outlined some parameters of modeling capabilities. This correlation has sketched out a few parameters of modeling abilities. From this table, we saw that every one of the techniques clarify its yields and are appropriate for complex models with the exception of genetic calculation. The correlation likewise uncovered that just fluffy Logic can be generally utilized for all the modeling capacities.

From this table we observed that all the techniques explain its outputs and are applicable for complex models except genetic algorithm. Comparison revealed that only fuzzy Logic can be widely used for all the modeling capabilities. Swarm Intelligence & Bayesian Network can also be used for most of the modeling capabilities except only small data set capabilities. This table will help computer scientist who are keen to contribute their works to the field of software reliability show in Table 1. 1.

3. Conclusion

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), Swarm Intelligence & Bayesian Network 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.

Table 1.1



S. No.	Technology Use	Explain Outputs	Suitability for small data sets	Suitability for large data sets	Can be redesigned for new data set	Reasoning process is visible	Applicability for complex models	Either known fact considered	Evaluation of processing time	Maintainability
1	Neural Network	N	N	Y	N	N	Y	Partially	Limited	Y
2	Fuzzy Computing	Y	Y	Y	Y	Y	Y	Y	High	N
3	Neuro-fuzzy System	Y	N	Y	Y	N	Y	Y	High	Y
4	Swarm Intelligence	N	Y	N	Y	Partially	Y	Y	low	Y
5	Bayesian Network	Y	N	Y	Y	Y	N	Partially	Limited	Y

References

[1] Rajib mall (2009) Fundamental of Software Engineering, 3rd edition, PHI learning private limited, New Delhi.

[2] http://ijrpublisher. com/gallery/139-january-2019. pdf (Base paper Published January-2019)

[3] Agarwal kk, Singh Y (2005) Software Engineering. New Age International Publisher, New Delhi.

[4] https://www. longdom. org/open-access/software-reliability-modeling-using-soft-computing-techniques-criticalreview-2165-7866-1000144. pdf

[5] Khoshgoftaar TM, Szabo RM, Guasti PJ Exploring the Behaviour of Neural-network Software Quality Models. Software Engg J (1995) 10: 89-96.

[6] N. Karunanithi , Y. K. Malaiya , D. Whitley, "Prediction of software reliability using neural networks",Proceedings of the Second IEEE International Symposium on Software Reliability Engineering, pp. 124– 130, 1991.

[7] N. Karunanithi, D. Whitley, Y. K. Malaiya, "Using neural networks in reliability prediction", IEEE Software, Vol. 9, no. 4, pp. 53–59, 1992.

[8] N. Karunanithi, D. Whitley, Y. K. Malaiya, "Prediction of software reliability using connectionist models", IEEE Trans Software Engg. , Vol. 18, No. 7, pp. 563-573, 1992.

[9] Ho SL, Xie M, Goh TN A Study of the Connectionist Models for Software Reliability prediction. Computers and Mathematics with Applications (2003) 46: 1037-1045.

[10] Su SY, Huang CY, Chen YS, Chen JX An Artificial Neural-Network- Based Approach to Software Reliability Assessment. TENCON, IEEE Region (2005) 10:1-6.

[11] Viswanath SPK Software Reliability Prediction using Neural Networks. PhD. Thesis, Indian Institute of Technology Kharagpur (2006).

[12] K. Y. Cai, L. Cai , W. D. Wang , Z. Y. Yu , D. Zhang, "On the neural network approach in software reliability modeling", The Journal of Systems and Software, vol. 58, no. 1, pp. 47-62, 2001.

[13] S. L. Ho, M. Xie and T. N. Goh, "A Study of the Connectionist Models for Software Reliability Prediction", in Computers and Mathematics with Applications Vol. 46, pp. 1037-1045, 2003.

[14] Aljahdali S, Debnath NC Improved Software Reliability Prediction through Fuzzy Logic Modeling. IASSE (2004).

[15] Aggarwal G, Gupta VK Neural Network Approach to Measure Reliability of Software Modules: A Review. International Journal of Advances in Engineering Sciences (2013).

[16] Quah, T. and Thwin, M. Application of Neural Networks for Predicting Software Development Faults (2000) Using Object-Oriented Metrics Proceedings of Ninth International Conference on Neural Information Processing(ICONIP_02), vol. 5, pp. 2312-2316.

[17] Khatatneh K, Mustafa T (2009) Software Reliability Modeling Using Soft Computing Technique. European Journal of Scientific Research 26: 147-152.

[18] Chatterjee S, Nigam S, Singh JB, Upadhyaya LN (2011) Application of Fuzzy Time Series in Prediction of Time Between Failures & Faults in Software Reliability Assessment. Fuzzy Inf Eng 3: 293-309.

[19] S. N. Sivanandam & S. N. Deepa, Principle of soft computing 2nd Edition, wiley publication India.