Journal of
Information Technology & Software Engineering

Kaswan, et al., J Inform Tech Softw Eng 2015, 5:1
DOI: 10.4172/2165-7866.1000144

Research Article Open Access

Software Reliability Modeling using Soft Computing Techniques: Critical Review

Kaswan KS[1]*, Choudhary S[2] and Sharma K[3]

[1]*Research Scholar, Department of Computer Science, Banasthali University, Banasthali, Rajasthan, India*
[2]*Department of Computer Science, Banasthali University, Rajasthan, India*
[3]*Department of Computer Science and Engineering, Delhi Technological University, Delhi, India*

Abstract

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.

Keywords: Neural network; Fuzzy logic; Genetic programming; Cuckoo search; Soft computing and software reliability

Introduction

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 [1]. 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 [2-4]. 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. 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. Soft computing techniques are the collection of different concepts and techniques which aim to overcome the difficulties encountered in real world problems. It deals with the problems that seem to be imprecise, uncertain and difficult to categorize. One may see soft computing as an attempt to mimic natural creatures: plants, animals, human beings, which are soft, flexible, adaptive and clever. In this sense soft computing is the name of a family of problem-solving methods that have analogy with biological reasoning and problem solving. This paper is organized as follows: section II includes different soft computing techniques. In section III we discuss the role of different soft computing techniques in software reliability models. Section IV contains description of soft computing technique in software reliability models. Comparison of these soft computing techniques in terms of modeling capabilities is given in section V. Finally, the paper concludes with section VI.

Soft-computing Techniques

Soft computing is an association of computing methodologies that includes as its principal members' fuzzy logic, chaos theory, neuro-computing, evolutionary computing and probabilistic computing [5]. Soft computing is based on natural as well as artificial ideas. It is referred as a computational intelligence [6]. 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.. In this section we have discussed about the classification of existing soft computing techniques as shown in Figure 1.

In Figure 1 we discussed about some soft computing techniques as neural networks, Fuzzy Logic, Support vector machine (SVM), Evolutionary computing, Bayesian Network and Chaos Theory. Then some technique is also used in the combination with the others as Neuro-Fuzzy, the combination of Neural Network and Fuzzy Logic. Evolutionary Computing System is further divided into Evolutionary Algorithm and Swarm Intelligence techniques.

Neural Networks

According to Nigrin A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates asynchronously, on local information; thus there is no overall system clock. Applications of neural networks are character recognition, image compression, stock market prediction, traveling salesman's problem, medicine, electronic nose, security and loan applications.

Support vector machine

Boser, Guyon, and Vapnik developed Support Vector Machine (SVM) was introduced, in COLT-92. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Support vector machines (SVM) have both a solid mathematical background and good performance in practical

***Corresponding author:** Kaswan KS, Research Scholar, Department of Computer Science, Banasthali University, Banasthali, Rajasthan, India, Tel: 228341; E-mail: KASWANKULDEEP@GMAIL.COM

Received January 29, 2015; **Accepted** March 31, 2015; **Published** April 13, 2015

Citation: Kaswan KS, Choudhary S, Sharma K (2015) Software Reliability Modeling using Soft Computing Techniques: Critical Review. J Inform Tech Softw Eng 5: 144. doi: 10.4172/2165-7866.1000144

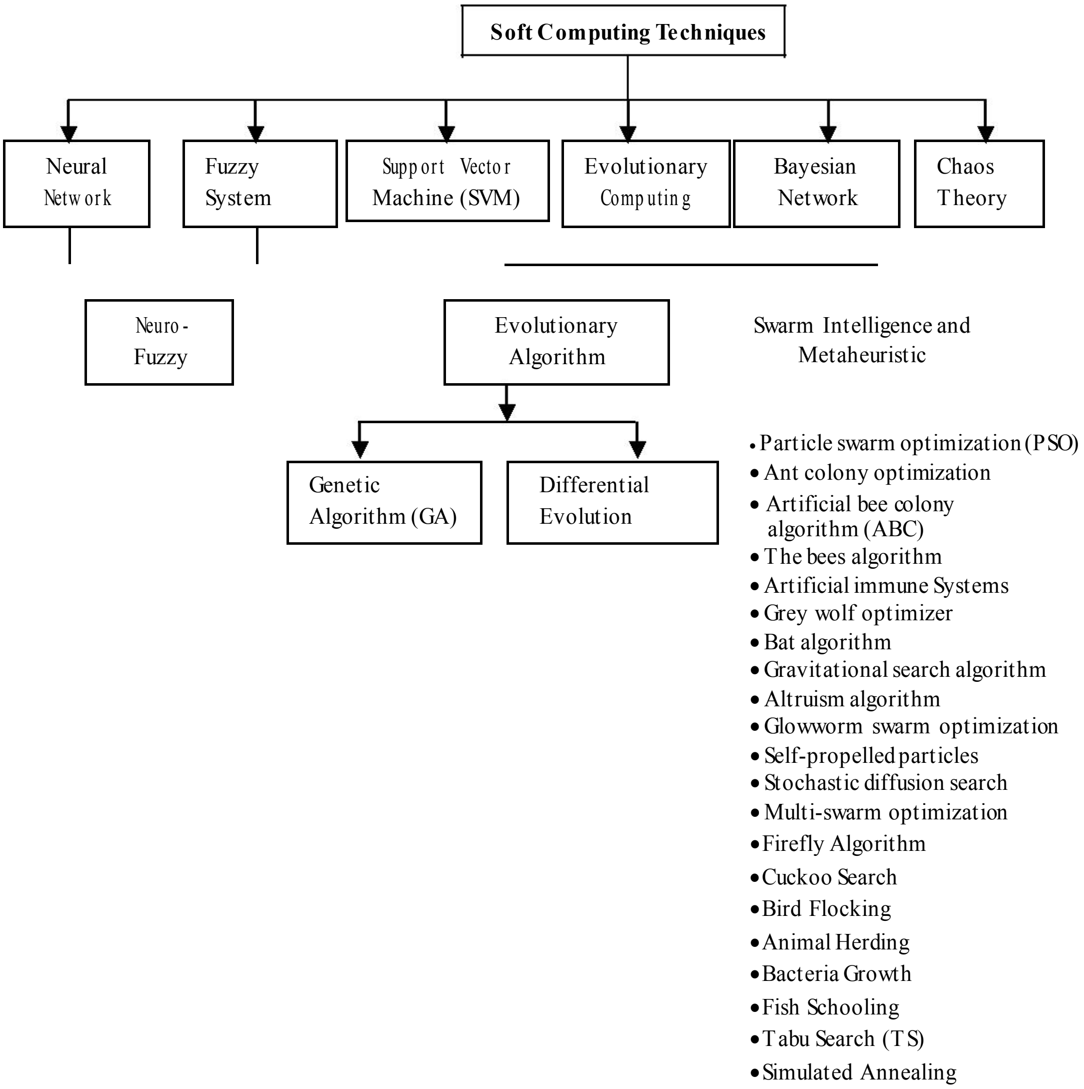

Figure 1: Soft Computing Techniques.

applications, such as image processing, artificial intelligence, medical, pattern recognition, machine learning, applied statistics, business intelligence, and information technology.

Fuzzy logic

Developed by Lotfi A. Zadeh at the University of California in Berkeley. It is a multi-valued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no and low/high, etc. [7,8]. The most significant application area of fuzzy logic has been in control field. Fuzzy control having been successfully applied to numerous problems, these includes fans control, complex aircraft engines and control surfaces, wheel slip control, helicopter control, automatic transmission, industrial and missile guidance.

Evolutionary computing

Evolutionary computing can be viewed as an adaptation of a probabilistic approach based on principles of natural evolution [9]. It can also defined as the stochastic search and optimization heuristic approach derived from the classic evolution theory, which are implemented on computers in the majority of cases [10]. Evolutionary algorithms have been successfully applied to numerous problems from different domains, bioinformatics, including optimization, automatic programming, signal processing, social systems [11].

Bayesian network

Bayesian networks are graphical models for reasoning under uncertainty, where the nodes represent variables (discrete or continuous) and arcs represent direct connections between them. various applications such as the impact of management style on statistical efficiency, studies of web site usability, operational risks, biotechnology, customer satisfaction surveys, healthcare systems and the testing of web services.

Chaos theory

A deterministic system is said to be chaotic whenever its evolution sensitively depends on the initial conditions. This property implies that two trajectories emerging from two different closes by initial conditions separate exponentially in the course of time. The necessary requirements for a deterministic system to be chaotic are that the system must be nonlinear, and be at least three dimensional [12]. Each soft computing technique can be used separately, but its complementary nature is its more powerful advantage. We can also create a hybrid system, combination of hard and soft computing, that can produce solutions to problems that are too complex or inherently noisy to tackle with conventional mathematical methods.

Potential Usages of Different Soft Computing Techniques in Software Reliability Models

Soft computing can be used for software faults diagnosis, reliability optimization and for time series prediction during the software reliability analysis. In this section we have discussed about application of soft computing technologies in software reliability.

Neural networks

Neural networks are simplified model of the biologic neuron system, it is massively parallel distributed processing system made up of highly interconnected neural computing elements that have the ability to learn and thereby acquire knowledge and make it available for use. Neural network has been applied for parameters estimation of the formal model and self learning process in order to predict the future outcomes. It has been shown that feed forward network can be applied for prediction. Back-error propagation is one of the most widely used neural network paradigms and has been applied successfully in a broad range of areas [13]. Karunanithi et al. [14,15] predict cumulative number of failure by design first neural network based software reliability model. They used feed-forward neural network, recurrent neural network and Elman neural network in their study and use execution time as the input of the network. They found that their models are better prediction models than some other statistical models [16]. Used connectionist models for software reliability prediction. Design the architecture of neural network by Falman's cascade correlation algorithm. They found that for end point prediction connectionist approach. Khoshgoftaar et al. [17] used the neural network for predicting the number of faults and introduced an approach for static reliability modeling. Then trained two neural networks; one with the complete set of principal components and one with the set of components selected by multiple regression model selection. Comparison of these models showed a better understanding of neural network software quality models. Sitte [18] compared, purposed neural network based software reliability prediction model, with recalibration for parametric models using some meaningful predictive measures with same datasets. Result showed that prediction with the help of neural network approach is better than others. Cai et al. [19] presented neural network based method for software reliability predictions, used back propagation algorithm for training. Performance of this purposed approach is evaluated by using different number of input nodes and hidden nodes. Result showed that its performance depends upon the nature of the handled data sets. Ho et al. [20] investigated a modified Elman recurrent neural network in modeling and predicting software failures and then performed a comprehensive study of connectionist models and their applicability to software reliability prediction and found them to be better and more flexible than the traditional models. Tian and Noore [21] proposed an on-line adaptive software reliability prediction model using evolutionary connectionist approach based on multiple-delayed-input single-output architecture, which showed better performance with respect to next-step predictability compared to existing NN model. Tian and Noore [22] presented an evolutionary neural network based method for software reliability prediction, used multiple-delayed-input single output architecture. Result showed that neural network architecture has a great impact on the performance of the network. Yu Shen Su et al. [23] purposed a model that uses the neural network approach to build a dynamic weighted combinational model. Then compared the performances of the neural network models with some conventional SRGMs from three aspects: goodness of fit, prediction ability for short-term prediction and long-term prediction. Result shows that purposed model has more accuracy with both goodness of fit and the prediction ability compared to existing conventional models. Viswanath [24] proposed two models such as neural network based exponential encoding and neural network based logarithmic encoding for prediction of cumulative number of failures in software. He use execution time as the input and applied on four data sets. Result showed that its result is better that other statistical model. Hu et al. [25] proposed an artificial neural network model to improve the early reliability prediction for current projects/releases by reusing the failure data from past projects/releases. Better prediction performance is observed in early phase of testing compared with original ANN model without failure data reuse. Su and Huang [26] proposed an artificial neural-network-based approach for software reliability estimation and modeling then further use of the neural network approach to build a dynamic weighted combinational model. The results obtained from the experiments show that the proposed model has a fairly accurate prediction capability

Kanmani et al. [27] purposed two neural network based software fault prediction models using Object-Oriented metrics. The results are compared with two statistical models using five quality attributes and found that neural networks do better Aljahdali et al. [28] explored connectionist artificial neural networks models as an alternative approach to derive these models by investigating the performance analysis of four different connectionist paradigms, multi-layer perceptron neural network, radial-basis functions, Elman recurrent neural networks and a neuro-fuzzy model, for modeling the software reliability prediction. In [29,30] Singh et al. used feed forward neural network for software reliability prediction and applied back propagation algorithm to predict software reliability growth trend. Also demonstrated a comparative analysis between the proposed approach and three well known software reliability growth prediction models using seven different failure datasets collected from standard software projects to test the validity of the presented method. Nirvikar Katiyar et al. [31] develop a non-parametric software reliability prediction system based on the neural network effects to improve the predictability by utilizing the diversity among the combined component neural networks. Huang et al. [32] derived software reliability growth models (SRGM) based on non-homogeneous poison processes (NHPP) using a unified theory by incorporating the concept of multiple change-points into software reliability modeling. Manjubala Bisi et al. [33] proposed a neural network based software reliability model to predict the cumulative number of failures based on Feed Forward architecture. The effect of encoding and the effect of different encoding parameter on prediction accuracy have been studied and its performance is tested using eighteen software failure data sets. Sandeep Kumar Jain et al. [34] proposed a method to estimate the reliability of the software consisting of components by using different neural network architectures. Then estimate the faults prediction behavior in the set of components over a cumulative execution time interval besides this the prediction of faults is estimated for the complete software. To predict the faults in each component of the software with the prediction of faults for the complete software for given cumulative execution time, apply the feed forward neural network architectures and its generalization capability.

Fuzzy system

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 [35]. Cai et al. [36] discussed the development of fuzzy software reliability models 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. Khalaf khatatneh [37] 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. Reformat [38] 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. Sultan Aljahdali et al. [39] investigated the use of fuzzy logic on building SRGM to estimate the expected software faults during testing process. 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. S. Chatterjee et al. [40] 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.

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. Kirti Tyagi et al. [41] proposed a model for estimating CBSS reliability, known as an adaptive neuro fuzzy inference system (ANFIS) that is based on these two basic elements of soft computing, neural network and fuzzy logic. ANFIS model gives a more accurate measure of reliability than the FIS model, as it reduces error from 11.74%, in case of FIS model, to 6.66% in ANFIS.

Genetic Algorithm (GA)

Genetic algorithm is a model of machine learning which derives its behavior from a metaphor of the process of evolution in nature. This is done by the creation within a machine of a population of individuals represented by chromosomes.

The fitness of each chromosome is determined by evaluating it against an objective function. To simulate the natural survival of the fittest process, best chromosomes exchange information to produce offspring chromosomes. The offspring solutions are then evaluated and used to evolve the population if they provide better solutions than weak populations members. Usually, the process is continued for a large number of generations to obtain a best-fit solution. Liang et al. [42,43] proposed a genetic algorithm optimizing the number of delayed input neurons and the number of neurons in the hidden layer of the neural network, predicting software reliability and software failure time. Sultan H. Aljahdali et al. [44,45] explored Genetic Algorithms (GA) as an alternative approach to derive software reliability models. For purposed system applicability three study sets; Military, Real Time Control and Operating System were used. Satya Prasad et al. [46] incorporate both imperfect debugging and change-point problem into the software reliability growth model (SRGM) based on the well-known exponential distribution the parameter estimation is studied. Proposed model is rated as better than the other considered models with respect to all the conditions are chosen.

Genetic Programming (GP)

Genetic programming can be viewed as an extension of the genetic algorithm, a model for testing and selecting the best choice among a set of results, each represented by a string. Genetic programming goes a step farther and makes the program or "function" the unit that is tested. Two approaches are used to select the successful program - cross-breeding and the tournament or competition approach. A difficult part of using genetic programming is determining the fitness function, the degree to which a program is helping to arrive at the desired goal. Costa et al. [47] Modeling software reliability growth with genetic programming. Experiments conducted to confirm the purposed hypothesis and demonstrate better results as compared to other traditional models and neural network model. E. Oliveira et al. [48] improve software reliability models using boosting techniques based on genetic programming. Boosting Technique combines several hypotheses of the training set to get better results. The most important improvement of this work is when consider models based on time and got excellent results by using just one function set. Y. Zhang et al. [49] Predicting for MTBF failure data series of software reliability by genetic programming algorithm. The evolution model of GP is then analyzed and appraised according to five characteristics criteria for some common-used software testing cases. Result showed the higher prediction precision and better applicability. Wasif Afzal et al. [50] discussed the suitability of using GP for software reliability growth modeling and highlight the mechanisms that enable GP to progressively search for fitter solutions. The experiments of using GP for software reliability growth modeling have indicated positive results. Eduardo Oliveira Costa et al. [51] introduced a new GP based approach, named (μ+λ)GP. This algorithm was introduced to improve the performance of GP. To evaluate this purposed algorithm, two kinds of models: based on time and on coverage were presented for experimental results, which is always better than classical GP. Zainab Al-Rahamneh et al. [52] proposed the use of Genetic Programming (GP) as an evolutionary com-putation approach to handle the software reliability modeling problem. Evaluate the GP developed model and results show that this purposed model is superior to other models such as Yamada S-shaped, Generalized Poisson, NHHP and Schneidewind reliability models.

Artificial Bee Colony (ABC)

Dervis Karaboga, in 2005, defined a new algorithm, motivated by the intelligent behavior of honey bees known as artificial bee colony. It is an optimized tool provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and bee's aim to discover the places of food sources with high nectar amount and finally the one with highest nectar. Tarun Kumar Sharma et al. [53] proposed a modified version of the ABC, the DABC (Dichotomous ABC), to improve its performance, in terms of converging to individual optimal point and to compensate the limited amount of search moves of original ABC. Also explored the applicability of the modified artificial bee colony algorithm to estimate the parameters of software reliability growth models (SRGM). The estimated model parameters were used to predict the faults in a software system during the testing process.

Ant colony

Ant Colony Optimization [54] is a technique which uses probability to solve problems where the computations are reduced with the help of graphs to get efficient paths. It has been applied to many fields as its robustness and is easy to collaborate with other methods. It has well performance to the optimization problem and has a good convergence rate. Changyou Zhenga et al. [55] proposed a parameter estimation method based on the Ant Colony Algorithm. Its results are derived from Numerical examples based on three sets of real failure data. Purposed method shows higher precision and faster convergence speed. Latha Shanmugam et al. [56] discussed a parameter estimation method based on Ant Colony Algorithm. The outcome of the experiment using six typical models demonstrated that this algorithm can be applied for estimating parameters. Higher precision and faster convergence speed is achieved through this method when compared with PSO algorithm. Latha Shanmugam et al. [57] studied enhancement and Comparison of Ant Colony Optimization Methods for Software Reliability Models. The Enhanced method shows significant advantages in finding the goodness of fit for software reliability model such as finite and infinite failure Poisson model and binomial models. It is comparatively giving better Estimation Accuracy (approximately 10%) than the existing Ant Colony Optimization.

Simulated Annealing (SA) algorithm

Simulated annealing (SA) is an iterative search method inspired by the annealing of metals [58,59] Starting with an initial solution and armed with adequate perturbation and evaluation functions, the algorithm performs a stochastic partial search of the state space. Nidhi Gupta et al. [60] the simulated annealing technique of mean field approximation for finding the possible minimum number of failed components in the sequential testing. These minimum numbers of failed components are depending upon the selection of time intervals or slots. Also purposed a new energy function with the mean field approximation. The algorithm of the whole process shows that this approach can generate the optimal solution. Pai and Hong [61] applied support vector machines (SVMs) for forecasting software reliability where simulated annealing (SA) algorithm was used to select the parameters of the SVM model. The experimental results reveal that the SVM model with simulated annealing algorithms (SVMSA) results in better predictions than the other methods. Mohamed Benaddy et al. [62] presented a hybrid approach based on the Neural Networks and Simulated Annealing. An adaptive simulated Annealing algorithm is used to optimize the mean square of the error produced by training the neural network, predicting software cumulative failure. The purposed adaptive Simulated Annealing gives better performance in execution time than the Real Coded Genetic Algorithm (RCGA), because of the search space, which reduced from a population of solutions for the RCGA to one solution for the proposed Simulated Annealing.

Tabu search algorithm

The Tabu Search (TS) is an optimization method, based on the premise that problem solving, in order to qualify as intelligent, and must incorporate adaptive memory and responsive exploration [63]. The Tabu method was partly motivated by the observation that human behavior appears to operate with a random element that leads to inconsistent behavior given similar circumstances. Caserta et al. [64] presented a new meta- heuristic-based algorithm for complex reliability problems. The algorithm effectively uses features of the Tabu Search paradigm, with special emphasis on the exploitation of memory-based mechanisms. It balances intensification with diversification via the use of short-term and long-term memory. The proposed algorithm proves to be robust with respect to its parameters and it is especially suited for very large scale instances of the reliability problem, when exact approaches are doomed to fail.

Cuckoo search algorithm

This algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the L´evy flight behavior of some birds and fruit flies. Cuckoo search algorithm is very successful in finding good and acceptable solutions to the problem of parameter estimation of Software Reliability Growth Models. This algorithm search strategy can efficiently navigate throughout the search space of the problem and locate very good solutions using fewer iterations and smaller populations. Najla Akram AL-Saati et al. [65] estimated parameters based on the available failure data. Cuckoo Search outperformed both PSO and ACO in finding better parameters tested using identical datasets, but worse in case of extended ACO. The Exponential, Power, S-Shaped, and M-O models are considered in this work. The search strategy of the cuckoo can efficiently navigate throughout the search space of the problem and locate very good solutions using fewer iterations and smaller populations

Summarization

In the past few decades a number of software reliability models have been analyzed, designed and evaluated. Soft computing plays an important role in the recent advancements in the software reliability growth models. Today these models included the application of different soft computing techniques such as Neural Network (NN), Fuzzy Logic, Genetic Algorithms (GA), Genetic Programming (GP), Artificial Bee Colony (ABC) and Ant Colony etc., a brief summary of soft computing techniques in software reliability models have been summarized in this section as shown in Table 1.

In this summary, we notice that different soft computing techniques are used in different shapes with these models. We observed that Neural Network approach is more liked by the researchers in software reliability models. Genetic Programming provides more accuracy than other soft computing techniques. Cuckoo Search, Stimulated Annealing and Tabu Search are used in this field but not so widely yet. This table data is important in case of comparison and selection of soft computing technique in terms of modeling capabilities.

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

Comparisons is very useful in case of optimal selection, user can view all possible choices on a single plate form and can select the best suited as his/her requirement. In Table 2, 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. 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. Ant colony, Stimulated Annealing, Tabu Search and Cuckoo Search can also be used for most of the modeling capabilities except only small data set capabilities. The rapid growth of soft computing techniques suggests that the impact of these algorithms will be used increasingly for software reliability models in the coming years. This table will help computer scientist who are keen to contribute their works to the field of software reliability.

Conclusion

In this paper, we have discussed about the work done by the various researchers, with the endeavor made to include as many references as possible from year 1990 to 2014. Based on this paper, we thrash out some

s.no	Study	Technology used	Project data	Summary of Result	References
1	N.Karunanithi,Y.K.Malaiya and D. Whitley	Neural Network	DS-1K.Matsumoto(1988) DS-2J.D. Musa(1987)DS-3M. Ohba(1984)	Explores the use of feed-forward neural networks as a model for software reliability growth prediction	[14]
2	N.Karunanithi,D.Whitley, Y.K. Malaiya	Neural Network	DS form Yoshiro Tohma project	After training the neural network with afailure history up to time t you can use thenetwork to predict the cumulative faults atthe end of a future testing and debuggingsession.	[15]
3	Taghi M.Khoshgoftaa,RobertM. Szabo and Peter J.Guasti	Neural Network	Data collected from three similar systems(Kernel2,Kernel3andKernel4)and twodissimilarsystems(Kernel5andKernel6)	The excellent predictive results observed inthe neural network models indicate thatneural networks should be seriouslyconsidered as an effective modeling too forsoftware engineers	[17]
4	Renate Sitte	Neural Network	Musa data-sets S1 & SS3	Neural Networks are not only much simpler to use than the recalibration method, but that they are equal or better trend predictors	[18]
5	S.L.Ho,M.Xie andT.N.Goh	Neural Network	DS-1:Military computer system DS-2:Musa-OkumotoModelDS-3: Goel Okumoto Model	The Elman model is comparatively better than the Jordan model and very much superior than the feed -forward model	[20]
6	NirvikarKatiyarand Raghuraj Singh	Neural Network	DS1and DS2 from handbookofsoftwarereliability engg.LyuM.R.(1996)New York	Purposed system achieves significantly lower prediction error compared with the single NN and traditional SRGMs	[31]
7	Sultan Aljahdali et al.	Neural Network	Real-time command and Control processing Commericial and military applications	NN provide models with smaller normalized root of mean of the square of error than the regression model in all considered cases	[]
8	Yu Shen Su, Chin-Yu Huang,YiShin and Jing Xun Chen	Neural Network	Realcommand & control project, John D Musa, Bell Lab	Achieve a dynamic weighted combinational model	[23]
9	Sultan H. Aljahdali and Khalid A. Buragga	Neural Network	Rea l time control project	The Elman recurrent NN is a robust technique for function prediction due capturing the dynamic behavior of the data set.	[28]
10	Manjubala Bisi et al.	Neural Network	18differentdatasets(Military, Real time System,RealtimeCommandandControl,On-line data Entry etc.)	A Feed Forward neural network with two encoding scheme such as exponential andlogarithmic function has been proposed.	[33]
11	Sandeep Kumar Jain & Manu Pratap Singh	Neural Network	Data collected from localtraining set.	Estimated the faults prediction behavior in the set of components over a cumulative execution time interval besides this the prediction of faults is estimated for the complete software.	[34]
12	S. Chatterjee, S. Nigam,Singh,Upadhyaya	Fuzzy Logic	DS-1: Musa J D(1975) software reliability data DS-2: Pham H(2006) system software reliability	Purposed models are flexible, assumption free and very simple in computation. It not requiredanyde-fuzzifiedtechniques separately and computation time is reduced	[40]
13	Sultan Aljahdali	Fuzzy Logic	Realcommand & control project, Militaryandoperatingsystem,JohnDMusa, Bell lab.	Developed models provide high performance modeling capabilities.	[39]
14	Khalaf Khatatneh andThaer Mustafa	Fuzzy Logic	datasetfrom command and controlapplications,Musa,John D.	Developed model can predict accurate results in most points of the target database	[37]
15	Sultan H. Aljahdali and MohammedE.El-Telbany	Genetic Algorithm	Data from threeprojects Military,RealSystem Controland Operating system	Measured the predictability of software reliability using ensemble of models which performed better than the single model and also find that the weighted average combining method for ensemble has a better	[44]
16	Sona Ahuja, Guru Sarand Mishra and Agam Prasad Tyagi	Genetic Algorithm	Musa Data-set, software reliability prediction andapplication, 1985	GA based hybrid stochastic search technique,has turned out to be good tool for optimizedsimulated trajectory for variable which areimportant performance indicatorto predictqualityof reliability of the predicted softwarefailure	[66]
17	Sultan H. Aljahdali and Mohammed E.	Genetic Algorithm	Data from three projects.They are Military,Real Time Control and Operating System.	As far as the predictability of the single AR model and ensemble of AR models trained byGA algorithm over the trained and test data isconcerned,the ensemble of modelsperformed better than the single model. Also,find that the weighted average combiningmethod forensemble has a betterperformance in a comparison with averagemethod.	[45]
18	R.Satya Prasad, O.Naga Raju and R.R.L Kantam	Genetic Algorithm	DS-1:Misra,P.N.,1983Software reliability analysis.IBM Syst. DS-2:Pham,H.,1993.Software reliability assessment:Imperfect debugging and multiple failure types in softwaredevelopment	This model is rated as better than the other considers models with respect to all conditions are chosen.	[46]
19	Eduardo Oliveira Costa et al.	Genetic Programming	DS-1John Musa at Bell Telephone Laboratories DS-2 failure data of a program called Space	GP is a suitable tool to discover an equation to modeling software reliability and is also able to discover the equation that better represents the data [	[47]

20	E.Oliveira,A.Pozo, and S.Vergilio	Genetic Programming	DS-1 John Musa atBellTelephone LaboratoriesDS-2failure data of a program called Space	Use boosting techniques to improve software reliability models based on Genetic Programming.	[48]
21	Wasif Afzal and Richard Torkar	Genetic Programming	Data available at:http://www.gp-field-guide.org.uk,2008.	The experiments of using GP for software reliability growth modeling have indicated positive results, which warrant further investigation with larger real-world industrial data sets.	[50]
22	Eduardo Oliveira Costa, Aurora Trinidad Ramirez Pozo, and Silvia Regina Vergilio	Genetic Programming	DS-1:software reliability data by J Musa(1980)DS-2: space program	Purposed(μ+λ) GPsystem results arealways better thanthe classical technique results also improves the performance forsmall datasets	
23	Zainab Al-Rahamneh, Mohammad Alaa F. Sheta, Sulieman Bani-Ahmad, Saleh Al-Oqeili	Genetic Programming	Data set from Y. Tohmapublished in IEEEtransactions onSoftwareEngineering, Vol. 15, No. 3,1989	Here adopted recalibrated and adjusted GP operators to speed up the convergence process	[51]
24	Tarun Kumar Sharma, MilliePant and Ajith Abraham	Artificial Bee Colony	IEEE Congress onEvolutionary Computation,Sheraton,VancouverWallCentre,Vancouver,BC,Canada,2006.,10428–10435	Modified version have a better success rate than original ABC	[52]
25	Changyou Zhenga,Xiaoming Liua,Song Huanga and Yi Yaoa	Ant Colony	SYS1, SYS2, SYS3 from Musa dataset	Experiments with three typical models show that this ant colony based algorithm demonstrates good applicability	[53]
26	Latha Shanmugam and Lilly Florence	Ant Colony	Musa real time data-set	Based on the results, it was found that the Enhanced Ant Colony Optimization Method is giving better estimation accuracy than Existing ACO method. Time and Space Complexity is also reduced.	[55]
27	The UK's expert	Ant Colony	failure data which is given in the Musa data Set from the DACS Web Site	Based on the results, we found that the proposedSimulatedAnt Colony Optimization Method is giving 15% better estimation accuracy	[57]
28	Najla Akram AL-Saati	Cuckoo Search	first group dataset: A Sheta. Et al. (2006)second group dataset:John Musa of Telephone Laboratories	CS do better than both PSO and ACO infinding better parameters tested using identical datasets.	[65]
29	Nidhi Gupta and Manu Pratap Singh	Stimulated Annealing	a series of tests conducted under certain stipulated conditions on 1,000 software components	Results show that after applying the sequential testing with MFA we can optimizethe number of failures up to a minimumvalue	[60]
30	Mohamed Benaddy and Mohamed Wakrim	Stimulated Annealing	John Musa of Bell Telephone Laboratories	The Performance in execution time of the proposed adaptive simulated Annealing is better than the RCGA, because of the search space, which reduced from a population of solutions for the RCGA to one solution for the proposed Simulated Annealing	[62]
31	Kirti Tyagi and Sharma	Neuro Fuzzy	collecte data from 47 classroom-based projects	Results show that the ANFIS improves the reliability evaluation of the FIS technique	[41]
32	M. Casertaa,and A.Márquez Uribe	Tabu Search	Berman and Ashrafi (1993)	Proposed algorithm is robust with respect to its parameters and it is especially suited for very large scale instances of the reliability problem.	[64]

Table 1: Summary of Soft Computing Techniques in Software Reliability Models.

Sr. No	Technology Used	Explain outputs	Suitability for small data sets	Can be re-designed for new data set	Reasoning process is visible	Applicability for complex models	Either known facts considered
1	Neural	No	No	No	No	Yes	Partially
	Networks						
2	Fuzzy Logic	Yes	Yes	Yes	Yes	Yes	Yes
3	Genetic	Partially	Partially	Yes	Yes	Partially	No
	Algorithms						
4	Genetic	Yes	No	No	No	Yes	No
	Programming						
5	Artificial Bee	Yes	Partially	Partially	No	Yes	Yes
	Colony						
6	Ant Colony	Yes	No	Yes	Yes	Yes	Yes
7	Stimulted	Yes	No	Yes	Yes	Yes	Yes
	Anealing						
8	Tabu Search	No	Partially	Yes	Yes	Yes	Yes
9	Cuckoo Search	Yes	No	Yes	Yes	Yes	Yes

Table 2: Comparison of Soft Computing Techniques in terms of Modeling Capabilities.

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. Further we compared soft computing techniques in terms of modeling capabilities, which enhances the selection process of soft computing technique for software reliability models.

References

1. Agarwal kk, Singh Y (2005) Software Engineering. New Age International Publisher, New Delhi.
2. Kapur PK, Garg RB (1990) Cost reliability optimum release policies for a software system with testing effort. Operations Research 27: 109-116.
3. Musa JD (1999) Software Reliability Engineering: More Reliable Software, Faster Development and Testing, McGraw-Hill.
4. Musa JD, Iannino A, Komodo K (1987) Software Reliability: Measurement, Prediction and Application. McGraw-Hill.
5. Bonissone P, Chen YT, Goebel K, Khedkar P (1999) Hybrid Soft Computing Systems: Industrial and Commercial Applications. Proceedings of the IEEE 87: 1641-1667.
6. Das SK, Kumar A, Das B, Burnwal AP (2013) On Soft computing Techniques in Various Areas. ACER.
7. Zadeh LA (1968) Fuzzy Algorithms. Information and Control 12: 94-102.
8. Zadeh LA (1965) Fuzzy Sets. Information and Control 8: 338-353.
9. Kuo W, Prasad VR (2000) An annotated Overview of System Reliability Optimization. IEEE Transaction on Reliability 49: 176-186.
10. Streichert F (2002) Introduction to Evolutionary Algorithms. Frankfurt MathFinance Workshop, University of Tubingen, Germany.
11. Abraham A (2005) Handbook of Measuring System Design. In: Peter H, Sydenham, Thorn R (eds.). Evolutionary Computation. John Wiley & Sons, Ltd.
12. Boccaletti S (2000) The control of chaos: theory and applications. Elsevier Science 329: 103-197.
13. Aggarwal G, Gupta VK (2013) Neural Network Approach to Measure Reliability of Software Modules: A Review. International Journal of Advances in Engineering Sciences.
14. Karunanithi N, Malaiya YK, Whitley D (1992) Prediction of software reliability using neural networks. Proceedings of the Second IEEE International Symposium on Software Reliability Engineering.
15. Karunanithi N, Whitley D, Malaiya YK (1992) Using neural networks in reliability prediction. IEEE Software 9: 53-59.
16. Karunanithi N, Whitley D, Malaiya YK (1992) Prediction of software reliability using connectionist models. IEEE Trans Software Engg 18: 563-573.
17. Khoshgoftaar TM, Szabo RM, Guasti PJ (1995) Exploring the Behavior of Neural-network Software Quality Models. Software Engg J 10: 89-96.
18. Sitte R (1999) Comparison of Software Reliability Growth Predictions: Neural networks vs. parametric recalibration. IEEE transactions on Reliability 48: 285-291.
19. Cai KY, Cai L, Wang WD, Yu ZY, Zhang D (2001) On the neural network approach in software reliability modeling. Journal of Systems and Software 58: 47-62.
20. Ho SL, Xie M, Goh TN (2003) A Study of the Connectionist Models for Software Reliability Prediction. Computers and Mathematics with Applications 46: 1037-1045.
21. Tian L, Noore A (2005) On-line prediction of software reliability using an evolutionary connectionist model. Journal of Systems and Software 77: 173-180.
22. Tian L, Noore A (2005) Evolutionary neural network modeling for software cumulative failure time prediction. Reliability Engineering and System Safety 87: 45-51.
23. Su SY, Huang CY, Chen YS, Chen JX (2005) An Artificial Neural-Network-Based Approach to Software Reliability Assessment. TENCON, IEEE Region 10: 1-6.
24. Viswanath SPK (2006) Software Reliability Prediction using Neural Networks. PhD. Thesis, Indian Institute of Technology Kharagpur.
25. Hu QP, Dai YS, Xie M, Ng SN (2006) Early software reliability prediction with extended ANN model (2006) Proceedings of the 30th Annual International Computer Software and Applications Conference, pp. 234-239, 2006.
26. Su YS, Huang CY (2006) Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. Journal of Systems and Software 80: 606–615.
27. Kanmani S, Uthariara VR, Sankaranarayanan V, Thambidurai P (2007) Object-oriented software failure fault prediction using neural networks. Information and Software Technology 49: 483-492.
28. Aljahdali AS, Buragga KB (2008) Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study. ICGST-AIML Journal.
29. Singh Y, Kumar P (2010) Application of feed-forward networks for software reliability prediction. ACM SIGSOFT Software Engineering Notes 35: 1-6.
30. Singh Y, Kumar P (2010) Redirection of Software Reliability Using feed Forward Neural Networks. International conference on Computational Intelligence and software Engineering.
31. Katiyar N, Singh R (2011) Effect of Neural Network for Prediction of Software Reliability. VSRD-IJCSIT 1: 490-500.
32. Huang CY, Lyu MR (2011) Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models. IEEE Transaction on Reliability 60: 498-514.
33. Bisi M, Goyal NK (2012) Software Reliability Prediction using Neural Network with Encoded Input. International Journal of Computer Applications.
34. Jain SK, Singh MP (2013) Estimation for Faults Prediction from Component Based Software Design using Feed Forward Neural Networks. IJARCCE.
35. Aljahdali S, Debnath NC (2004) Improved Software Reliability Prediction through Fuzzy Logic Modeling. IASSE.
36. Cai KY, Wen CY, Zhang ML (1991) A critical review on software reliability modeling. Reliability Engineering and System Safety 32: 357-371.
37. Khatatneh K, Mustafa T (2009) Software Reliability Modeling Using Soft Computing Technique. European Journal of Scientific Research 26: 147-152.
38. Reformat M (2005) A fuzzy-based multimodel system for reasoning about the number of software defects. International Journal of Intelligent Systems 20: 1093-1115.
39. Aljahdali S (2011) Development of Software Reliability Growth Models for Industrial Applications Using Fuzzy Logic. Journal of Computer Science 7: 1574-1580.
40. 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.
41. Tyagi K, Sharma A (2014) An adaptive neuro fuzzy model for estimating the reliability of component-based software systems. Applied Computing and Informatics 10: 38-51.
42. Tian L, Noore A (2005) Evolutionary neural network modeling for software cumulative failure time prediction. Reliability Engineering &amp; System Safety 87: 45-51.
43. Tian L, Noore A (2005) On-line prediction of software reliability using an evolutionary connectionist model. Journal of Systems and Software 77: 173-180.
44. Aljahdali SH, El-Telbany ME (2008) Genetic Algorithms for Optimizing Ensemble of Models in Software Reliability Prediction. ICGST-AIML.
45. Aljahdali SH, El-telbany ME (2009) Software Reliability Prediction Using Multi-Objective Genetic Algorithm. IEEE.
46. Prasad RS, NagaRaju O, Kantam RRL (2010) SRGM with Imperfect Debugging by Genetic Algorithms. International Journal of software engineering & applications 11.
47. Costa E, Vergilio S, Pozo A, Souza P (2005) Modeling software reliability growth with genetic programming. ISSRE.

48. Oliveira E, Pozo A, Vergilio S, (2006) Using boosting techniques to improve software reliability models based on genetic programming. ICTAI.

49. Zhang Y, Chen H (2006) Predicting for MTBF failure data series of software reliability by genetic programming algorithm. In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Washington, USA.

50. Afzal W, Torkar R (2008) Suitability of genetic programming for software reliability growth modeling. IEEE

51. Costa OE, Pozo ATR, Vergilio SR (2010) A Genetic Programming Approach for Software Reliability Modeling. IEEE Transactions on Reliability.

52. Al-Rahamneh Z, Reyalat M, Sheta FA, Bani-Ahmad S, Al-Oqeili S (2011) A New Software Reliability Growth Model:Genetic-Programming-Based approach. Journal of Software Engineering and Applications, 4: 476-481.

53. Sharma TK, Pant M, Abraham A (2011) Dichotomous Search in ABC and its Application in Parameter Estimation of Software Reliability Growth Models. IEEE.

54. Li, W.,Q. Yin and X. Zhang (2010) Continuous quantum ant colony optimization and its application to optimization and analysis of induction motor structure. Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, Changsha.

55. Zhenga C, Liua X, Huanga S, Yaoa Y (2011) A Parameter Estimation Method for Software Reliability Models. Procedia Engg 15: 3477-3481.

56. Shanmugam L, Florence L (2012) A Comparison of Parameter best Estimation Method for software reliability models. International Journal of Software Engineering & Applications.

57. Shanmugam L, Florence L(2013) Enhancement and comparison of ant colony optimization for software reliability models. Journal of Computer Science 9: 1232-1240.

58. Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. Science, 220: 498– 516.

59. Cerny V (1985) Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. Journal of Optimization Theory and Application 45: 41–51.

60. Gupta N, Singh MP (2006) Evolutionary algorithms, simulated annealing and tabu search: a comparative study. IJE Transactions B: Applications.

61. Pai PF, Hong WC (2006) Software reliability forecasting by support vector machines with simulated vector machines with simulated annealing algorithms. The Journal of Systems and Software 79: 747–755.

62. Benaddy M, Wakrim M (2012) Simulated Annealing Neural Network for Software Failure Prediction. International Journal of Software Engineering and Its Applications.

63. Glover F, Laguna M (1997) Tabu Search: A Tutorial. Kluwer Academic Publishers, Boston.

64. Caserta M, Uribe MA (2006) Tabu search-based meta-heuristic algorithm for software system reliability problems. Computers & Operation Research 36: 811-822.

65. Akram N. AL-Saati, Abd-AlKareem M (2013) The Use of Cuckoo Search in Estimating the Parameters of Software Reliability Growth Models. International Journal of Computer Science and Information Security.

66. Ahuja S, Mishra GS, Tyagi AP (2002) Jelinski – Moranda Model for Software Reliability Prediction and its G. A. based Optimised Simulation Trajectory.D. E. I. Dayalbagh, Agra.