A COMPREHENSIVE REVIEW ON SOFTWARE RELIABILITY MODELING USING SOFT COMPUTING TECHNIQUES

Anusha Merugu[1] Chaithanya Kadari[2]
[1,2]Assistant Professor, Dept of CSE, Sri Venkateswara Engineering College, Suryapet.

Abstract-

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. In this paper, we have given a review of existing soft computing techniques, and then basically examined the work done by different researchers in the field of software reliability. Further to this, we have likewise analyzed soft computing techniques regarding software reliability modeling abilities.

IndexTerms- *Neural Network,Fuzzy Logic, Genetic Programming, Cuckoo Search, Soft Computing And Software Reliability.*

I. INTRODUCTION

Software building is a control whose point is the generation of value software, that is conveyed on time, inside spending plan, and that fulfills its prerequisites [1]. Software Engineering assumes a noteworthy job in software life on the grounds that there is dependably a requirement for the high caliber. Software reliability is the most quantifiable part of its quality.

Software reliability can be characterized as the likelihood of disappointment free task for a predetermined timeframe in a predefined situation [2] [3] [4]. 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. The client can get to the present and future reliability through testing utilizing these models, just as settle on choices about the software, for example, regardless of whether the item is discharged in its present state or require further testing so as to enhance the nature of 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.

II. SOFT-COMPUTING TECHNIQUES

In software engineering, soft computing (now and then alluded to as computational insight, however CI does not have a concurred definition) is the utilization of estimated answers for computationally hard assignments, for

example, the arrangement of NP-complete issues, for which there is no known calculation that can register a correct arrangement in polynomial time. Soft computing contrasts from ordinary (hard) computing in that, in contrast to hard computing, it is tolerant of imprecision, vulnerability, incomplete truth, an estimate. Basically, the good example of soft computing is the human personality.

The main constituents of Soft Computing (SC) are Fuzzy Logic (FL), Evolutionary Computation (EC), Machine Learning (ML) and Probabilistic Reasoning (PR), with the last subsuming conviction systems and parts of learning hypothesis.

Soft Computing turned into a formal territory of concentrate in Computer Science in the mid-1990s.[1] 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 processability, 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.[2][3] .

we discussed about the different classification schemes of existing soft computing techniques as shown in Figure-1.

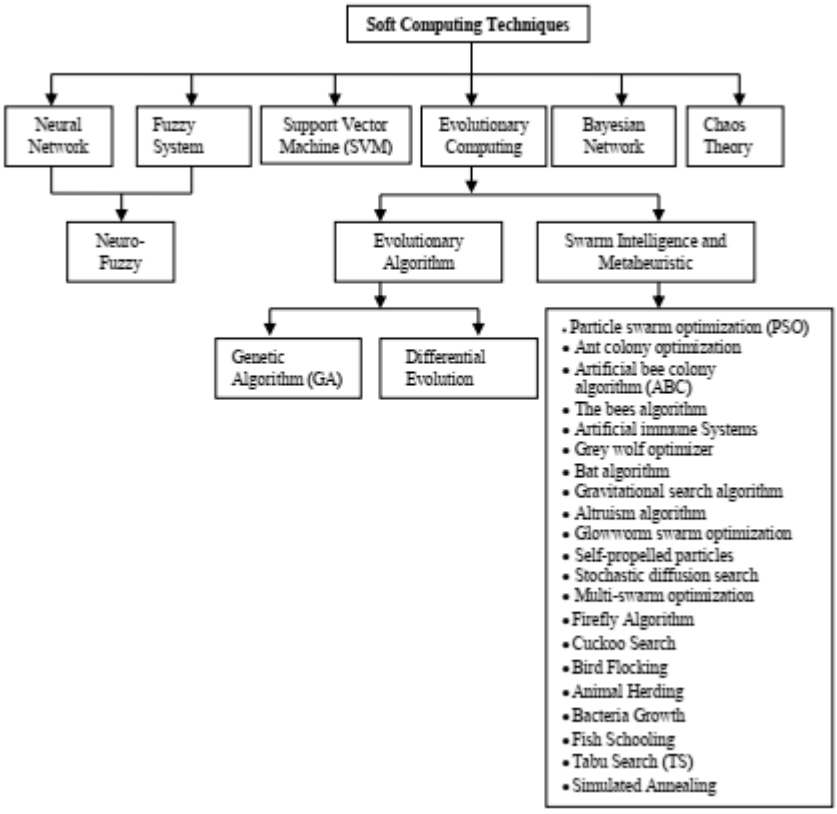

Fig. 1. Soft Computing Techniques

In figure-1 we talked about some soft computing techniques like neural systems, Fuzzy Logic, Support vector machine (SVM), Evolutionary computing, Bayesian Network, and Chaos Theory. At that point, a few techniques are additionally utilized in the blend with the others as Neuro-Fuzzy, the mix of Neural Network and Fuzzy Logic. Developmental Computing System is additionally partitioned into an Evolutionary Algorithm and Swarm Intelligence techniques.

2.1 Neural Networks

As indicated by Nigrin, A. (1993) neural system is a circuit made out of countless preparing components that are neurally based. Every component works non-concurrently, on neighborhood data; along these lines, there is no general framework clock. Uses of neural systems are character acknowledgment, picture pressure, stock exchange forecast, voyaging sales rep's concern, medication, electronic nose, credit applications, and security.

2.2 Support Vector Machine

Boser, Guyon, and Vapnik created Support Vector Machine (SVM) were in 1992, presented, in COLT-92. Bolster vector machines (SVMs) are a lot of related administered learning techniques utilized for arrangement and relapse. Bolster vector machines (SVM) have both a strong scientific foundation and great execution in pragmatic applications, for example, picture handling, computerized reasoning, medicinal, design acknowledgment, machine learning, connected insights, business knowledge, and data innovation.

2.3 Fuzzy logic

Created by Lotfi A. Zadeh in 1965, at the University of California in Berkeley. It is a multi-esteemed rationale that enables middle of the road esteems to be characterized between customary assessments like genuine/false, yes/no and low/high, and so on [7] [8]. The most huge application territory of fluffy rationale has been in the control field. Fluffy control having been effectively connected to various issues, these incorporate fans control, complex air ship motors and control surfaces, wheel slip control, helicopter control, programmed transmission, mechanical and rocket direction.

2.4 Evolutionary Computing

Transformative computing can be seen as an adjustment of a probabilistic methodology dependent on the standards of regular advancement [9]. It can likewise be characterized as the stochastic search and enhancement heuristic methodology got from the great advancement hypothesis, that are actualized on PCs typically [10]. Developmental calculations have been effectively connected to various issues from various spaces, bioinformatics, including improvement, programmed programming, flag handling, social frameworks [11].

2.5 Bayesian Network

Bayesian systems are graphical models for a reason under vulnerability, where the hubs speak to factors (discrete or consistent) and bend speak to coordinate associations between them. different applications, for example, the effect of the executive's style on measurable proficiency, investigations of site ease of use, operational dangers, biotechnology, consumer loyalty overviews, social insurance frameworks and the testing of web administrations.

2.6 Chaos Theory

A deterministic framework is said to be clamorous at whatever point its development delicately relies upon the underlying conditions. This property infers that two directions rising up out of two unique closes by introductory conditions separate exponentially over the span of time. The fundamental prerequisites for a deterministic framework to be disordered are that the framework must be nonlinear, and be no less than three dimensional [12].

Each soft computing innovation can be utilized independently, yet its integral nature is its all the more dominant favorable position. We can likewise make a crossbreed framework, a blend of hard and soft computing, that create answers for issues that are excessively perplexing or characteristically loud to handle with traditional scientific strategies.

POTENTIAL USAGES OF SOFT COMPUTING TECHNIQUES IN SOFTWARE RELIABILITY MODELING

Soft computing techniques can be utilized for software deficiencies conclusion, reliability improvement and for time arrangement expectation amid the software reliability examination. In this segment, we examined the utilizations of soft computing advances in software reliability modeling.

a. Neural Networks

Neural systems are a disentangled model of the biologicneuron framework, it is enormously parallel conveyed preparing framework made up of very interconnected neural computing components that can learn and in this manner obtain information and make it accessible for use. The neural system has been connected for parameters estimation of the formal model and self-learning process so as to anticipate the future results. It has been demonstrated that feedforward system can be connected for forecast. Back-blunder spread is a standout amongst the most generally utilized neural system ideal models and has been connected effectively in an expansive scope of regions [13].

Karunanithi et al. [14] [15] 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 [16]. 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.

Khoshgoftaar et al. [17] utilized the neural system for foreseeing the quantity of flaws and presented a methodology for static reliability modeling. At that point prepared two neural systems; one with the total arrangement of foremost segments and one with the arrangement of parts chosen by different relapse demonstrate choice. Correlation of these models demonstrated a superior understanding of neural system software quality models.

Sitte [18] thought about, purposed neural system based software reliability expectation show, with recalibration for parametric models utilizing some important prescient measures with the equivalent datasets. The outcome demonstrated that expectation with the assistance of the neural system approach is superior to other people.

Cai et al. [19] 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.

Ho et al. [20] 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.

Tian and Noore [21] proposed an on-line versatile software reliability expectation demonstrate utilizing a developmental connectionist approach dependent on various postponed input single-yield engineering, which indicated better execution as for subsequent stage consistency contrasted with existing NN display.

Tian and Moore [22] introduced a transformative neural system based technique for software reliability expectation, utilized various deferred input single yield design. The outcome demonstrated that neural system engineering greatly affects the execution of the system.

Yu Shen Su et al. [23] purposed a model that utilizes the neural system way to deal with construct a dynamic weighted combinational model. At that point thought about the exhibitions of the neural system models with some ordinary SRGMs from three angles: integrity of fit, expectation capacity for momentary forecast and long haul forecast. The outcome demonstrates that the purposed model has more precision with the two goodnesses of fit and the expectation capacity contrasted with existing ordinary models.

Viswanath [24] proposed two models, for example, neural system based exponential encoding and neural system based logarithmic encoding for expectation of an aggregate number of disappointments in software. It requires execution investment as the info and connected on four informational indexes. The outcome demonstrated that its outcome is superior to another factual model.

Sandeep Kumar Jain et al. [34] proposed a technique to gauge the reliability of the software comprising of segments by utilizing distinctive neural system structures. At that point gauge the flaws expectation conduct in the arrangement of parts over a total execution time interim other than this the forecast of deficiencies is assessed for the total software. To foresee the shortcomings in every segment of the software with the forecast of flaws for the total software for given total execution time, apply the feed forward neural system structures and its speculation ability.

b. Fuzzy System

Fluffy Logic is gotten from fluffy set hypothesis managing the thinking that is proper instead of correctly found from established predicate rationale. A fluffy model is a mapping between semantic terms, connected to factors. In this way the contribution to a yield from a fluffy model can be either numerical or semantic [35].

Cai et al. [36] talked about the advancement of fluffy software reliability models instead of probabilistic software reliability models (PSMs). It depended on the verification that software reliability is fluffy in nature. An exhibition of how to build up a fluffy model to describe software reliability was additionally displayed.

Kirti Tyagi et al. [41] proposed a model for assessing CBSS reliability, known as a versatile neuro-fluffy surmising framework (ANFIS) that depends on these two fundamental components of soft computing, neural system, and fluffy rationale. ANFIS show gives a more precise proportion of reliability than the FIS display, as it lessens blunder from 11.74%, on account of the FIS demonstrate, to 6.66% in ANFIS.

c. Genetic Algorithm (GA)

A genetic calculation is a model of machine learning which derives its behavior from a metaphor of the procedure of advancement in nature. This is finished by the creation inside a machine of a populace of people spoken to by chromosomes.

The wellness of every chromosome is controlled by assessing it against a goal work. To mimic the characteristic survival of the fittest procedure, best chromosomes trade data to create posterity chromosomes. The posterity arrangements are then assessed and used to develop the populace on the off chance that they give preferable arrangements over frail populaces individuals. For the most part, the procedure is proceeded for an expansive number of ages to get a best-fit arrangement.

Satya Prasad R. et al. [46] join both defective troubleshooting and change-point issue into the software reliability development demonstrate (SRGM) in light of the notable exponential appropriation the parameter estimation is

contemplated. The proposed model is evaluated as superior to anything the other considered models as for every one of the conditions are picked.

d. Genetic Programming (GP)

Genetic programming can be seen as an expansion of the genetic calculation, a model for testing and choosing the best decision among a lot of results, each spoken to by a string. Genetic bit goes a stage more distant and makes the program or "capacity" the unit that is tried. Two methodologies are utilized to choose the effective program cross-rearing and the competition or rivalry approach. A troublesome piece of utilizing genetic programming is deciding the wellness work, how much a program is touching base at the ideal objective.

Eduardo Oliveira Costa et al. [51] presented another GP based methodology, named (μ+λ)GP. This calculation was acquainted with enhance the execution of GP. To assess this purposed calculation, two sorts of models: in light of time and on inclusion were displayed for test results, which is in every case superior to established GP.

Zainab Al-Rahamneh et al. [52] proposed the utilization of Genetic Programming (GP) as a developmental calculation way to deal with handle the software reliability modeling issue. Assess the GP built up a model and results demonstrate this purposed model is better than different models, for example, Yamada S-formed, Generalized Poisson, NHHP and Schneidewind reliability models.

e. Artificial Bee Colony (ABC)

Dervis Karaboga, in 2005, characterized another calculation, persuaded by the canny conduct of bumble bees known as fake honey bee state. It is an enhanced device gives a populace based search strategy in which people called nourishments positions are changed by the counterfeit honey bees with time and honey bee's expect to find the spots of sustenance sources with high nectar sum and at long last the one with the most elevated nectar.

Tarun Kumar Sharma et al. [53] proposed a changed form of the ABC, the DABC (Dichotomous ABC), to enhance its execution, as far as uniting to individual ideal point and to remunerate the restricted measure of search moves of unique ABC. Additionally investigated the relevance of the adjusted counterfeit honey bee settlement calculation to assess the parameters of software reliability development models (SRGM). The assessed model parameters were utilized to anticipate the deficiencies in a software framework amid the testing procedure.

f. Ant Colony

Subterranean insect Colony Optimization [54] is a system which utilizes likelihood to tackle issues where the calculations are diminished with the assistance of diagrams to get proficient ways. It has been connected to numerous fields as its vigor and is anything but difficult to work together with different techniques. It has a decent execution to the advancement issue and has a decent combination rate.

Latha Shanmugam et al. [57] considered upgrade and Comparison of Ant Colony Optimization Methods for Software Reliability Models. The Enhanced technique demonstrates huge favorable circumstances in finding the decency of fit for software reliability models, for example, limited and boundless disappointment Poisson model and binomial models.

g. Simulated Annealing (SA) Algorithm

Mimicked toughening (SA) is an iterative search technique enlivened by the strengthening of metals [58, 59] Starting with an underlying arrangement and equipped with sufficient bother and assessment works, the calculation plays out a stochastic fractional search of the state space Nidhi Gupta et al. [60] the mimicked strengthening system of mean field estimate for finding the conceivable least number of fizzled parts in the successive testing. These base quantities of fizzled segments are relying on the choice of time interims or spaces. Likewise purposed another

vitality work with the mean field estimate. The calculation of the entire procedure demonstrates that this methodology can produce the ideal arrangement.

Pai and Hong [61] connected help vector machines (SVMs) for estimating software reliability where the mimicked tempering (SA) calculation was utilized to choose the parameters of the SVM show. The exploratory outcomes uncover that the SVM demonstrate with recreated toughening calculations (SVMSA) results in preferred forecasts over alternate techniques.

Mohamed Benaddy at al. [62] displayed a half breed approach dependent on the Neural Networks and Simulated Annealing. A versatile recreated Annealing calculation is utilized to enhance the mean square of the blunder created via preparing the neural system, anticipating software aggregate disappointment. The purposed versatile Simulated Annealing gives preferred execution in execution time over the Real-Coded Genetic Algorithm (RCGA), as a result of the search space, which decreased from a populace of answers for the RCGA to one answer for the proposed Simulated Annealing.

h. Tabu Search Algorithm

The Tabu Search (TS) is a streamlining technique, in light of the start that critical thinking, so as to qualify as smart, and should join versatile memory and responsive investigation [63]. The Tabu strategy was incompletely roused by the perception that human conduct seems to work with a random component that prompts conflicting conduct given comparable conditions.

M.Caserta et al. [64] exhibited another meta-a heuristic-based calculation for complex reliability issues. The calculation adequately utilizes highlights of the Tabu Search worldview, with extraordinary accentuation on the abuse of memory-based systems. It offsets escalation with expansion by means of the utilization of present moment and long haul memory. The proposed calculation turns out to be powerful as for its parameters and it is particularly suited for substantial scale cases of the reliability issue when correct methodologies are destined to fall flat.

i. Cuckoo Search Algorithm

This calculation depends on the commit brood parasitic conduct of some cuckoo species in mix with the L'evy flight conduct of a few winged creatures and natural product flies. The cuckoo search calculation is exceptionally effective in discovering great and worthy answers for the issue of parameter estimation of Software Reliability Growth Models. This calculation search system can proficiently explore all through the search space of the issue and find great arrangements utilizing less cycles and littler populaces.

Najla Akram AL-Saati et al. [65] assessed parameters dependent on the accessible disappointment information. Cuckoo Search beat both PSO and ACO in discovering better parameters tried utilizing indistinguishable datasets, however more awful in the event of expanded ACO. the Exponential, Power, S-Shaped, and M-O models are considered in this work. The search system of the cuckoo can effectively explore all through the search space of the issue and find great arrangements utilizing less cycles and littler populaces.

COMPARISON OF DIFFERENT SOFT COMPUTING TECHNIQUES IN TERMS OF MODELING CAPABILITIES

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 No. 1 we thought about various soft computing techniques as far as software reliability modeling capacities, for example, informational collections, re-alterations for the new informational collection, process perceivability, realities, and yields and so forth.

Table 1. Comparison of Soft Computing Techniques in terms of Modeling Capabilities

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 Networks	No	No	No	No	Yes	Partially
2	Fuzzy Logic	Yes	Yes	Yes	Yes	Yes	Yes
3	Genetic Algorithms	Partially	Partially	Yes	Yes	Partially	No
4	Genetic Programming	Yes	No	No	No	Yes	No
5	Artificial Bee Colony	Yes	Partially	Partially	No	Yes	Yes
6	Ant Colony	Yes	No	Yes	Yes	Yes	Yes
7	Stimulted Anealing	Yes	No	Yes	Yes	Yes	Yes
8	Tabu Search	No	Partially	Yes	Yes	Yes	Yes
9	Cuckoo Search	Yes	No	Yes	Yes	Yes	Yes

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. An insect province, Stimulated Annealing, Tabu Search and Cuckoo Search can be utilized for the vast majority of the modeling capacities aside from just little informational collection abilities. The fast development of soft computing techniques proposes that the effect of these calculations will be utilized progressively for software reliability models in the coming years. This table will help PC researcher who is quick to contribute their attempts to the field of software reliability.

CONCLUSION

In this paper, we have examined the work done by different researchers, with the undertaking made to incorporate however many references as could reasonably be expected from the year 1990 to 2014. In view of this paper, 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. Further, we thought about soft computing techniques as far as modeling abilities, that improves the determination procedure of soft computing method for software reliability models.

REFERENCES

[1] K.K. Aggarwal, Yogesh Singh, “Software Engineering”, New Age International publisher, New Delhi, 2005.

[2] Kapur, P.K. and Garg, R.B., "Cost reliability optimum release policies for a software system with testing effort", Operations Research, Vol. 27, No. 2, pp. 109-116, 1990.
[3] Musa J.D., "Software Reliability Engineering: More Reliable Software, Faster Development and Testing", McGraw-Hill, 1999.
[4] Musa, J.D., Iannino, A. and Komodo, K., "Software Reliability: Measurement, Prediction and Application", McGraw-Hill, 1987.
[5] P. Bonissone, Y-T Chen, K. Goebel, & P. Khedkar, "Hybrid Soft Computing Systems: Industrial and Commercial Applications", Proceedings of the IEEE, 87(9): 1641-1667, September 1999.
[6] Santosh Kumar Das, Abhishek Kumar, Bappaditya Das and A.P.Burnwal, "On Soft computing Techniques in Various Areas", ACER, pp. 59–68, CS & IT-CSCP, 2013.
[7] L. A. Zadeh, "Fuzzy Algorithms", Information and Control , Vol. 12, pp. 94-102, 1968.
[8] L. A. Zadeh, "Fuzzy Sets", Information and Control, Vol. 8, pp. 338-353, 1965.
[9] Kuo, W. & Prasad, V. R., "An annotated Overview of System Reliability Optimization", IEEE Transaction on Reliability 49(2):176-186, 2000.
[10] FelixStreichert,"IntroductiontoEvolutionary Algorithms", Frankfurt MathFinance Workshop, university of Tubingen. Germany, April 2-4, 2002.
[11] Ajith Abraham, "Evolutionary Computation", Handbook of Measuring System Design, edited by Peter H. Sydenham and Richard Thorn. John Wiley & Sons, Ltd. ISBN: 0-470-02143-8. 2005.
[12] S. Boccaletti et al., "The control of chaos: theory and applications", Elsevier Science, 2000.
[13] Gaurav Aggarwal and Dr. V.K Gupta, "Neural NetworkApproach to Measure Reliability of Software Modules: AReview", in International Journal of Advances inEngineering Sciences Vol.3, Issue 2, April, 2013.
[14] 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.
[15] N. Karunanithi, D. Whitley, Y.K. Malaiya, "Using neural networks in reliability prediction", IEEE Software, Vol. 9, no. 4, pp.53–59, 1992.
[16] 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.
[17] T. M. Khoshgoftaar, R. M. Szabo, and P. J. Guasti,"Exploring the Behavior of Neural-network Software Quality Models," Software Eng. J., Vol. 10, No. 3, pp. 89–96, May 1995.
[18] R. Sitte, "Comparison of Software Reliability Growth Predictions: Neural networks vs. parametric recalibration", IEEE transactions on Reliability, pp. 285-291, 1999.
[19] 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.
[20] 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.
[21] L. Tian, A. Noore, "On-line prediction of software reliability using an evolutionary connectionist model", The Journal of Systems and Software, Vol. 77, no. 2, pp. 173–180, 2005.
[22] L. Tian, A. Noore, "Evolutionary neural network modeling for software cumulative failure time prediction", Reliability Engineering and System Safety, Vol. 87, no. 1, pp.45–51, 2005.
[23] Yu Shen Su, Chin-Yu Huang, Yi Shin Chen and Jing Xun Chen, "An Artificial Neural-Network-Based Approach to Software Reliability Assessment", TENCON, IEEE Region 10, pp-1-6, 2005.
[24] S.P.K. Viswanath, "Software Reliability Prediction using Neural Networks", PhD. Thesis, Indian Institute of Technology Kharagpur, 2006.
[25] Q.P. Hu, Y.S. Dai, M. Xie, S.H. Ng, "Early software reliability prediction with extended ANN model", Proceedings of the 30th Annual International Computer Software and Applications Conference, pp. 234-239, 2006.
[26] Su, Y.-S., Huang, C.-Y., "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models", Journal of Systems and Software 80 (4), 606–615, 2006.

[27] Kanmani, S., Uthariaraj, V.R., Sankaranarayanan, V., Thambidurai, P., "Object-oriented software failure fault prediction using neural networks", Information and Software Technology, Vol. 49, 483–492, 2007.
[28] Sultan H. Aljahdali and Khalid A. Buragga, "Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study", in ICGST-AIML Journal, ISSN: 1687-4846, Vol. 8, Issue II, 2008.
[29] Y. Singh, P. Kumar, "Application of feed-forward networks for software reliability prediction", ACM SIGSOFT Software Engineering Notes, Vol. 35, no. 5, pp. 1-6, 2010.
[30] Y. Singh, P. Kumar, "Redirection of Software Reliability Using feed Forward Neural Networks", International conference on Computational Intelligence and software Engineering, pp. 1-5, 2010.
[31] Nirvikar Katiyar and Raghuraj Singh, "Effect of Neural Network for Prediction of Software Reliability", in VSRD-IJCSIT, Vol. 1 (7), pp 490-500, 2011.
[32] C.Y. Huang, M.R. Lyu, "Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models", IEEE Transaction on Reliability, Vol. 60, no. 2, pp. 498-514, 2011.
[33] Manjubala Bisi and Neeraj Kumar Goyal, " SoftwareReliability Prediction using Neural Network withEncoded Input", International Journal of Computer Applications (0975 – 8887) Vol. 47– No.22, 2012.
[34] Sandeep Kumar Jain and Manu Pratap Singh, "Estimation for Faults Prediction from Component Based Software Design using Feed Forward Neural Networks", IJARCCE, Vol. 2, Issue 7, 2013.
[35] Sultan Aljahdali and Narayan C. Debnath, " ImprovedSoftware Reliability Prediction through Fuzzy LogicModeling", IASSE , 17- 21, 2004.
[36] Cai, K.Y., Wen, C.Y., Zhang, M.L., "A critical review on software reliability modeling", Reliability Engineering and System Safety 32 (3), 357–371, 1991.
[37] Khalaf Khatatneh and Thaer Mustafa, "Software Reliability Modeling Using Soft Computing Technique", in European Journal of Scientific Research ISSN 1450-216X Vol.26 No.1, pp.147-152, 2009.
[38] Reformat, M., "A fuzzy-based multimodel system for reasoning about the number of software defects",International Journal of Intelligent Systems 20 (11), 1093–1115, 2005.
[39] Sultan Aljahdali, "Development of Software Reliability Growth Models for Industrial Applications Using Fuzzy Logic", Journal of Computer Science 7 (10): 1574-1580, 2011.
[40] S. Chatterjee, S. Nigam, J.B. Singh, L.N. Upadhyaya,"Application of Fuzzy Time Series in Prediction of TimeBetween Failures & Faults in Software ReliabilityAssessment", in Fuzzy Inf. Eng. 3: 293-309, 2011.
[41] Kirti Tyagi and Arun Sharma, "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems", Applied Computing and Informatics,2014.
[42] L. Tian and A. Noore, "Evolutionary neural network modeling for software cumulative failure time prediction",Reliability Engineering & System Safety, Vol. 87, no. 1, pp. 45 – 51, 2005.
[43] L. Tian and A. Noore., "On-line prediction of software reliability using an evolutionary connectionist model",Journal of Systems and Software, Vol. 77, no. 2, pp. 173– 180, 2005.
[44] Sultan H. Aljahdali and Mohammed E. El-Telbany, "Genetic Algorithms for Optimizing Ensemble of Models in Software Reliability Prediction", in ICGST-AIML Journal, Vol. 8, Issue I, June 2008.
[45] Sultan H. Aljahdali and Mohammed E. El- telbany, "Software Reliability Prediction Using Multi-Objective Genetic Algorithm", 978-1-4244-3806-8/09/$25.00, IEEE, 2009.
[46] R.SatyaPrasad, O.NagaRaju and R.R.LKantam, "SRGM with Imperfect Debugging by Genetic Algorithms", in International Journal of software engineering & applications (IJSEA), Vol. 11, No. 2, April 2010.
[47] E. Costa, S. Vergilio, A. Pozo, and G. Souza. Modeling software reliability growth with genetic programming. InISSRE '05: Proceedings of the 16th IEEE InternationalSymposium on Software Reliability Engineering, Washington, USA, IEEE Computer Society, 2005.
[48] E. Oliveira, A. Pozo, and S. Vergilio, "Using boosting techniques to improve software reliability models based on genetic programming", in ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, Washington, USA, IEEE Computer Society, 2006.
[49] Y. Zhang and H. Chen. 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, IEEE Computer Society, 2006.

[50] Wasif Afzal and Richard Torkar, "Suitability of genetic programming for software reliability growth modeling",International Symposium on Computer Science and its Applications, 978-0-7695-3428-2/08 $25.00, IEEE, 2008.

[51] Eduardo Oliveira Costa, Aurora Trinidad Ramirez Pozo, and Silvia Regina Vergilio, "A Genetic Programming Approach for Software Reliability Modeling", IEEETransactions on Reliability, Vol. 59, NO. 1, March 2010.

[52] Zainab Al-Rahamneh, Mohammad Reyalat, Alaa F. Sheta,Sulieman Bani-Ahmad, Saleh Al-Oqeili, " A NewSoftware Reliability Growth Model:Genetic-Programming-Based approach", Journal of Software Engineering and Applications, 4, pp. 476-481, 2011.

[53] Tarun Kumar Sharma, Millie Pant and Ajith Abraham, "Dichotomous Search in ABC and its Application in Parameter Estimation of Software Reliability Growth Models", 978-1-4577-1124-4/11/$26.00_c IEEE, 2011.

[54] Li, W., Q. Yin and X. Zhang, "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, Sep. 23-26, IEEE Xplroe Press, Changsha, pp: 313-317. DOI: 10.1109/BICTA. .5645311, 2010.

[55] Changyou Zhenga, Xiaoming Liua, Song Huanga and YiYaoa, "A Parameter Estimation Method for Software Reliability Models", in Elsevier Ltd. Selection, 2011.

[56] Latha Shanmugam1 and Dr. Lilly Florence, "A Comparison of Parameter best Estimation Method for software reliability models", International Journal of Software Engineering & Applications (IJSEA), Vol. 3, No.5, 2012.

[57] Latha Shanmugam and Lilly Florence, "Enhancement and comparison of ant colony optimization for software reliability models", Journal of Computer Science 9 (9): pp. 1232-1240, 2013.

[58] Kirkpatrick Jr., S., Gelatt, C., Vecchi, M. "Optimization by simulated annealing", Science, 220 (4598), 498–516, 1983.

[59] Cerny, V., "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm", Journal of Optimization Theory and Application, 45 (1), pp. 41–51, 1985.

[60] Nidhi Gupta and Manu Pratap Singh, "Evolutionary algorithms, simulated annealing and tabu search: a comparative study", IJE Transactions B: Applications, Vol. 19, No. 1, 2006.

[61] Pai, P.F., Hong, W.C., "Software reliability forecasting by support vector machines with simulated vector machines with simulated annealing algorithms", The Journal of Systems and Software 79, 747–755, 2006.

[62] Mohamed Benaddy and Mohamed Wakrim, "Simulated Annealing Neural Network for Software Failure Prediction",InternationalJournalofSoftware Engineering and Its Applications, Vol. 6, No. 4, 2012.

[63] Glover, F., Laguna, M. "Tabu Search", Kluwer Academic Publishers, Boston, 1997.

[64] M. Caserta and A. Márquez Uribe, "Tabu search-based meta-heuristic algorithm for software system reliability problems", 0305-0548/$, Elsevier Ltd, 2006.

[65] Najla Akram AL-Saati and Marwa Abd-AlKareem, "The Use of Cuckoo Search in Estimating the Parameters of Software Reliability Growth Models", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 6, 2013.

[66] Sona Ahuja, Guru Saran Mishra and Agam Prasad Tyagi, "Jelinski – Moranda Model for Software Reliability Prediction and its G. A. based Optimised Simulation Trajectory", D. E. I. Dayalbagh, Agra, pp. 399-404, 2002.