A STUDY OF VARIOUS SOFTWARE RELIABILITY SYSTEMS BY USING ANN

VidushiAwasthi[a], Shiv Kumar Sharma[b],
Department of Mathematics, Chandigarh University, Mohali, Punjab- India.
E-mail: [a]vidushiawasthi@gmail.com, [b]shivksharma2005@gmail.com

ABSTRACT:

One of the quantifiable credits of software quality is reliability.Programmable/ Software Reliability Growth Model (SRGM) canbe used for continuousquality during difficult times. In all conditions where test workfluctuatesover time, the customary time sensitive SRGM may not be clear enough. In order toclose this gap,testingwork was used insteadof time in SRGM. It may be unwise to put forwarda restricted test pressure limit in advance, because the testworkwill be endlesswithinthe incomprehensible test time.

Later in this article, we propose a permanent test stressservice related to the old inhomogeneous Poisson processmodel (NHPP). We use an artificialneuralnetwork (ANN) to configure the proposed model,whichcontains frustration data from the software.Here, it is reasonable to obtain a huge load of gameplansforthe comparison model, which represents past disappointment data in a comparable way. We use artificial intelligence methods to select game planswith reasonable load for the model to describe the past and future data well. We use a reasonable softwaredisappointmentdata set to decompose thepresentation of the proposed model from the current model. Use the artificial neural network method to design theGeneral Direct Software Reliability Growth Model (SRGM) through test work.: Thetrue quality software shown by current research mainlyfocuseson the best methodof generalguessingmodeling.

KEYWORDS: Artificial Neural Network, Back Propagation, Software Reliability Growth Model, Software Testing, Testing Effort Estimation, Non-Homogeneous Poisson Process (NHPP), Software Testing, Pattern mapping technology, basic model running time.

1. INTRODUCTION

The American National Standards Institute (ANSI) describes scheduling stability as the probability that the project system will move without dissatisfaction within a predetermined amount of time in a predetermined environment 1. In the test phase, scheduling quality progress improves by changing frustrating issues. The schedule credibility enhancement model verifies the quality of current fidelity by evaluating the limits of the model and uses actual disappointment data to predict the future stability of the structure. The SRGM display execution depends on the possibility of heuristic fault classification. But since 1972, more than 200 SRGMs have been proposed; however, the assessment will develop more detailed and insightful models.

Generally, there are two plans of game for stable software standard enhancement models: parametric models and non-parametric models. The non-homogeneous Poisson process (NHPP), the Markov model, and the Bayes model are three parameterized models. The NHPP model is useful in the areas of durable quality gear and programming. Thus, experts often use the NHPP model to plan

unshakable quality planning studied2. In3 the main rooms of the NHPP model with respect to constant dispersion. At that time, 4 NHPP models with different types of S-shape, 2 proposed a logarithmic Poisson runtime model and a basic runtime model. 5 and 6 proposed imperfect research models. Subsequently, 7 proposed the generalized NHPP model, 8 showed the generalized Weibull NHPP model. In10 reviewed NHPP-based SRGM to improve program quality. In11 discussed a quality-advanced model for NHPP programming with imperfect scans. The In12 survey SRGM uses recorded project disappointment data to improve its forward-looking accuracy.

The standard parameter SRGM has two credits for everything that really matters. First, the model's job is to expect the error cycle to follow a predetermined propagation. Second, restrictions, such as the dependent and self-sufficient components in the parameter SRGM, have broad explanations, for example, the overall disappointment number or the unsatisfactory area rate. In any case, the restrictions of the model are influential, that is, the restrictions will vary with the additional data that is allowed to be opened.

Of course, the non-parametric programming constant quality model based on artificial neural network (ANN) can check the model boundary without hesitation. It is used to reverse research and collect questions by deviating from common neural associations. The number of hidden layers of the neural network can be expanded by expanding the multi-layer feedback to improve the accuracy of the ANN model evaluation. Compared with ordinary parameter models, the immovable quality model based on ANN programming has better limit evaluation and perceptual accuracy15,16. In16 came up with an important ANN-based estimation idea of the hard-hitting quality of programming. In17 proposed an evaluation between neural correlation and parametric models to expect reliability. In18 applied backpropagation neural correlation to programming credibility to predict accompanying frustration time. In19 proposed a neural association method that combines dynamic development of weighted models to program assumptions and evaluations of stable quality.

In20 distinguished between a pseudo-neuronal association and three standards verifiable NHPP models in anticipation of programming stability. In21 applied neuro-associated thinking to the standard programming reliability breakthrough model to improve programming accuracy through ruthless quality guessing. In 22suggested theexploration on reliability theory using lgcm model inneural network. In23 proposed a neurogenesis technology based on the essential model to improve the accuracy of project quality solving conjectures. In24 audited the precise strategies in decision-making. Backpropagation and hippocampus estimation for quantifiable game plans are discussed in 25. It collects relentless quality scheduling models to improve scheduling stability in 26.

In the testing phase of the project, attempts such as work, experimentation, and computational time were consumed. The deficiencies and corrections will depend on the responsibilities assumed. The exponent, Rayleigh, logistic, and Weibull limits have been used extensively to describe the cost of trial work 27-31.

The following area describes the process of using neuron association fragments to further design the proposed model. Part of the idea of testing work is introduced. The other part introduces the execution of programming and faithfully evaluates the quality of the proposed model. It also

discusses the evaluation results through the use of heuristic classification of common-sense disillusionment by calculating the existence of standard basic models, neural correlation-based models, and proposed models.

2. NEURAL NETWORK SEGMENT:

Neural tissue is an organization of interconnected non-linear neurons that are concluded through examination of the public nervous system. The limitation of neural tissue is to communicate performance plans when given data patterns34. The proposed ANN-based SRGM and testing work is constructed using three main parts: neuron, network design, and learning estimation. • Neuron: Neuron is a huge part of the collaboration of neural tissues. It guides the data segment and performance data segment by prompting work. The interaction of isolated neurons is numerically characterized as

$$x = f(j) and\ j = \sum_{i=1}^{K} U_i t_i$$

Here,

f(j) = performance work, which maps to Information about the latest performance information component.

Σ = the addition of the information components of the information collected using the relevant input load.

$u_i = u_1, u_2, u_3, \ldots u_k$ is the corresponding weight of the input data element K is the total number of input data elements, namely $t_1, t_2, t_3, \ldots t_k$and

x = a final output single nerve yuan.

• Neural network architecture: An important multilayer feedforward neural network is used to design the proposed model. The computation of the data here progresses, starting from the data layer to carry the information segment, to the performance layer, through a mysterious layer to design the data and the performance segment to convey the performance. The proposed ANN-based SRGM uses a single layer of data and output along with the test work, each layer has a single neuron, and the three mystery layers each have a single mystery neuron.

• Readiness Estimation: The proposed ANN-based SRGM finds the difference between the estimated output and the actual output by processing the neural correlation with the input data preparation and changing the load, and establishes it by backpropagation readiness calculations until the final model is satisfied. This performance bottom improves the evaluation performance of the model design by extending the correlation plan on the regression path through the trend stack of the smooth neural correlation.

3. TEST EFFORT MODELING:

In28 proposes a test work-dependent programming reliability development model that is based on the Goel-Okumoto model. In 35 the estimation of complexity in software reliability growth

modeling discussed for complexity of software reliability. Assuming that M (y) is a tentative effort invested in energy x, the mean value capacity μ (x) of SRGM is

$$\mu(y) = n\left[1 - e^{-rm(y)}\right] \quad 0 < p < 1; \quad d > 0$$

where

r = the defect recognition rate deciphered according to the intensity of the test per unit of time y. and M (y) = The measurement standard for an integral inversion test in the energy interval [0, y].

In29 proposed the Weibull curve as a test pressure work for excellent SRGM. The absolute stress of the combustion test in the time interval [0, y] is

$$M(y) = i\left(1 - e^{jx^k}\right)$$

where `i` is the complete stressto be burned.Ifitproves to continueindefinitely, `j` is the scale limit and the cis-form limit. At k = 1, we get excellent trial work. We chose thelog-power32 model becauseof its characteristic properties.Weconverted the model in time2 toconvert it to a stress-based model.

4.EXPERIMENTAL IMPLEMENTATION:

4.1 Running the proposed ANN-based SRGM andtestwork:

The method of running initially dependson the design of ANN-based SRGM through test work asfollows:

1. Select the basic programming constantenhancement model and test pressure function as the help model to organizeNeural association plan,and has the proper authorization capabilitiesrelated to performance and hidden layer neurons.

2. Using the presetneuronassociations, check the single neuron stack and measure the suggested ANN-based SRGM visualization exam throughtrial work.

4.2 Basic model selection:

The ANN-based SRGM with stress testing proposed in this paper has a single neuron in the information and performance layer, and one neuron in each of the three secret layers. The measurement of neurons in the overlay is found by the selected support model and the selected stimulus limit to plan the ANN-based programming reliability model. Among the proposed models, a very disappointing logging power model 32 was selected and an impressive test effort was performed because of their simplicity and ability to provide the best fit.

Tends to show ANN-based SRGM and uses reverse causal scheduling estimate for test work. In the standard logarithmic power SRGM, the time "y" is replaced by a special test job applying the time shift applicable to the NHPP model. If M (y) is the pending test consumption invested in energy t, then the mean value capacity μ (x) of the SRGM32 logarithmic power is given as follows:

$$\mu(y) = i\ log\left(1 + M(y)\right)^k$$

and

$$M(y) = V(1 - e^{-jy})$$

where, i, j and k are constants.

M (y) = a measure of the test consumption inverted in the energy interval [0, y].

Thus, along these paths, the mean value of the μ (y) capacity of the logarithmic power SRGM with excellent test work is as follows:

$$\mu(y) = i\ log\left(1 + V(1 - e^{-jy})\right)^{k}$$

The performance and test workload of the proposed SRGM based on the multi-faceted feedback ANN are based on:

$$z(y) = u4\log\left(1 + u2(1 - e^{-uly})\right)^{u3}$$

where u1, u2, u3 and u4 (> 0) are the loads of the proposed programming reliability model based on artificial neural networks, and the weight estimation is prepared by calculating recoil.

As shown in the prerequisites for the start limit in the ANN-based scheduling steadfast quality improvement model, the trigger limit is set to be basic, constant, and sufficiently differentiable, and the output limit can be set as a compound design g (f (y)). The result of the proposed anticipated ANN-based multilayer SRGM with test workload constraints can be obtained from its mean limit as a composite design, as shown below: Equation (10) is the mean capacity of the proposed ANN based on in the SRGM test job.

The expectation is:

$$f(y) = 1 - e^{-hy},\quad g(y) = Ny,\quad q(y) = \log(1 + y)^{k}\quad and\quad p(y) = iy$$

and then we get,

$$p\left(q\left(g(f(y))\right)\right) = p\left(q(g(1 - e^{-hy}))\right) = p\left(q(N(1 - e^{-hy}))\right)$$

$$= p\left(\log(1 + N(1 - e^{-hy}))\right)^{k}$$

$$= i\left(log(1 + N(1 - e^{-hy}))\right)^{k}$$

From equations (6) and (8), we see that the mean value limit and the proposed ANN-based SRGM test work are defined by (y), g (y), q (y) and p (y).

From the above deduction, we see that the mystery layer neuron has an initial boundary, y, log (1 + y), and the direct sanction work is used in the performance layer neuron of the proposed ANN p (y) = y - SRGM-based test job. Note that the activation limit is infinite and differentiable everywhere. The starting limits for hidden layer neurons are sorted from the different average limits of the selected SRGM and the test job. If we encourage neural associations with initial constraints (such as y and log (1 + y)) in the mysterious layer e and the performance layer with heaps u1, u2, u3, and u4, then we will get layer hiding. And the performance layer is as follows.

$$a(y) = 1 - e^{-uly}$$

$$b(y) = u2(1 - e^{-uly})$$

$$c(x) = \log\log\left(1 + u2(1 - e^{-uly})\right)^{u3}$$

and

$$x(y) = u4(\log\left(1 + u2(1 - e^{-uly})\right)^{u3})$$

Consider equations (6) and (12), which are SRGM logarithmic powers with amazing test work, and u1 = j, u2 = V, u3 = k, and u4 = i.

By developing a comparison technique, we can fundamentally cultivate neural associations with appropriate stimulation limits and individually charge some irregular SRGMs.

4.3 Standardization of failed data:

Unsatisfactory project data is scheduled in pairs {xi, yi}, where xi is the total test time and yi is the number of failed merges for comparison. Before processing the ANN, the failing educational record should be standardized in the range of [0, 1] for its most significant characteristics.

4.4 Backpropagation Training Algorithm:

The proposed SRGM based on ANN and test work is established through reverse-induced scheduling calculations, with a standardized actual scheduling disappointment-guided classification, and a large number of neural associations to reduce errors, a large number of associations are presented irregularly. The iterative difference in the associated load continues until the failure is not actually a predefined barrier limit.

5. RESEARCH EVALUATION RESULTS:

5.1 Failure data set:

Two utility frustration data sets 33 Musa P1 and Musa P14c are familiar with the evaluation and testing of the proposed SRGM based on artificial neural networks. Here, these two educational lists are implied as DS-1 and DS-2 respectively. Data set 34 is as follows:

• DS-1: 136 failures, 21,700 lines of code are used to stabilize the control structure, which is accumulated by Musa at Telephone Laboratories.

• DS-2: Of the 180,000 lines of code extracted from Moussa's tactical application project, 101 are unsatisfactory.

A large number of experts use DS-1 to stress the display of their model, because the communication of the test is balanced and does not show the miracle of learning or changes in the data. Surprisingly, DS-2 was a challenge for SRGM because the widespread instability of the data was similar to the test group's miracle of learning. The final message appears to have shown direct anger.

6.2 Model evaluation criteria:

In order to consider the introduction of the relentless quality programming model, the researchers proposed some measures to evaluate the model. We measure the existence of the proposed model using R2 (confirmation coefficient), RMSE (mean square error) and AIC (Akaike information criterion). The value of R2 can go from 0 to 1. The closer R2 is to 1; the better it fits, the better. RMSE is a model that measures the difference between actual characteristics and expected characteristics. AIC is the respectability of the attack of the normal quantifiable model. AIC is considered an activity that can be used to classify models and provides discipline for models with more constraints. RMSE and AIC are prepared as follows:

$$\mathrm{RMSE} = \frac{\sqrt{1}}{\mathrm{k}} \sum (\mathrm{F} - \mathrm{O})^2$$

where k is the absolute focus number taken by the test run, F is the test error of the evaluation, and O is the test error that is actually noticed. The smallest RMSE meets the best execution.

$$\mathrm{AIC} = \mathrm{k} \log \left(\frac{\mathrm{RSS}}{\mathrm{K}} \right) + 2\mathrm{U}$$

Where k represents the mean value of the number, the RSS address represents the general grid, and U is the individual load in tissue engineering. The smallest AIC shows the best decency for organizational design.

6.3 Performance Analysis:

We used the ANN-based system to evaluate the demonstration test of the proposed model, and combined and uncombined test work to prove that the past was better.

The final evaluation results of R2, RMSE and AIC are shown in Table 1. As expected, the effort-based model seems to provide the best performance among the typical fine time models. In addition, it is clear that the use of neural correlation to evaluate the limits further enhances the display effect. DS-2 classification teaching has widespread instability, because when it is separated from DS-1, the learning miracle of the test group, as all fairness records show. The Standard Model is vulnerable to wild instability in the data. However, whenever there is a learning miracle and there is reasonable instability in the data, the proposed ANN-based SRGM with test workload limitations will perform better. AIC is considered to provide a general display record of the model.

6. CONCLUSION:

Numerous time-sensitive SRGMs have been proposed. These SRGMs expect to receive a consistent commitment throughout the test period. In any case, this is not reality. Two or three effort-based SRGMs have been proposed before. In this article, we propose a clear logarithmic power SRGM, which has excellent test work, and investigates the working limit using false neural associations. Investigating ANN and ordinary form constraints, we will break down the SRGM introduction regardless of whether we join the test work. The inspection claims that the ANN-based test work

model will describe frustrating data in any case, because the data is highly jittery and cannot be displayed.

REFERENCE:

1. Michael, R.L. (1996). Software Reliability Engineering Handbook. IEEE Computer Society Press. New York: McGraw Hill.

2. Xie, M. (1991). Software reliability modeling. Singapore: World Scientific.

3. Goel, A.L. and Okumoto, K. (1979). A time-dependent error detection rate model for measuring the reliability and other performance of software. Stability IEEE Transactions, 28(3), 206-11.

4. Yamada, S. and Osaki S. (1985). Software Reliability Growth Models: Models and Applications. IEEE transactions related to software engineering, 11 (12), 1431-37.

5. Oba, M. and Butterfly, X.M. (1989). Does it affect the reliability of incomplete debugging software? Proc 11th International Conf Software Eng, New York, 237−44.

6. Kapur, P.K. and Garg,R.B. (1992). Software reliability growth model of error elimination phenomenon. Software Engineering Journal, 7 (4),291-4.

7. Goel,A.L. (1985). Software Reliability Model: Assumptions, Limitations and Applicability. IEEE Trans on Software Engineering, 11(12), 1411-23.

8. Subburaj, R. and Gopal, G. (2006). A generalized exponential Poisson model to increase the reliability of software. International J. in Performance Engineering, 2(3), 291-301.

9. Subburaj, R., Gopal, G. andKapur, P.K. (2012). A software reliability growth model that computes debug and learning metrics. International J. in Performance Engineering, 8(5), 539-49.

10. Lai, R. and Garg, M. (2012). Detailed study of reliability model of NHPP software. Software magazine, 7 (6), 1296-306.

11. Roy, P.,Mahapatra, G.S. and Dey, K.N. (2014). A model for improving the reliability of NHPP software, which involves incomplete debugging and error generation. International Journal of Reliability, Quality and Safety Engineering, 21(2), 1-32.

12. Rana R, et al. (2014). Select a software reliability growth model and use historical project data to improve forecasting accuracy. Systems and software magazine, 98, 59-78.

13. Roy, G.S.P. and Dey, M.K.N. (2015). Professional system with applications, 42 (10), 4209-718.

14.Zhixin, J., Hong-bin, Z. (2009). Reliability modeling study of neural network-based WEDM. Proc IEEE Symp measurement technology and mechatronics automation (ICMTMA`09), Zhangjiajie, Hunan, China, 277-80.

15. Karunanithi, N., Whitley, D. andMalaiya, Y.K. (1992). Neural network is used for reliability prediction. IEEE Software, 9, 53-9.

16. Karunanithi, N., Whitley, D. andMalaiya, Y.K. (1992). Software reliability prediction using the connection attention model. IEEE Trans on Software Eng, 18 (7), 563-74.

17. Sitte, R. (1999). Software Reliability Growth Prediction Comparison: Neural Network Parameters Readjustment. Reliable IEEE transformer, 48 (3), 285-91.

18. Cai, K.Y. et al. (2001). A neural network approach to software reliability modeling. J Systems and software, 58 (1), 47-62.

19. Su, Y.S. and Huang, C.Y. (2007). A neural network-based approach for software reliability estimation using a dynamic weighted coupling model. Systems and Software J., 80(4), 606-15.

20. Zheng, J. (2009). Software reliability prediction using a neural network ensemble. Professional system with applications, 36 (2), 2116-22.

21. Wang, G.and Li. W. (2010). A study of software reliability combination model based on neural network. IEEE 2nd WRI World Conference on Software Engineering, 2, 253-6.

22. Kaur, H. and Sharma,S. K. (2020). Exploration on reliability theory using lgcm model inneural network, Advances in Mathematics: Scientific Journal, 9(8), 5349–5359.

23. Roy, P. et al. (2014). A combination model of powerful feedforward and circular neural network-based dynamic weights for software reliability prediction. Application soft computing, 22, 629-37.

24. Das, T.K.(2016). Decision-making intelligent technology: research. Indian science and technology magazine, 9(12).

25. Jabarullah, B.M. and Babu, C.N.K. (2015). BPNN-hippoamy algorithm for statistical classification Indian science and technology journal, 8(14).

26. Gayathry, G. (2015). Classification of software reliability models to improve the reliability of Selvi RT software. Indian Journal of Science and Technology, 8 (29).

27. Kapur, P.K., Grover, P.S. and Younes, S. (1994). Modeling of incomplete debugging phenomena caused by test operations. Minutes of the 5th International Symposium on Software Reliability Engineering. Monterey, California, 178-83.

28. Yu, C. (2007). Evaluation of software reliability growth models that depend on test tasks Stability IEEE Transactions, 56(2), 198-211.

29. Yamada, S. (1986). Software reliability growth model test work, Stability IEEE Transactions. 1986; 35(1): 422-24.

30. Yamada, S. (1993).Software Reliability Growth Model for Evil Testing Efforts. Stability IEEE Transactions, 42(1), 100-05.

31. Kapur, P.K.et al. Use an ensemble of artificial neural networks to improve software reliability in complex software system architectures.

32. Zhao, M. andXie, M. (1992). Log Power For computing reliability models of NHPP software. IEEE transaction for reliability, 14-22.

33. Musa, J.D. (1980). Reliability data set in DACS software. The data and analytics center of the software.

34, Indhurani, L. andSubburaj, R. (2015).An artificial neural-network approach to software reliability growthmodeling, Procedia Computer Science, 57, 695 -702.

35. Thakur P, Sharma S. K. Estimation of complexity in softwarereliability growth modeling, Advances and Applications in Mathematical Sciences2020; 19(6), 563-572.