ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.8, 5349–5359
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.8.6

EXPLORATION ON RELIABILITY THEORY USING LGCM MODEL IN NEURAL NETWORK

HARPREET KAUR[1] AND SHIV KUMAR SHARMA

ABSTRACT. Reliability of software is always very notable to both, the producer and the consumer. Reliability in itself marks the very quintessence of any new beginning. This study broadly explores the software Reliability growth models and its majorly classified groups, i.e. Concave and S-shaped models where it majorly discusses the flaws identification rate and easily let us know on how flaw identification and correction yields better Reliability of the software. It further expresses and gleans a few software Reliability growth replicas which include, Empirical Bayes techniques which is for statistical thesis in which the previous distribution is approximated from the data. Using Bayesian equation is been glean for obtaining quantifiable results. Further, it also explores artificial neural network based LGCM software Reliability and NHPP software Reliability; considering the data available in any non-linear continuous functions and unambivalent explains different layers of ANN covering the LGCM base, and in similar approach discussing the NHPP (Non Homogeneous Poisson Process) covering its suggest value feature and failure vigour feature.

1. INTRODUCTION

Reliability theory is a collection of thoughts, scientific models, and techniques coordinated to anticipate, gauge, comprehend, and enhance the life expectancy and failure distributions of schemes and their parts [8]. Reliability theory enables scientists to anticipate the age related failure energy for an arrangement of given design (Reliability structure) and given reliability theory of its segments.

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 68T07.

Key words and phrases. Bayes deduction, Empirical Bayes, Neural Network, LGCM, NHPP, Reliability software.

Reliability: the possibility that a system will continue to function unescorted by failure for a defined amount of normal units or a defined time - Precision, security, working facets of usability and user-affability - period may be in normal or time units.

Exemplar of accepted units - runs, sheets of yield, deal, phone sounds, occupations, semi conductor wafers, inquiry, API calls - Failure intensity = non success per normal or period unit

Quick development of technology, information underpinning based on telecommunications and high speed computing ability acting a vital part in many areas for instance economics, politics, culture etc. At the mean time, some system including but not restricted to operational system become complex and bring tremendous loss. So, that ways to choose the reliable software in other way, PC frameworks are constrained by programming so we need software to be reliable. Software reliability which is studied by many research organizations describes the chance of fiasco free software operation. Many models are there to guess and forecast the correlation betwixt software reliability and time.

The reliability of the software is basically two types of models: the first is the effort to anticipate the reliability of the software from the constraints of the plan and the second is the effort to anticipate the reliability of the software from the test information. The main kinds of Model is habitually called “deformity thickness" replicas furthermore exploits cryptographic qualities, for instance cryptogram lines, creation of circles, inputs etc. to assess amount of imperfections in the product. The next kinds of model are often referred to as “consistency software growth" [8]. In effect, these models try to link information about the position of the deformity with its abilities, for example an Exponential Function. When connection is excellent, the function can be accustomed to foretell future behaviour. The models of software reliability growth are at the heart of this relationship.

The beyond explanation cover all four aspects of item for consumption, distinct value, which speaks only in accordance with the specifications. In other words reliability is quality over time, which is influenced by time and the environment as opposed to quality, which is a degree of confirmation only without considering the duration of time and the operating environment [5]. An additional significant distinction among quality and reliability is that reliable systems can be manufactured by means of less reliable apparatus through varying

the creation constitution, while it is not possible to produce high quality systems with lower quality components. Adding up one or more analogous apparatus in comparable can enlarge scheme reliability [1], [3].

Reliability must have a significant impact, in the event that it is identified with time. It very well may be clarified with reliability of an engine vehicle. In the event that any vehicle fizzles, significantly in the wake of adhering to maker's guidelines indicated, the vehicle will be considered as inconsistent. In the event that it doesn't flop before its expressed life, it would be considered as solid [6], [2].

In the present situation of worldwide challenge and advancement, it is basic that Indian enterprises become completely aware of the need to create solid items satisfying global guidelines. Despite the fact that the "reliability quality Designing" has taken birth during World War II with a huge commitment by resistance faculty, today it has taken another shape by mixing itself in all periods of the item life cycle from proposition to assembling [3].

2. SRGM Models

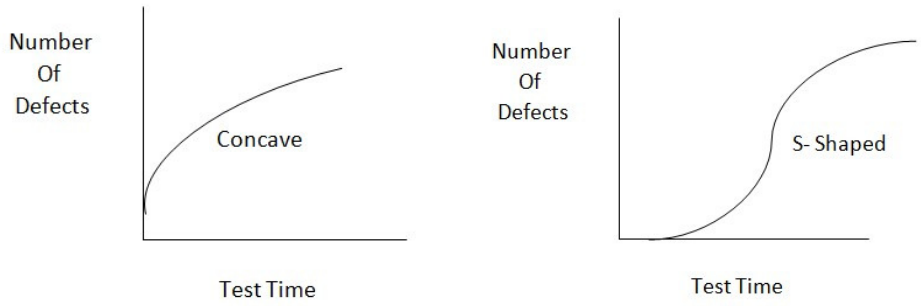
Software dependability is generally characterized by the chance that a framework will work devoid of a fiasco during a predefined period of point beneath certain working conditions. The reliability of the software is apprehensive about time among the error otherwise its equal, the error pace. In this testimony we are thinking about the information of a test domain, so we report the recognition rate of imperfections instead of the error rate. A deformity position is usually a failure in the middle of assessment; however the analysis software can furthermore identify a flaw despite of details in which the analysis continues work. Deformities may moreover be identified at some stage in configuration audits otherwise cipher surveys, although we do not think about this type of exercise in this description. Occasion in an ordeal situation is an equivalent word for measuring the test, which may be predicted in different customs. Information on the discovery of deformity includes a period for each imperfection or accumulation of deformity. Replicas of software reliability development are arithmetic alinset of fault identification data for mathematical purposes. The utilities are worn to calculate fiasco frequencies or the quantity of unused errors in the symbols.

3. Fault Count Models

This class of models is worried about demonstrating the number of disappointments seen or blames identified in given testing interims. As deficiencies are evacuated, from the framework, it is expected that the watched number of disappointments per unit time will diminish. In the event that this is along these lines, at that point the - total number of disappointments versus time bend will in the end level off. Note that time here can be calendar time, CPU time, number of experiments run or some other applicable metric. In this setup, the time interims might be settled from the earlier and the watched number of disappointments in every interim is dealt with as an irregular variable. A few models have been proposed to depict such disappointment marvels. The fundamental thought behind the greater part of these models is that of Poisson dissemination whose parameter takes diverse structures for various models. It ought to be noticed that Poisson conveyance has been observed to be a fantastic model in numerous fields of use where intrigue is in the quantity of event

4. Types of SGRM Models

Software trust worthiness development models gathered into two types of model such as Bowl- formed and S-formed. A limited amount of signs must include limited figure of errors. Rebuilding in addition to innovative product can cause innovative errors, which enhance the predetermined digit of failures [7], [8]. A few of models openly report the innovative description of the fault during the investigation at the same time as others believe that they are insignificant before managed beside the arithmetical reliability of appropriate software of the development model to the facts.



It is expected to facilitate the degree of shortcoming recognition is relative to the amount of cryptographic failures. Whenever a blemish is repair, there is a smaller quantity general fault within the cryptogram. Therefore, degree of knowledge of the defect decreases with the amount of fault stained .The bowl-shaped model exactingly pursues this scheme. In the S-formed model, it is whispered with the aim of the initial test does not as effective as after the test, hence there is an increase interval through which the degree of failure of cognition increases [7], [8]. This could be a good option. Suppose the first quality control tests honestly repeat the tests that the developers have already done or if the first quality control tests detect defects in different products that prevent the quality control from locating faults in the result in question. For example, a utility test can also detect errors that need to be corrected before running the software. Eight hours of application testing is collected, however, defect records are minimal due to the fact that defects are not remembered as part of software test statistics. After the defects have been modified, the rest of the request verification records (later, the modulation factor in the S-shaped arc) seems to be the concave model.

5. Software reliability development with Empirical Bayes technique

Empirical Bayes technique is procedures for arithmetical implication in which the previous circulation is approximated from the data [2], [3]. It also compares Bayesian and likelihood methods under different assumptions about prior information. Bayesian approaches are strictly connected to probability approaches. A probability distribution is used to define our previous views about a parameter or set of parameter. In directive to get the approximation the supply of "r" mathematically sovereign particular trials in "n" consequences. The "Binomial Distribution" is given by

$$P(r) = \binom{n}{r} \theta^r (1-\theta)^{n-r}.$$

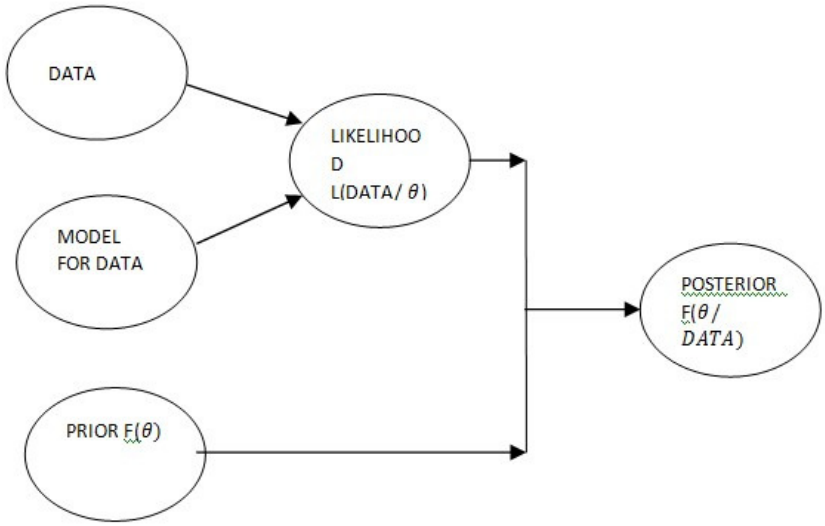
Their recurrence relation in "Binomial Distribution" is

$$P(r+1) = \frac{n-r}{r+1} \frac{p}{q} P(r).$$

The sources of prior information:

Individual Bayes: previous info subjective

Experiential Bayes: previous data from past data



6. Previous Data by Bayes Deduction

Bayes deduction offers apparatus for uniting previous data about sample data to provide inferences on model parameters [2]. A vector parameter θ can procedure as follows: θ Can be expressed in term of pdf $f(\theta)$ in prior information.

After evaluating data we observe that some data for specified model has likelihood

$$L(DATA/\theta) = L(\theta; DATA).$$

In Bayes theorem, the conditional distribution of θ given data (Posterior of) is

$$f(\theta/DATA) = \frac{L\ (\ DATA/\theta)\ f(\theta)}{f(L\ \left(\frac{DATA}{\theta}\right) f\ (\theta)\, d\theta} = \frac{R\ (\theta)\ f(\theta)}{f(R\ \ (\theta)\, f\ (\theta)\, d\theta},$$

where $R(\theta)$ is the related possibility and the manifold integral is calculated over $f(\theta) > 0$.

7. Neural network based LGCM of software reliability

ANN is the functional aspects Inspired by biological ANN [6], [1], [3]. It is based on available data in any non linear continuous functions and also has

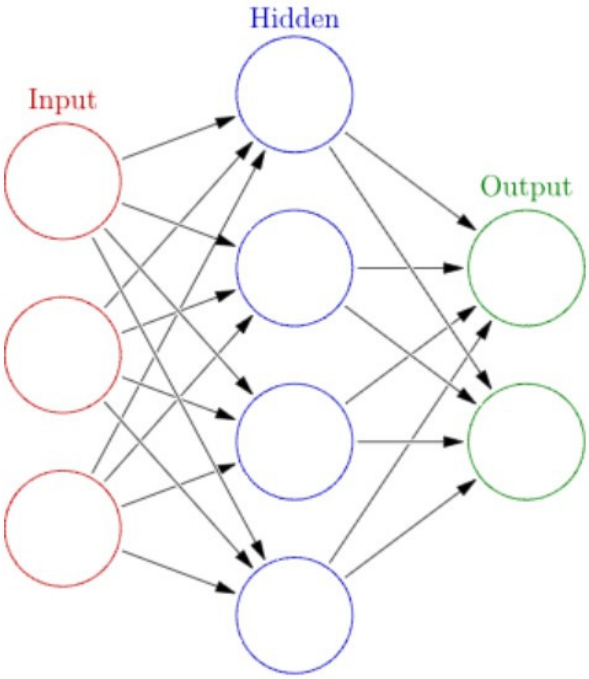


FIGURE 1. Artificial neural Network layer diagram

the learning capability it is same as an ”synthetic creature panicky structure” to get, progression and pass on information in terms of in sequence equipment Artificial neural network is following three layers:

(i) Enter level
(ii) Production level
(iii) Unknown level

Enter Level: The enter layer speaks with the outside condition that shows an example to the “neural system”. Its activity is to manage every one of the sources of info as it were [3]. This information gets exchanged to the shrouded layers which are clarified beneath. The info layer ought to speak to the circumstance for which we are preparing the “neural ’system. Each info “neuron” ought to speak to some free factor that has an impact over the yield of the “neural” structure.

Production Level: The yield level of the aural structure gathers and transmits the data as needs be in method it has been intended to give. The example displayed by the yield level can be straightforwardly followed back to the info level [1]. The volume of "neuron" in yield level ought to be specifically identified with the sort of exertion that the "neural" system was performing. To decide the volume of neuron in the yield level, first think about the proposed utilization of the "neural" system.

Unknown Level: The shrouded level is the accumulation of neurons which has actuation work connected on it and it is a middle of the road level initiate among the information level and the yield level. Its activity is to route the information sources acquired by its past level. So it is the level which is mindful extricating the requisite highlights from the information [6]. Numerous investigates has been made in assessing the volume of neurons in the concealed level yet at the similar period none of them was effective in finding the exact outcome. The production of an ANN considered as the compound functions. Now, the "logistic function" which is used to the mean worth purpose it writes as:

$$M(t) = \frac{s}{1 + ze^{\beta}} \quad \text{where} \quad s,\ z > 0.$$

In LGCM fits the mean worth purpose M (t) with a form of function. Now when we derive the composite function from logistic function then we are able to build an ANN based model. Suppose

$$p(x) = e^{\beta}, q(x) = \frac{1}{1 + zx}, r(x) = sx.$$

Then we get

$$r(q(p(x))) = r(q(e^{\beta})) = r(\frac{1}{1 + ze^{\beta}}) = \frac{s}{1 + ze^{\beta}}.$$

We see that LGCM can be decomposition into the functions of $p(x)$, $q(x)$ and $r(x)$.

8. Neural Network based NHPP of Software Reliability

NHPP stands for Non homogenous Poisson process. This process is based on parametric models .Basic parametric models is NHPP. NHPP Models was firstly proposed in 1979.

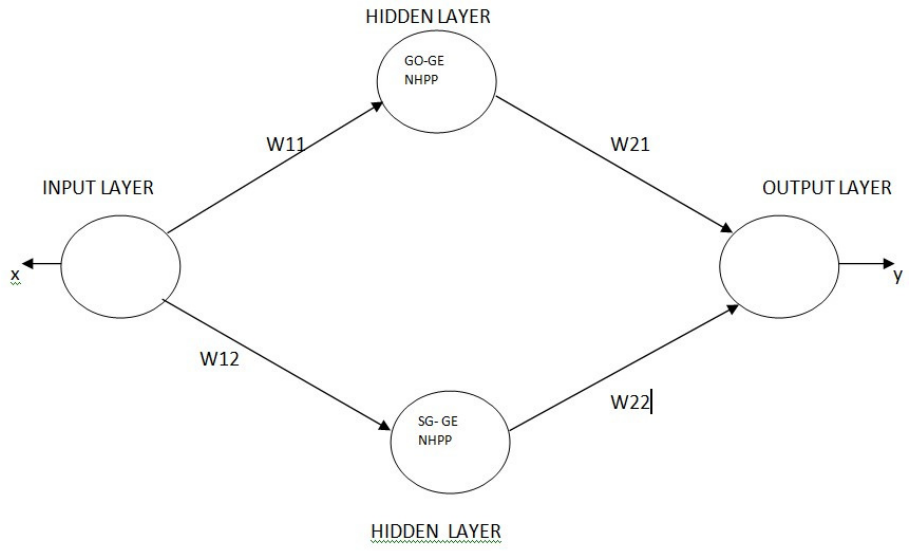


FIGURE 2. Proposed Feed Forward Neural Network Combination Model

NHPP model described by MVF (mean worth function) $\mu(t)$ and fiasco strength function $\lambda(t)$ which is the rate of change of mean value function. [5], [4]. The fiasco strength function of NHPP is capable of

$$\lambda(t) = (N - \mu(t))h(t),$$

where $h(t)$ fiasco appearance degree per fault.

Mean worth function of Geol. General NHPP (G-O GE NHPP) model

$$f(x) = a(1 - e^{-bx^c}).$$

Mean worth function of Subburaj-Gopal Widespread model

$$f(x) = a(1 - e^{-\frac{x}{b}})^c.$$

In a potent move towards to approximating the factors of software reliability growth replicas with a real-valued heritable procedure [4].

We criticized the output of the proposed neural network combination model is as follows [5]

$$Y(t) = W_{21}(1 - e^{-w_{11}x^{c_1}}) + W_{22}(1 - e^{\frac{-x}{W12}})^{c_2}.$$

Here w_{1j}, $w_{2j}(> 0)$ are the weights of the FFNN and their values are determined by the training algorithm [4], [5]. Here, c_1, $c_2(> 0)$ are activation function parameters whose values are also evaluated through the learning of the proposed FFNN. But this result is not sufficient as in equation (3) because of this result back to the result of input layer not in output layer so therefore the actual result is as

$$Y(t) = W_{11}(1 - e^{-w_{12}x^{c_1}}) + W_{12}(1 - e^{\frac{-x}{W_{22}}})^{c_2}.$$

Equation (4) gives the output result.

9. Conclusion

In this paper, we studied about various research papers and read like neural network, NHPP model, LGCM model, Empirical Bayes and software reliability growth model and its types. This document will also confirm that if the basic models are more flexible, we will obtain a better SRGM based on ANN. Also compared the NHPP model and reevaluate the formula of NHPP model. The primary reasons for these papers were to comprehend the idea of programming reliability utilizing the counterfeit neural organize. This paper presents the idea of neural model what's more, its engineering. In future we will plan a neural model for figuring the unwavering quality. A neural organize is considered as an advancing system which is used to scale the yield. We have demonstrated that neural system can be utilized for building programming dependability development models. NNs were ready to give models little SSE than the relapse demonstrate in every single thought about case. On the off chance that relapse shows with higher request have been considered most likely less SSE is acquired. Be that as it may, the quantity of the relapse show parameters will be expanded. This will require more perceptions for giving dependable gauge of the parameters. At present, we are examining the utilization of developmental calculations in to unravel the product unwavering quality development demonstrating issue.

References

[1] P. Roy, et al.: *Neuro-genetic approach on logistic model based software reliability prediction*, Expert Systems with Applications, **42**(10) (2015), 4709-4718.

[2] N.R. Barraza: *A Parametric Empirical Bayes Model to predict Software Reliability Growth*, Procedia Computer Science, **62** (2015), 360-369.

[3] A.L. GOEL, K. OKUMOTO: *Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures*, IEEE Transactions on Reliability, **28**(3) (1979), 206-211.

[4] I. LAKSHMANAN, S. RAMASAMY: *An artificial neural-network approach to software reliability growth modelling*, Procedia Computer Science, **57** (2015), 695-702.

[5] T. KIMA ET AL.: *An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm*, The Journal of Systems and Software, **102** (2015), 134-144.

[6] Y.S. SU, C.Y. HUANG: *Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models*, Journal of Systems and Software, **80** (2007), 606-615.

[7] B.B. SAGAR, R.K. SAKET, COL. G. SINGH: *Exponential Weibull distribution approach based inflection S-shaped software reliability growth model*, Ain Shams Engineering Journal, **7**(3) (2016), 973-991.

[8] G.E.P. BOX: *All Models are Wrong, some are Useful*, (1996), 1-29.

DEPARTMENT OF MATHEMATICS,
CHANDIGARH UNIVERSITY GHARUAN,
MOHALI PUNJAB, INDIA

DEPARTMENT OF MATHEMATICS,
CHANDIGARH UNIVERSITY GHARUAN,
MOHALI PUNJAB, INDIA