International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0063
ISSN (Online): 2279-0071

International Journal of Software and Web Sciences (IJSWS)
www.iasir.net

NEURAL NETWORK BASED SOFTWARE RELIABILITY

Gaurav Aggarwal[1], Dr. V.K. Gupta[2]
[1]Research Scholar, Jagannath University, Jaipur, Rajasthan, INDIA.
[2]Assistant Professor, Rajasthan University, Jaipur, Rajasthan, INDIA.

Abstract: *Software Reliability is the main concern due to the growing nature of industry. The reliability check is performed after the coding phase. The check failure leads to the reimplementation of the software. This paper designed a software reliability model that performs its working in two phases. First phase of the model is completed before the coding after the design phase. In this phase the design is checked against the requirements. The second phase of the model is placed after the implementation phase. The model is analyzed and results towards optimization.*

Keywords: *Neural Network, Software Reliability Error back propagation, Coding Phase*

I. Introduction

Software reliability is about to define the stability or the life of software system with different properties. These properties include the trustfulness of software system, software cost, execution time, software stability etc. The aspects related to these software system includes the probability of software faults, frequency of fault occurrence, criticality of fault, associated module with respective fault etc. In a software development process, the pre estimation of software reliability is required to deliver the software product. According to the required level of software quality estimation of software cost, development time is also estimated. There are number of quality measure that approves the software reliability [1]. Each stage of software life cycle itself takes some time quantum to deal with software reliability. Higher the software quality, lesser the software maintainability.

Software reliability growth models, refers to those models that try to predict software reliability from test data [2]. These models try to show a relationship between fault detection data (i.e. test data) and known mathematical functions such as logarithmic or exponential functions. The goodness of fit of these models depends on the degree of correlation between the test data and the mathematical function [3].

II. Existing Software Reliability Model

The author of [4] proposed a software reliability growth model on the basis of the existing model like logistic growth curve model etc. The model is designed on the basis of the tangential function. The tangential model must be drawn in the positive axis. It varies from 0 to infinity similar to the software reliability. The software reliability is inversely proportional to the fault detection as the no of fault detection decrease the reliability increases. The zero fault detection means the infinite reliability and the zero software reliability means the infinite faults. The proposed model suits the behavior of the software reliability so fits to the software reliability.

In the beginning of testing, there is exponential number of faults in the software code. The number of faults is unknown but they are fixed in number. All faults are of same type. Each fault can be detected independent of each other. The remaining number of fault and the remaining time is useful to determine the other parameters. The probability of occurring of each fault is same. Each fault occurred can be removed instantaneously. The mean value function can be given as

$$m(t) = \left(fv = \left(\frac{[1-\exp(-\phi t)]}{[1+\exp(-\phi t)]} \right) \right) > 0 ? fv : 0 \quad (1)$$

The failure intensity can be expressed as

$$\lambda(t) = \frac{dm(t)}{d(t)} \quad (2)$$

According to the failure intensity of the software at time t is proportional to the expected number of faults remaining in the software.

III. Proposed Model

In the existing software reliability models the failure check is performed after the coding and the implementation phase. Sometimes due to faulty designs requires reimplementation of the project. It leads to wastage of the resources and the time. The proposed software reliability model performs its working in two phases. First phase of the model is completed before the coding after the design phase. In this phase the design is checked against the requirements. This phase uses the error back propagation of the neural network. The second phase of the model is placed after the implementation phase. This model uses the mean time failure and intensity to increase the reliability. The detail working of the model can be understood by following model:

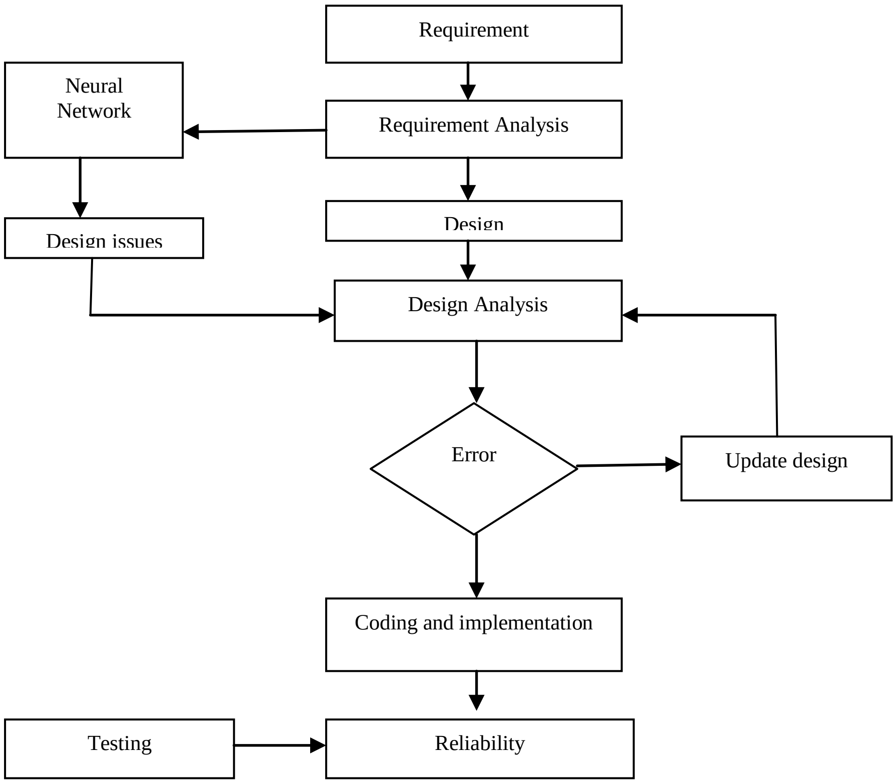


Figure 1: Block Diagram of Software Reliability Model.

A. Methodology

The methodology complete in two phases one is training phase and the other is testing phase.

Training Phase:

1. Input the requirement analysis .
2. Input the design issues corresponding to the requirement analysis.
3. Train the network(calculate weight matrix) . by using threshold activation function.

Testing Phase:

4. Input the requirement
5. Analyze the requirement
6. The requirement analysis is given as input to the neural network
7. Neural network process the requirement analysis and provides corresponding design issues.
8. The design issues are checked in the design.
9. If any error occur then design is updated and go to step 8
10. Perform Coding
11. Then get mean time failure

$$m(t) = \left(fv = \left(\frac{[1-\exp(-\phi t)]}{[1+\exp(-\phi t)]}\right)\right) > 0 ? fv : 0$$

12. Calculate intensity of failure

$$\lambda(t) = \frac{dm(t)}{d(t)}$$

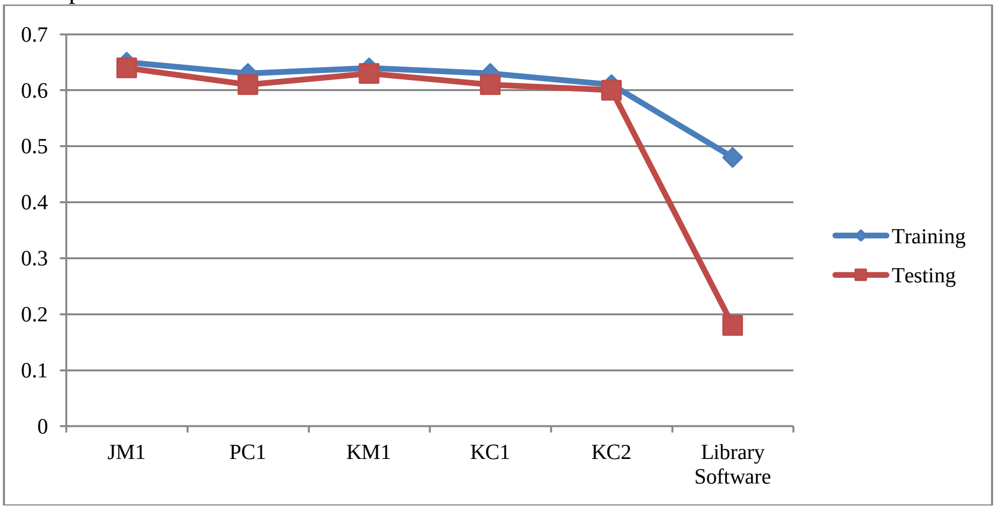
13. Remove failures.

IV. Implementation

The proposed methodology is analyzed in two manners. In the first way, library software is built from the initial phase and complete methodology is applied on the software for high reliability. The software is build for the Vaish College of Engineering Rohtak Haryana. In the second way, the proposed neural network based methodology is analyzed on the datasets downloaded from [5]. The dataset predicts the defects in the five modules of the NASA products. The NASA products under analysis are JM1,PC1, KM1, KC1, and KC2. The variables in this dataset are evaluated by using static measures i.e. prediction variables. The subsets in the dataset are prepared by classifying the set on the basis of size of module. This results in the high prediction performance. The quality is better in the class level data prediction as compared to method level data prediction. The defect prediction is more accurate in the large modules as compared to small modules. The present work uses the 60% of the dataset for the training purpose and rest for the testing purpose. The target of this work on this dataset is to find the software reliability by finding the defects accurately. :

Dataset	RMSE	
	Training Data	Testing Data
JM1	0.65	0.64
PC1	0.63	0.61
KM1	0.64	0.63
KC1	0.63	0.61
KC2	0.61	0.60
Library Software	0.48	0.18

The graphical representation of the above values is shown below:



The reduce in the error confirms the better performance of the model.

V. Conclusion

This paper introduces a software reliability model that performs the reliability check before the implementation phase as well as after the implementation phase. This model is suitable only if the software is building from the scratch, otherwise only the second phase of the model is applicable and the performance of the model is still better than other software reliability growth curve models. In future the model can be analyzed over large software been developed.

References

[1] Garima Chawla et. al. , *A Fault Analysis based Model for Software Reliability Estimation*, International Journal of Recent Technology and Engineering (IJRTE),ISSN: 2277 -3878, Volume-2, Issue-3, July 2013.

[2] Rita G. Al gargoor et. al. , *Software Reliability Prediction Using Artificial Techniques*, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No 2, July 2013,ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784,www.IJCSI.org.

[3] Al-Rahamneh Z., Reyalat M. Sheta A. F., Bani-Ahmad S., and Al-Oqeili S., *A New Software Reliability Growth Model: Genetic-Programming-Based Approach* , Journal of Software Engineering and Applications, 2011, pp. 476-481.

[4] Gaurav Aggarwal, V.k Gupta, "Software Reliability Growth model," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, Issue 1, January 2014, ISSN 2277 128X.

[5] http://promise.site.uottawa.ca/SERepository/datasets