International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

The Implementation And Automation Of Software Reliability Assessment Model Using Neuro Fuzzy System

Bonthu Kotaiah
[1] Assistant Professor, Maulana Azad National Urdu University, Hyderabad, India
[1]kotaiah.bonthuklce@gmail.com

ABSTRACT

Dependency on computer aided systems is increasing heavily now a days in the modern era with the evolution of latest concepts, technologies and tools in the field of computer science. At the same time the quality of service is being affected by some software failures or faults or bugs in order to meet the required level of performance and accuracy levels which ultimately leads so many people to go for alternatives of the old softwares. To address the issues so many software engineers, developers and system analysts recognized that some software functions are critical and their consequences of the problems are significant enough and they tried to find out/suggest the solutions for them. Let us examine the different definitions given by different scientists for the software quality and software reliability. In this project we proposed an approach for the implementation of Software Reliability Assessment Model based on Neuro-Fuzzy based Systems using MTBF values as a parameter.

1.INTRODUCTION

1.1 Neuro Fuzzy Models

The Neuro Fuzzy system concept is to find the parameters of a fuzzy system by means of learning algorithms or learning laws obtained from Neural Networks methodology. The Supervised or Un-supervised learning techniques/algorithms that can be used to create fuzzy systems for data can be used to model Software Reliability Assessment model; a common way to apply a learning algorithm to a fuzzy system is to represent it in a special neural-network-like architecture like single layer or multi-layer architecture. Then a learning algorithm – such as back propagation or similar kind of algorithms can be used to train the system to model the Software Reliability. In this context, some problems will be raised in modeling since Neural network learning algorithms are usually based on gradient descent methods. They cannot be applied directly to a fuzzy system, because the functions used in the inference process are

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

usually not differentiable. There are two solutions to this problem:

a) Replace the functions used in the fuzzy system (like min and max) by differentiable functions, or

b) Do not use a gradient-based neural learning algorithm but a better-suited procedure.

The Neuro Fuzzy technique, then, is used to derive a fuzzy system from data, or to enhance it by learning from examples. The exact implementation of the Neuro Fuzzy model does not matter. It is possible to use a neural network to learn certain parameters of a fuzzy system, like using a self-organizing feature map to find fuzzy rules, or to view a fuzzy system as a special neural network and to apply a learning algorithm directly.

The Structure of Adaptive Neuro Fuzzy Inference System is shown in Figure 1.4.

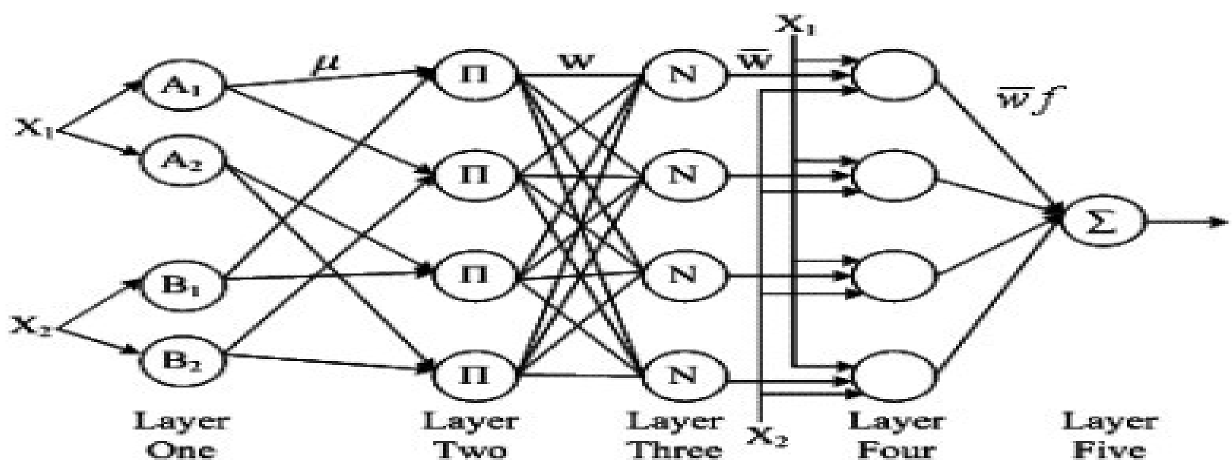


Figure 1.4: Structure of Adaptive Neuro Fuzzy Inference System.

The rule structure is made up of Neuro Fuzzy commands of “IF”, “AND, “Continues” and “THEN” carefully connected together to arrive at the specified desired parameter (output).

1.2 Objectives

The objectives set for the study in the project are as follows:

- To review and examine the literature on Software Reliability in relation with MTBF and reliability estimation models and the available metrics.

- To identify the factors that affect the Software Reliability at different phases of SDLC(Software Development Life Cycle)
- To find out, analyze the effect of errors, bugs and failures in the software under the development.
- To collect the datasets of software programs from running software with the appropriate runtime errors that is useful for the assessment.
- To formulate a theoretical analysis for the evaluation of the metrics used for assessment and development of model.
- To identify how MTBF relates with the Software Reliability.
- To calculate the metrics with the given datasets both analytically and programmatically.
- To develop a model for the assessment of Software Reliability based on Neuro Fuzzy Systems approach and to implement it.
- To train the neural network with some collected Software Reliability parameters (at different phases of SDLC) mapped to numerical data and are loaded into neural network at input layer.
- To assess and evaluate the performance of the trained network for Software Reliability at the design level with some numerically approximated values by using fuzzy membership function (sigmoid).
- To compare the approximated Software Reliability against the expected reliability approximation.
- To compare the performance of proposed model against conventional FIS (Fuzzy Inference System) models based on evaluation and validation metrics to prove that the proposed model is the promising one than the others.
- To validate the proposed model both theoretically and statistically.

2. Background

In 2014 Kirti Tyagi, Arun Sharma [19] suggested An adaptive Neuro Fuzzy model for estimating the reliability of component-based software systems. In their work, they proposed a model for estimating CBSS reliability, known as an adaptive Neuro Fuzzy Inference System (ANFIS), that is based on these two basic elements of soft computing, and the research study compare its performance with that of a plain FIS (Fuzzy Inference System) for different data sets. In 2013 Hu.et.al [13] estimate the Software Reliability through testing, an extended adaptive testing strategy, namely Modified Adaptive Testing (MAT). In the same year, Pooja Rani *et.al.* proposed a risk predicting tool based on Neuro Fuzzy approach for software Risk Prediction. Firstly Fuzzy Inference System is created and then Neural Network based three different training algorithms: BR (Bayesian Regulation), BP (Back propagation) and LM (Levenberg-Marquardt) are used to train the neural network. In [2013] Lance Fiondella et.al presents an efficient methodology based on the multivariate Bernoulli (MVB) distribution to analyze the reliability of a software application considering COCOF.Unlike the earlier techniques, the proposed methodology introduces only a quadratic number of parameters [11].

In 2012 Singh et al [15] explain that the transfer function is a function used to transform the activation level of a neuron into an output signal. The behavior of the ANN depends on both the weights and the activation function that is specified for the neuron. Neurons are structured depending on the learning algorithm used. Learning consists of adjusting the weight and threshold values until certain criteria are encountered for example the training error falls below some user-defined level, the number of training cycles or epochs exceeds a certain value or all the validation tests have been performed.

In 2011, we propose an adaptive framework of incorporating path testing techniques into the reliability estimation for modular software systems. Three estimated methods based on common program structures are presented to calculate the path reliability. The derived path reliabilities then are used as an approximation of software reliability. When more testing paths are available as testing proceeds, our proposed methods can then recalibrate the estimated accuracy [12]. In 2011 Bragina, T. & Tabunshchyk, G. Analysed the project risk, that depends on basic software risk level of projects with iterative lifecycle and software development project model [17]. Fuzzy model for the project risk are proposed by the authors. In 2011, Marko Palviainen et.al defines an approach for reliability evaluation that assists software developers in producing predicted reliability values for not-yet-implemented components, to measure reliability values for components and to evaluate how different component selections affect the reliability of a software system.[14].

3. Implementation: Software Reliability Assessment Model Using Neuro-Fuzzy System

After observed different issues in predicting the Software Reliability [1][10] and also different levels of severity faced by organizations in the case of reliability issues [4], An attempt has been made to implement a model based on the Neuro Fuzzy System. Here, the complex problem of Software Reliability modeling can be split to number of sub-problems:

- Selection of the Software Reliability model structure.
- Estimation of the Software Reliability model parameters like MTTF [2] (Mean- Time-To-Failure) and Mean-Time- To-Repair (MTTR) , MTBF and availability etc.,
- Evaluation of the model prediction capabilities.

The preliminary prediction of Software Reliability prediction using Neural Network Based Systems and the comparison of Software Reliability Assessment Methods [20] with Neuro Fuzzy Based Systems [6][7][9] was observed by the researcher to arrive at the conclusion for the development of the proposed approach [5][8]. Figure 3.3 shows the proposed model of the research analysis, where the parameters concerned with the reliability assessment are given as the inputs for network. Weights are been added at the consequent layer ad using the predefined rules a decision is obtained at the validation. Based upon the outcome of validation the assessment will be finalized. A generalized block diagram of the proposed approach is shown in the Figure 3.3. In this research work a Neuro Fuzzy interference model is designed for the assessment of reliability of a software growth model, the algorithm mainly focuses on MTBF which is analyzed and calculated theoretically and practically. Fuzzy rules employed for the proposed model are:

- *If MTBF (Mean time Between Failure) >0.8 then reliability is very high*
- *If 0.7<MTBF <0.8 then reliability is high*
- *If 0.6<MTBF <0.7 then reliability is moderate*
- *If 0.5<MTBF <0.6 then reliability is low*
- *If 0.4<MTBF <0.3 then reliability is very low*

In order to assess the reliability of software during design phase using Neuro Fuzzy Logic, the model is divided into five phases. These phases are as follows:

3.1.1 Recognition Phase

The objectives of identification phase are:

1. To identify the reliability factors.
2. To evaluate a mathematical analysis for the approximation constraints.

3.1.2 Quantification Phase

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

The objectives of the quantification phase are:

1. To identify the how the reliability factors are linked with assessment of reliability and quantify them.
2. To evaluate a mathematical analysis for the relationship.

3.1.3 Assessment Phase

The objectives of the measurement phase are

1. To assess the metrics for the estimation of reliability.
2. To conduct the experiments for the generation of results.

3.1.4 Testing Phase

The objectives of verification and validation phase are

1. To verify the model's effectiveness in the assessment of software reliability.
2. To compare and validate the reliability model with conventional models.

3.1.5 Maintenance Phase

The objectives of finalization phase are

1. To incorporate the changes and suggestions.
2. To finalize the metrics and model for assessment.

3.1.6 Revision Phase

There is a review and revision of output of every phase for the refinement of the study. The aim of this phase is to check whether the objectives of each phase are fulfilled or not. The review and review process is depicted in Figure 3.2.

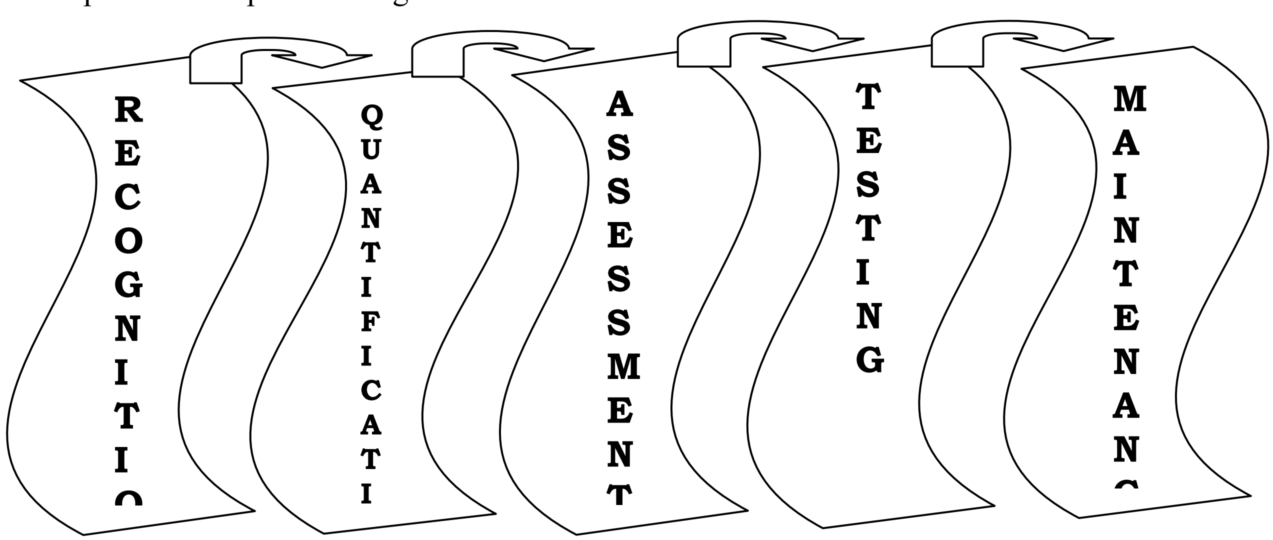
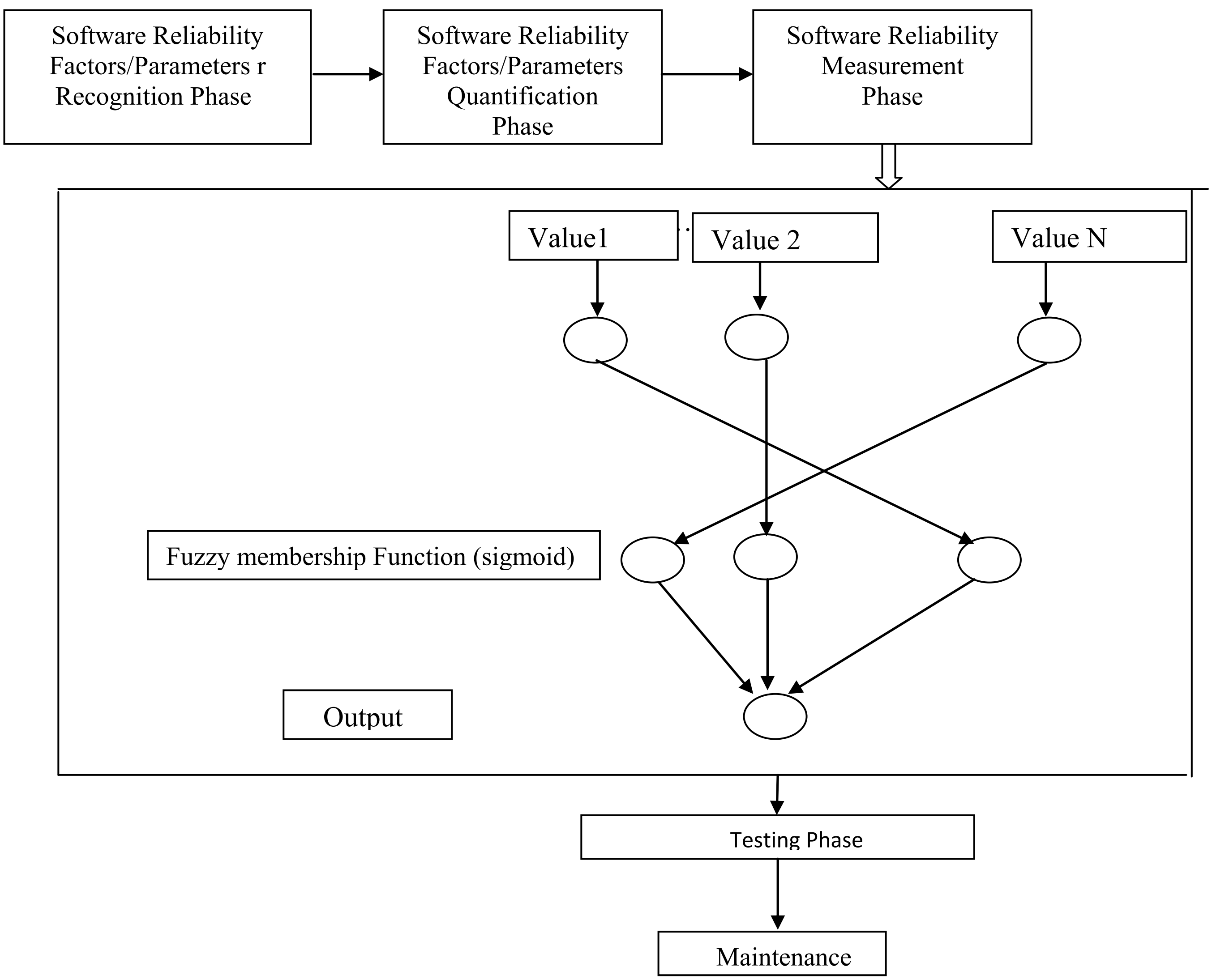


Figure 3.2: Process Flow of the Proposed Approach

3.2 Implementation of the Proposed Approach

The Figure 3.3 shows the steps to be followed for implementation of the approach [3]. The values of Parameter Normalized MTBF are given as input at the input layer of Neural Network. Sigmoid fuzzy membership function at the hidden layer of neural network is applied and the Software Reliability approximated value is found out. During validation, the values assessed using conventional FIS system and the value obtained from the proposed Neuro Fuzzy systems based model will be compared against two evaluation criteria like MSE (Mean Squared Error) and Average Error (AE).



e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

Figure 3.2: Proposed Neuro Fuzzy based Model of Software Reliability Assessment

Mathematical approximation of proposal model metrics

For approximating the values of the proposed model metrics, a quantitative approach is adopted for calculating the appropriate results. The formula that has been used to calculate approximated values is defined as:

Formula: Ca (x_i) = C (a) - h X f(a), based on Euler's theorem

Where, C (a) = Set of Measured values.

'h' can be derived by,

$$x_1 + x_0 n h$$

Where, n= no. of values in the dataset.

$x_0 = 0$ and $x_1 = 1$ (since the probability ranges from 0 to 1). Here 'x' is MTBF. f(a) can be function, denoted as,

f(a)=MTBF/(1+MTBF)

Ca (x_i) is the set of values to be approximated.

Procedure for 'h' Calculation

Let us take, $x_0 = 0$ and $x_1 = 1$ then, 1= 0 + 17 * h

h= 1/17 = 0.058

According to [18], Iterations are Performed at least 5 to 10 iterations to arrive at good approximated Software Reliability value.

At every iteration, to calculate % of Reliability, use the following formula

% of Reliability = (Average of Approximated values)/ (Average of Measured values) * 100

At final iteration, if the approximated value falls above 99.00%, then the study can say that it is good approximation.

3.3 Data Collection

To validate the proposed model, 17 programs of Glace EMR Medical Billing Software are taken (on which researcher has worked previously as a Software Engineer at L Cube Innovative Solutions Pvt. Ltd.). The MTTF (Mean Time To Failure), MTTR (Mean Time To Repair), MTBR (Mean Time Between Repair) and Software Reliability Approximated value based on the program execution observations are calculated. The 3 values were put to the input layer of Neural Network. Sigmoid fuzzy membership function at the hidden layer of neural network was applied and the Software Reliability approximated value was found out. The previous values assessed using conventional FIS traditional Software Reliability Growth Models and the proposed Neuro Fuzzy systems based model is compared.

It was found that the proposed model is the promising one. The amount an element is in a set is measured with a membership function. Membership functions range from 0 to 1. Membership functions are used to describe linguistic terms such as low, medium and high. There are various types of fuzzy membership functions such as triangular, trapezoidal, and Gaussian [16].

The dataset contains failure observations of 17 programs in Glace EMR Billing Software, in time series (i, Xi) and is used to predict the performance of the proposed model. Where, i = serial number of the program. The models are applicable to the area of Software Reliability engineering. Table 4.1 shows all the 17 projects and the information recorded. The application types are Patient Registration, Service Entry, Reports, Online Patient Insurance Verification applications. The attributes recorded for each software are Software Code, Type of Application, Size of Software (in Lines of Code (LOC)), Number of Failures.

Table 4.1: Software Reliability Data Project Information

Software	Type of Application	Size(LOC)	No. of Failures
GE01	Patient Registration	22,300	14
GE02	Patient Registration	10,500	25
GE03	Patient Registration	9,800	21
GE04	Patient Registration	31,870	39
GE05	Patient Registration	12,400	48
GE06	Service Entry	4,870	36
GE07	Service Entry	26,490	35
GE08	Service Entry	23,400	35
GE09	Service Entry	21,700	45
GE10	Reports	10,890	10
GE11	Reports	28,740	58
GE12	Reports	36,350	54
GE13	Online Patient Insurance	61,800	32
GE14	Online Patient Insurance	34,700	21
GE15	Online Patient Insurance	39,800	52
GE16	Online Patient Insurance	43,200	58

GE17	Online Patient Insurance	44,600	56

3.4 Parameters used for Measurement and Validation

MTBF consists of Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR). MTTF is the difference of time between two consecutive failures and MTTR is the time required to fix the failure.

MTTF = Average time between 2 observed failures. i.e., average time it takes for a system to fail

MTBF = Average time between consecutive software system failures =MTTF+MTTR

MTTR = Average time taken to repair the system after the occurrence of failure.

<u>Methods to Assess Software Reliability</u>

There are two methods to assess the software reliability. The methods are as follows:

- Counting failures at periodic intervals of time($\mu(\tau)$) : Observe the way of cumulative failure count = Total Number of failures observed until execution.

  τ - Time from start of system execution.

- Based on Failure Density($\lambda(\tau)$):

  Observe the trend of number of failures per unit time. i.e., no. of failures observed per unit time after 'τ' time units from the staring stage of the system execution, which can be called as failure intensity at time, 'τ'.

The value of different parameters is calculated as:

Software Reliability = MTBF / (1+MTBF).

Normalized MTBF = (sum of MTBF values observed)/ number of readings.

In the study, the parameter is normalized MTBF to arrive at the effective assessment of Software Reliability using the proposed approach.

3.5 Parameters Estimation and Experimental Results

The experiments are conducted to prove that the proposed approach gives better results than the conventional Fuzzy Inference System (FIS) and other traditional Software Reliability assessment models. The MATLAB software

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

has been used as a base for conducting the experiments.

3.5.1 Parameters Estimation

The calculation of normalized Mean Time Between Failures (MTBF) is done mathematically and the results are shown in Table 4.3 and table 4.4 respectively. Then, the assessed values are given as input to Neuro Fuzzy Systems based approach for the better assessment of Software Reliability. Based on the experiments conducted and the results observed, the researcher concludes that the proposed approach perform well. To conduct the experiments the dataset is recorded based on the factors like total production time, uptime, downtime, No. of breakdowns at two intervals of time 'x1' and 'x2' and the recorded values are shown in Table 4.2.

Table 4.2: Production Time Analysis for the Program Dataset

S.No	Program #	Total Production time(Hrs.)	Uptime at x1(Hrs.)	Uptime at x2(Hrs.)	Downtime at x1(Hrs.)	Downtime at x2(Hrs.)	No. of breakdowns at x1(Hrs.)	No. of breakdowns at x2(Hrs.)
1	GE01	256	216	202	40	54	3	11
2	GE02	324	260	203	64	121	9	16
3	GE03	236	168	154	68	82	2	19
4	GE04	600	450	435	150	165	16	23
5	GE05	371	300	265	71	106	13	35
6	GE06	447	430	410	17	37	15	21
7	GE07	865	560	525	305	340	10	25
8	GE08	843	615	575	228	268	4	31
9	GE09	943	720	706	223	237	17	28
10	GE10	135	85	78	50	57	4	6
11	GE11	242	130	132	112	110	36	22
12	GE12	369	240	206	129	163	24	30
13	GE13	122	68	64	54	58	23	9
14	GE14	107	72	74	35	33	6	15
15	GE15	371	265	253	106	118	18	34
16	GE16	453	370	398	83	55	21	37
17	GE17	325	285	256	40	69	27	29

Calculations

For all observations, the following parameters can be calculated as follows:

Total Production time= Uptime+ down time

$$\textbf{MTBF} = \frac{\text{Total uptime (total time- total downtime)}}{\text{Number of Breakdowns}} \quad \text{for all observations}$$

Where,

MTTF= Mean Time To Failure (in hours/minutes/seconds).

MTTR= Mean Time To Repair (in hours/minutes/seconds).

MTBF= Mean Time Between Failures (in hours/minutes/seconds).

$$\text{MTTR} = \frac{\text{Total downtime}}{\text{Number of breakdowns}} \quad \text{for all observations}$$

$$\text{MTTF} = \frac{\text{(Failure at obs.1+ Failure at obs.2+…+ Failure at obs.N)}}{\text{Number of software programs under test}}$$

Table 4.3: Calculation of MTBF & MTTR

S.No.	Program	MTTF	MTTR	MTBF
1	GE01	0	9.12	45.18
2	GE02	0	7.336	20.78
3	GE03	0	19.158	46.05
4	GE04	0	8.25	23.51
5	GE05	0	4.25	15.32
6	GE06	0	1.447	24.09
7	GE07	0	22.05	38.5
8	GE08	0	32.82	86.14
9	GE09	0	10.791	33.784
10	GE10	0	11	17.12
11	GE11	0	4.056	4.80
12	GE12	0	5.042	8.43

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

13	GE13	0	4.396	5.03
14	GE14	0	4.016	8.46
15	GE15	0	4.679	11.08
16	GE16	0	2.719	14.18
17	GE17	0	1.93	9.69

The ratio of MTTR and MTBF w.r.t. number of programs in the dataset is represented in Figure 4.5, Figure 4.6 respectively.

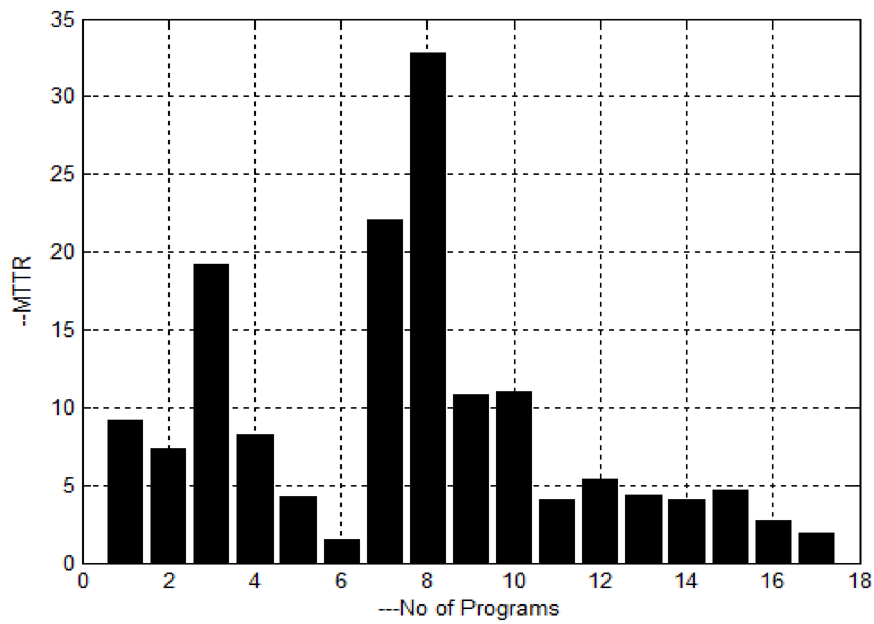


Figure 4.5: Analysis of MTTR Ratio w.r.t. Number of Programs

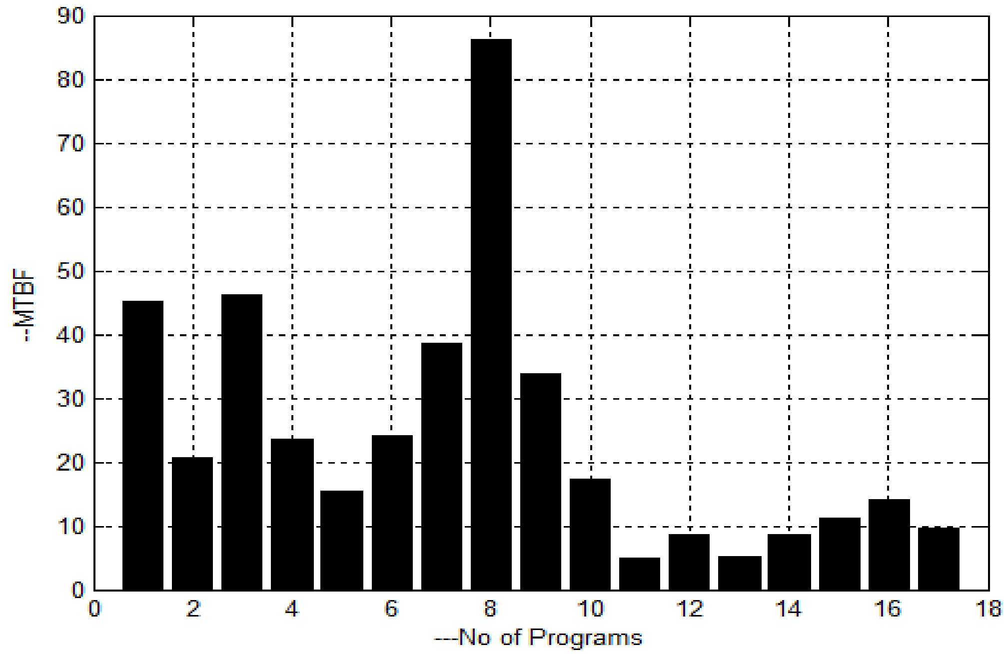


Figure 4.6: Analysis of MTBF Ratio w.r.t. Number of Programs

3.5.2 Implementation of the Proposed Approach through MATLAB

The experiment is conducted with the 17 programs of Glace EMR Medical Billing. The analysis is done using FIS (fuzzy interference system) and the proposed Neuro Fuzzy model in MATLAB environment. The Fuzzy Inference Systems based model is presented with the parameter Normalized MTBF in Figure 4.7. The generated membership function based on employed fuzzy IF-THEN rules ranging from "very low" to "very high" are shown in Figure 4.8. The dataset will be given as training data to ANFIS systems and the comparison between ANFIS output and FIS output is shown in Figure 4.9. The sequence of the steps from feeding of the input to generation of output at different layers of Neuro Fuzzy System like input, processing of input through membership function and Fuzzy IF-THEN rules, the generation of output is shown in Figure 4.10. The generated error after learning of inputs through training is based on number of epochs in Figure 4.11. Finally, the performance level of the proposed approach for conversion of input space to output space is depicted as a surface in Figure 4.12.

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

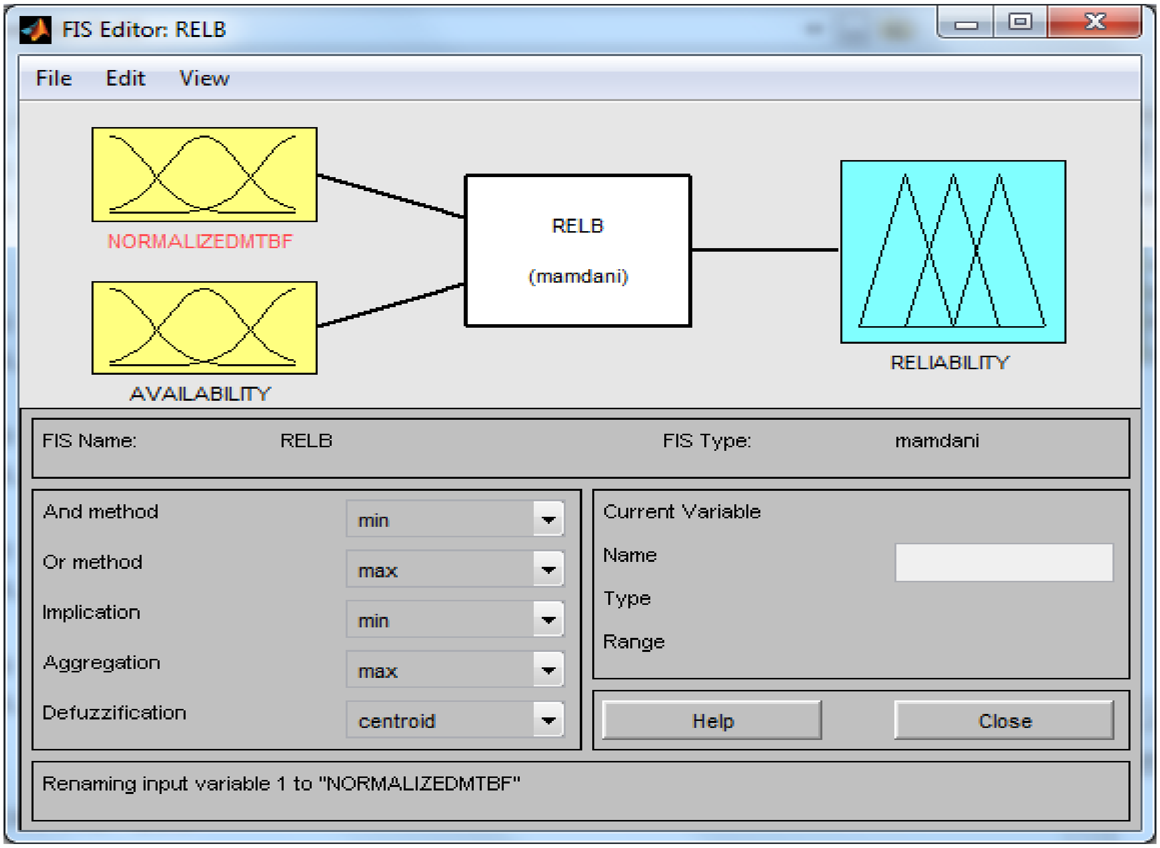


Figure 4.7: FIS System Model

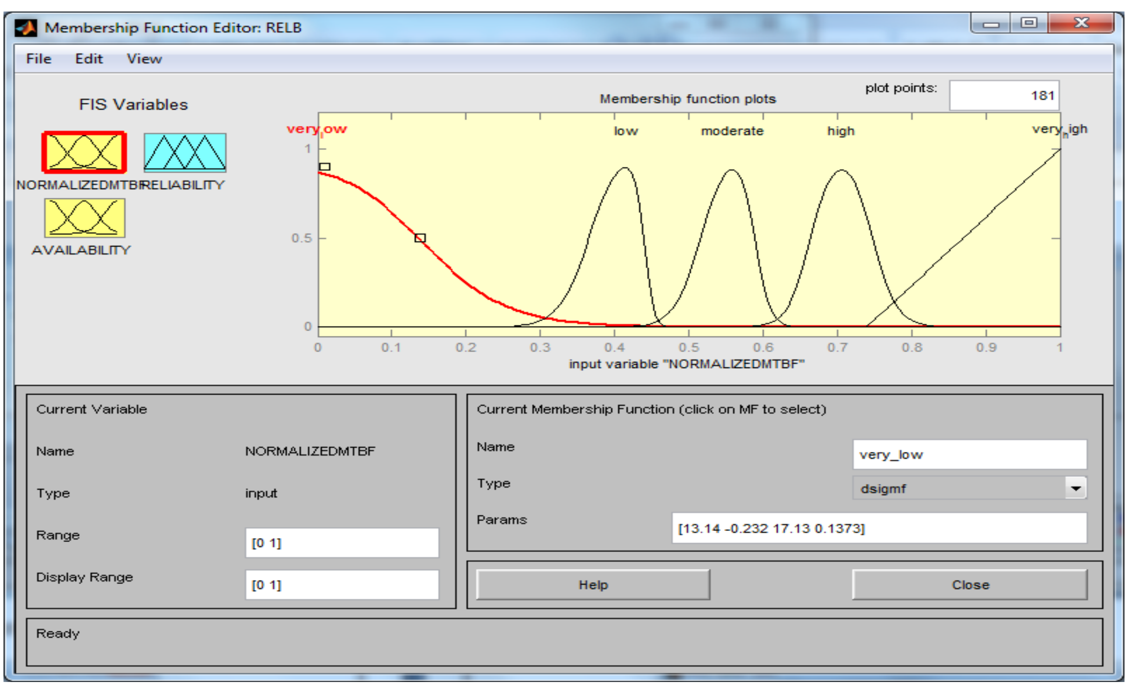


Figure 4.8: Membership Function for MTBF and Availability

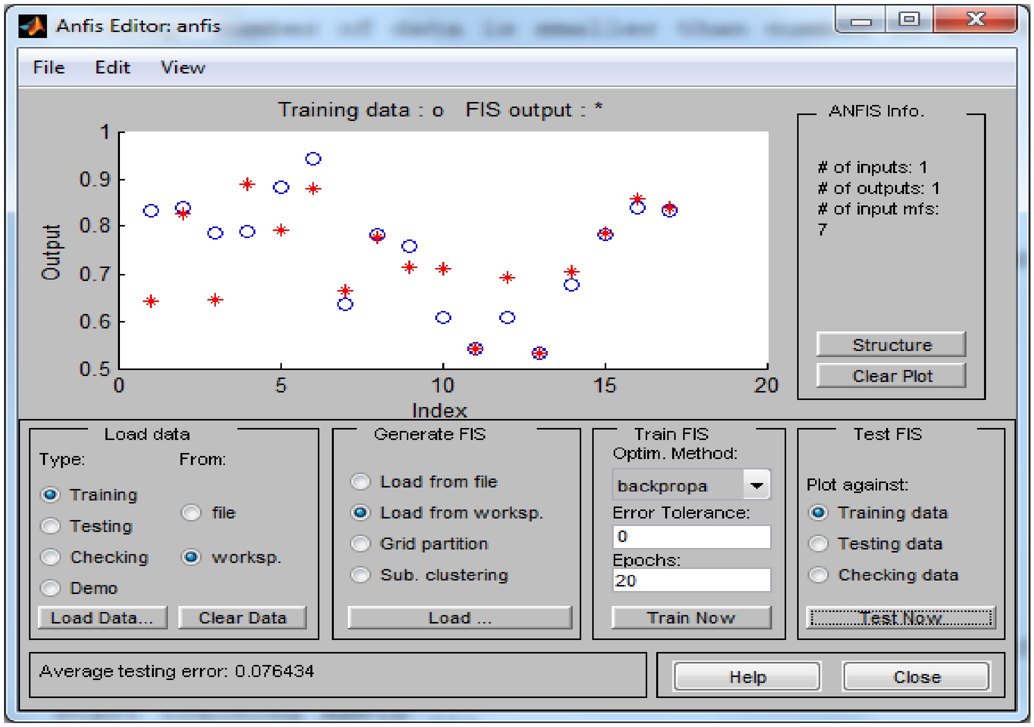


Figure 4.9: Neuro Fuzzy Inference Model

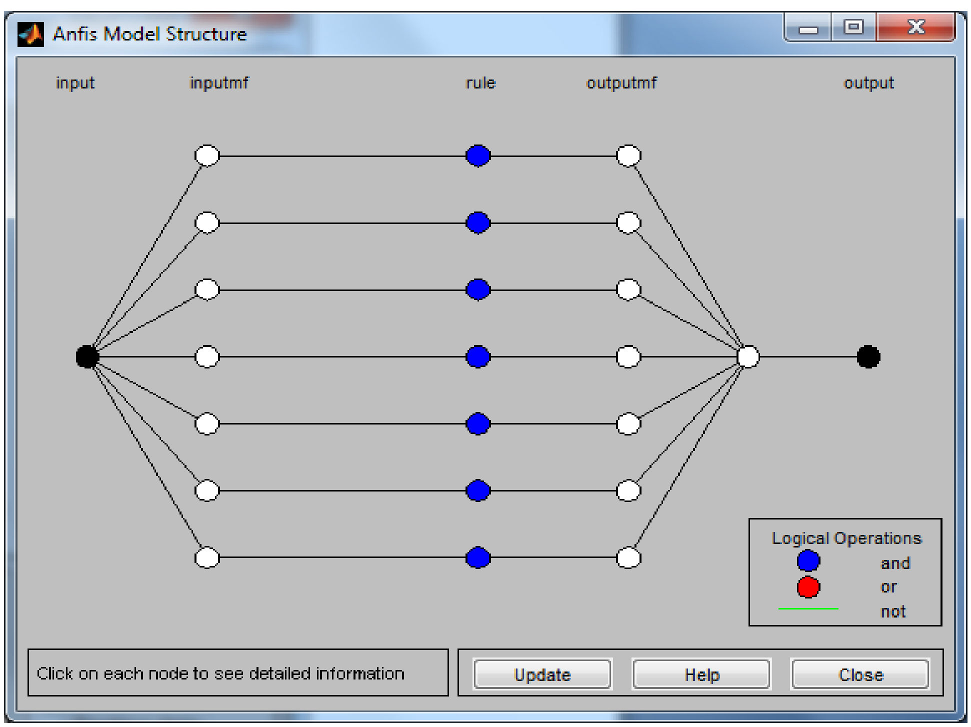


Figure 4.10: Neuro Fuzzy Structure

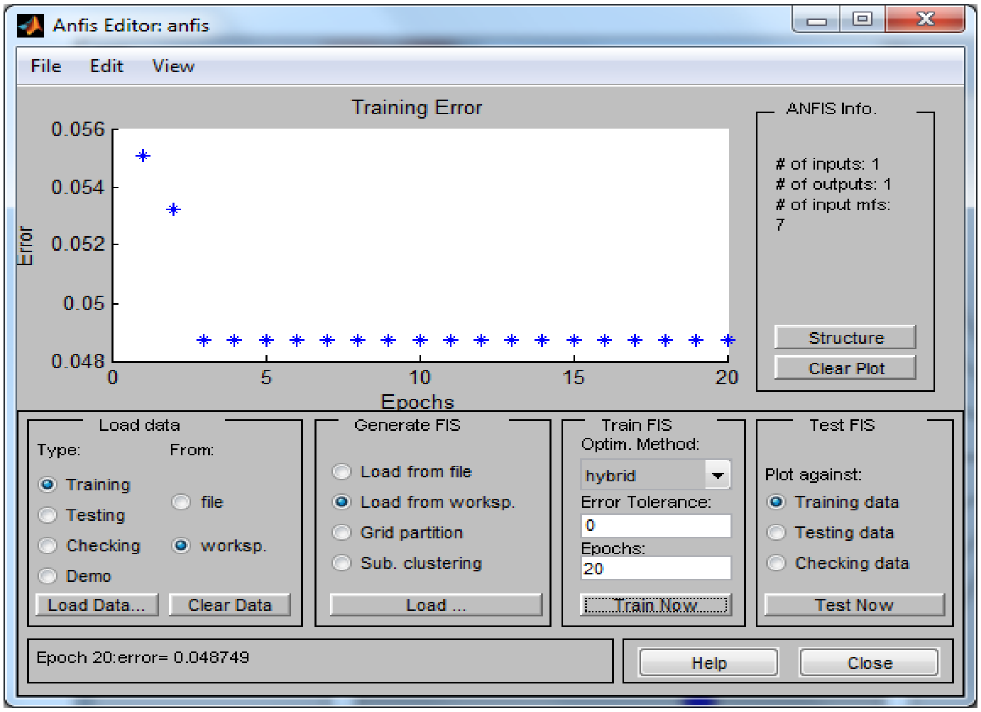


Figure 4.11: Error Tolerance

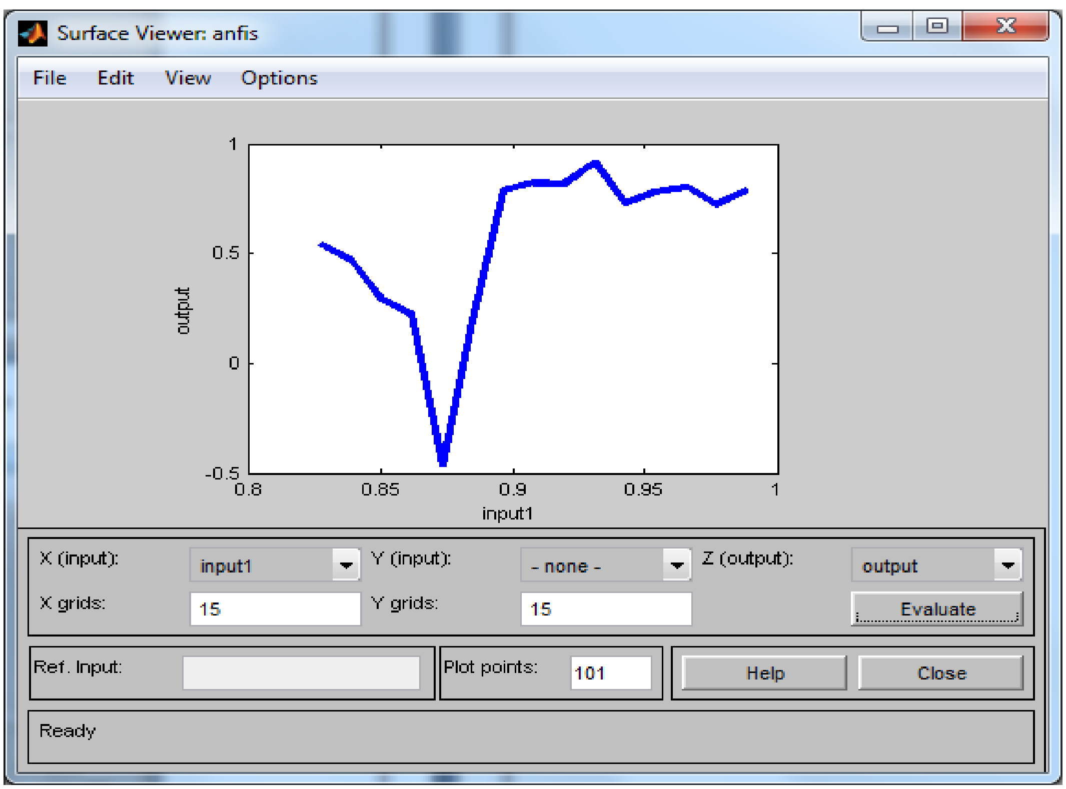


Figure 4.12: Performance Analysis

4. Conclusion and Future Work

The conclusions made by our project study are as follows:

- Normalized MTBF is taken into consideration for the effective assessment of Software Reliability using Neuro Fuzzy based Systems.
- Fuzzy Rules are implemented properly to guide the Assessment process.
- Fuzzy Membership function like sigmoid function is taken into consideration for the assessment using ANFIS.
- The observations conclude that proposed network model performs better in terms of less error in prediction as compared to existing analytical models and hence it is a better alternative to do Software Reliability test using neural network. However it can be seen from the Software Reliability estimates that the ANFIS method proposed in this research provides a good fit than analytical models.
- The researcher plans to predict Software Reliability using hybrid intelligent system. In addition to neural network model genetic programming can be applied further.
- The researcher plans to develop a novel recurrent architecture for Genetic Programming (GP) and Group Method of Data Handling (GMDH) and also GMR(Group Maturity Rating) in combination of Fuzzy logic for predicting Software Reliability.

5. Limitations and Delimitations

As every coin has two faces. The proposed work too suffers from the following limitations:

- This approach is tested on a smaller dataset using one software only.
- It would be more accurate if it has been tested on multiple datasets on different software outcomes.
- No research study completes in all aspects, there is always a scope for further improvement.
- High values of MTBF increases the Software Reliability, means there exists a linear relation between the parameters considered.
- The model is validated with only a small dataset(17 programs of GlaceEMR software).

On the other hand the approach has the following merits:

- The Neuro Fuzzy approach is most accurate mode of measurement.
- The approach is very easy to implement.

References

[1] Bonthu Kotaiah, R. A. Khan (2013), "Different Issues in Predicting the Software Reliability", *International Journal of Science and Engineering Investigations (IJSEI)*, Vol. 2, Issue 12, Pp.65-68, ISSN: 2251-8843.

[2] Bonthu Kotaiah, R. A. Khan, D.S.R. Krishna (2013), "A Neural Network methodology for Software Reliability prediction of long-term MTTF", *Mathematical Sciences International Research Journal*, Volume 2, Issue No.1, ISSN 2278-8697 Pp. No.101-103.

[3] Bonthu Kotaiah, R.A. Khan (2015), "Effective Assessment of Software Reliability by Using Neuro-Fuzzy System", *International Journal of Research (IJR)* e-ISSN: 2348-6848, p-ISSN: 2348-795X Volume 2, Issue 08, Pp. No. 250-270.

[4] Bonthu Kotaiah, R.A. Khan (2013), "Assessing the level of Software Reliability using Neuro Fuzzy System", *International Conference on Information & Engineering Sciences (ICIES -2013)* (Print Journal).

[5] Bonthu Kotaiah, R.A. Khan (2013), "The Prelimilinary Prediction of Software Reliability Prediction using Neural Network Based Systems", *International Conference on Mathematical Modelling And Numerical Simulation(ICMMANS), BBAU, Conference Proceedings, Pp. No.62.*

[6] Bonthu Kotaiah, R.A. Khan(2013), "Comparison of Software Reliability Assessment Methods with Neuro Fuzzy Based Systems", *International Conference on Nanoscience And NanoTechnology (ICNN), BBAU, Conference Proceedings. Pp. No.256.*

[7] Bonthu Kotaiah, R.A. Khan(2013), "Software Reliability Assessment by using Neural Networks with Fuzzy Logic based systems", *International Conference on Advances in Computer Science(AET-ACS),Elsevier,onlinelink:http:searchdl.org/public/book_serieselsevierst/6/ 540.pdf.*

[8] Bonthu Kotaiah, R.A. Khan (2014), "A Comparative Study on Software Reliability assessment with Neural Networks and Neuro Fuzzy based systems", "*International Conference on Modeling and Computing" (ICMC-2014)*, PP No.12, Souvenier, BBAU, Lucknow, Conference Proceedings.

[9] Bonthu Kotaiah , R.A. Khan(2013), **“**Software Reliability Assessment by using Neural Networks with Fuzzy Logic Based Systems**”**, *Advances In Engineering And Technology Conference(AET)*, Delhi, NCR, Elsevier, Conference Proceedings.

[10] Bonthu Kotaiah , R.A. Khan(2014), “Issue in the Assessment of Software Reliability with Mathematical Modelling Based Safety-Critical System”, *National Conference on Information Security Challenges(NCISC- 2014)*, Organized by Department of Information Technology, Babasaheb Bhimrao Ambedkar University(BBAU), Lucknow, Conference Proceedings.

[11] Fiondella, Lance, Rajasekaran, Sanguthever, Gokhale, Swapana, (2013), Efficient Software Reliability analysis with correlated component failures, *IEEE Trans. Reliab.* 62 (1), 244–255.

[12] Hsu, C., Huang, C., (2011), “An adaptive reliability analysis using path testing for complex component based software systems”, *IEEE Trans. Reliab.* 60 (1), 158–170.

[13] Hai Hu, Chang-Hai Jiang, Kai-Yuan Cai, W. Eric Wong, Aditya P. Mathur, (2013), “Enhancing Software Reliability estimates using modified adaptive testing”, *Information and Software Technology Journal Elsevier*, pp. 288–300.

[14] Palviainen, Markov, Evesti, Antti, Ovaska, Eila, (2011), The reliability estimation, prediction and measuring of component – based software. *J. Syst. Softw.,* 1054–1070.

[15] Singh, R.P., Rawat, V., Pawar, M. & Mishra, R.K.(2012), “Design of an optimal multi-layer neural network for eigenfaces based face recognition”, *Recent Research in Science and Technology*, 4(1), 24–32.

[16] Wong, J., Ho, D., & Capretz, L. F. (2008), “Calibrating function point backfiring conversion ratios using neuro-fuzzy technique”, *International Journal of Uncertainty, Fuzziness and Knowledge- Based Systems*, 16(06), 847-862.

[17] Bragina, T. & Tabunshchyk, G. (2011), “Fuzzy model for the software projects design risk analysis”, in *the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), in 11th International Conference* (pp. 335-341). *IEEE.*

[18] Hosmer, D., Lemeshow, S. (1989), *Applied Logistic regression,* John Wiley and Sons.

[19] Kirti Tyagi, Arun Sharma (2014), “An adaptive Neuro Fuzzy model for estimating the reliability of component-based software

systems", *Applied Computing and Informatics 10, Elsevier B.V.* , P. No. 38–51,

[20] Bonthu Kotaiah, T.Arundathi, P.M. Jahan (2014), "Comparative analysis of machine learning techniques for Software Reliability prediction", Mathematical Sciences International Research Journal: Vol. 3 Spl Issue (2014) ISSN 2349-1353.

Appendix- A

MATLAB Program for Practical Validation

```
clear
% MTBF input
aa= VECTOR OF MTBF VALUES;
af=aa./mean(aa);
% Read the FIS structure named as RELB
F=readfis('RELB.fis');
% Evalate the input with the given fuzzy
structure
ff=evalfis([aa./max(aa)+.7,aa+.7]',F)
% this section is regarding ANFIS
% train the data for it give MTBF and
Availability as inputs
trnData = [af , aa];
numMFs = 7;
mfType = 'dsigmf';
epoch_n = 100;
% generate a new anfis with this training data
in_fis = genfis1(trnData,numMFs,mfType);
out_fis = anfis(trnData,in_fis,60);
ff'
mean(ff)
% evaluate the data with input anfis structure
oo=evalfis([aa]',out_fis)'
mean(oo)
```

Mr. Bonthu Kotaiah obtained his Bachelor's degree in Computer Applications from ANU in 2001 and M.C.A from ANU in 2008 and persued Ph.D. from Dept. of I.T, Babasaheb Bhimrao Ambedkar University (A Central University) Lucknow, UP. His research interests include software Engineering, Neural networks, Fuzzy Logic and Neuro-Fuzzy Systems. Presently, he is working as Assistant Professor in the Department of CS & IT, MANUU, Gachibowli, Hyderabad.